MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MAKİNE ÖĞRENMESİ ALGORİTMALARI İLE ŞİFRELİ TRAFİĞİN SINIFLANDIRILMASI

NUR BETÜL DEMİREL

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Bilgi Güvenliği Mühendisliği Programı

DANIŞMAN

Dr. Öğr. Üyesi Aydın ERDEN

İSTANBUL, 2025

MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES

CLASSIFICATION OF ENCRYPTED TRAFFIC WITH MACHINE LEARNING ALGORITHMS

NUR BETÜL DEMİREL

MASTER THESIS

Department Of Computer Engineering

Information Security Engineering Program

Thesis Supervisor

Asst. Prof. Aydın ERDEN

İSTANBUL, 2025

ÖNSÖZ

Tez çalışma sürecim boyunca katkılarını esirgemeyen, her zaman bana destek olan ve vaktini ayıran değerli danışman hocam Dr. Öğr. Üyesi Aydın ERDEN'e,

Tezin araştırma sürecinde sorularıma sabırla yanıt veren arkadaşlarım Cemile UĞURLU, Mesut UĞURLU ve Ahmet AYDIN'a,

Her zaman yanımda olup desteğini esirgemeyen kıymetli eşim Mehmet DEMİREL ve motivasyon kaynağım oğlum Buğra DEMİREL'e

En içten teşekkürlerimi sunarım.

Ocak 2025 **Nur Betül DEMİREL**

İÇİNDEKİLER

ÖZET

MAKİNE ÖĞRENMESİ ALGORİTMALARI İLE ŞİFRELİ TRAFİĞİN SINIFLANDIRILMASI

Günümüzde internet kişisel, ticari ve askeri alanda birçok hassas verinin paylaşıldığı temel iletişim platformu haline gelmiştir. Ağ üzerinden paylaşılan bu verilerin güvenliği, yetkisiz erişim ve kötü niyetli faaliyetlerin artan tehdidi nedeniyle giderek daha fazla endişe kaynağı haline gelmektedir. Bu endişeyi gidermek ve veri güvenliğinin sağlanması için ağ üzerinden iletilen verilerde şifreleme yöntem ve protokolleri kullanılmaktadır. Şifreleme yöntemleri ve protokolleri, kullanıcıların verilerinin güvenliğini sağlamak için faydalı olsa da aynı zamanda ağ üzerindeki faaliyetlerini gizlemek isteyen saldırganlar tarafından da kullanılabilmektedir. Bu nedenle, saldırganların ağ üzerindeki faaliyetlerini gizlemek için şifreleme yöntemlerini kullanması, çözülmesi gereken önemli bir sorun olarak karşımıza çıkmaktadır. Sorunun çözümü ise ancak şifreli ağ trafiğini doğru bir şekilde analiz etmek ve sınıflandırmak ile mümkün olabilecektir. Öte yandan şifreli trafiği deşifreleme yapmadan ağda faaliyet gösteren saldırganları tespit etme konusunda, mevcut ticari güvenlik çözümleri genellikle yetersiz kalmaktadır.

Günümüzde hem sektörde hem de akademide kullanımı gittikçe artan ve daha da artacağı öngörülen makine öğrenmesi teknikleri şifreli ağ trafiğinin sınıflandırılması konusunda çözüm sunma potansiyeline sahiptir. Bu düşünceden hareketle, bu çalışmada şifreli ağ trafiğinin deşifre edilmeden makine öğrenmesi yöntemleri ile sınıflandırılması ve ağ üzerindeki saldırganların tespiti için kullanılabilecek makine öğrenmesi modelleri geliştirilmesi hedeflenmiştir.

Çalışma kapsamında denetimli makine öğrenmesi algoritmalarından K-Nearest Neighbors (KNN), Support Vector Machine (SVM) ve Logistic Regression (LR) algoritmaları ile makine öğrenmesinin bir alt dalı olan topluluk öğrenmesi algoritmalarından Light Gradient Boosting Machine (LGBM) algoritması kullanılmış ve deşifreleme işlemi yapılmadan şifreli paketler olarak iletilen ağ trafik verileri analiz edilmiştir. Çalışmada ağ trafiğinde yaygın olarak kullanılan sınıfları içeren ve gerçek internet trafik verilerinden elde edilmiş olan ISCXVPN2016 veri seti kullanılmıştır.

Çalışma kapsamında denetimli makine öğrenmesi algoritmaları ve topluluk öğrenmesi algoritmasıyla elde edilen başarı oranları karşılaştırılmış ve en yüksek sınıflandırma başarısı LGBM topluluk öğrenmesi algoritması ile %96,53 doğruluk oranında elde edilmiştir.

Gerçek dünyada, modelin başarısı kadar modelin sınıflandırma hızı da önemlidir. Bu nedenle, algoritmaların başarı oranlarının yanı sıra hızının analiz edilebilmesi amacıyla test süreleri de hesaplanmış ve karşılaştırmalara dahil edilmiştir. Bu kapsamda başarı oranında olduğu gibi hız anlamında da en iyi performans gösteren algoritmanın LGBM olduğu ve 0,000074 saniyelik birim test süresi ile makine öğrenmesi algoritmalarından en iyi sonucu veren KNN'den yaklaşık iki kat daha hızlı sınıflandırma yapabildiği sonucuna ulaşılmıştır.

Ayrıca çalışmada elde edilen başarı oranları ile literatürde yer alan diğer çalışmaların başarı oranlarının kıyaslamalarına da yer verilmiştir. Bulunan sonuçlar, yapılan diğer çalışmaların genelinden daha yüksek performans elde edildiğini göstermiştir.

ABSTRACT

CLASSIFICATION OF ENCRYPTED TRAFFIC WITH MACHINE LEARNING ALGORITHMS

In recent years, the Internet has become a fundamental communication platform for sharing sensitive data in personal, commercial, and military domains. As the amount of data shared over the network grows, concerns about the security of this data have increased due to the rising threat of unauthorized access and malicious activities. To mitigate these concerns and ensure data security, encryption methods, and protocols are applied to the data transmitted over the network. While these encryption methods and protocols help secure users' data, they can also be exploited by attackers seeking to conceal their activities on the network. Consequently, the detection of attackers aiming to hide their activities has emerged as a critical issue. The solution to this problem can only be achieved by accurately analyzing and classifying encrypted network traffic. However, current commercial security solutions often fall short when it comes to detecting attackers on the network without decrypting encrypted traffic.

Machine learning techniques, which are increasingly utilized in both industry and academia and which are anticipated to see further growth, offer significant potential for addressing the classification of encrypted network traffic. Motivated by this, the aim of this study is to classify encrypted network traffic without decryption and to develop machine learning models that can be used for the detection of attackers on the network.

In this study, supervised machine learning algorithms such as K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Logistic Regression (LR), along with ensemble learning algorithms like Light Gradient Boosting Machine (LGBM), a subfield of machine learning, are employed. Encrypted network traffic data, transmitted as encrypted packets, is analyzed without performing decryption. The ISCXVPN2016 dataset, which includes commonly used classes in network traffic and is derived from real internet traffic, is utilized in this study.

The classification accuracy rates achieved by the supervised machine learning algorithms and the ensemble learning algorithm are compared. The highest classification success, with an accuracy rate of 96.53%, is obtained using the LGBM ensemble learning algorithm.

In real-world applications, the classification speed of the model is as important as its accuracy. Therefore, in addition to comparing accuracy rates, the test durations for the algorithms are also calculated and included in the comparisons. In terms of speed, similar to accuracy, the LGBM algorithm demonstrates the best performance, achieving classification approximately twice as fast as KNN, which provided the best result among the machine learning algorithms, with a unit test duration of 0.000074 seconds.

Moreover, a comparison of the success rates obtained in this study with those in the existing literature reveals that the results achieved in this work outperform the majority of other studies.

KISALTMALAR

ANN : **A**rtificial **N**eural **N**etwork

ARPANET : **A**dvanced **R**esearch **P**rojects **A**gency **Net**work

BDT : **B**agging **D**ecision **T**ree

CV : **C**ross **V**alidation

CNN : **C**onvolutional **N**eural **N**etwork

DPI : **D**eep **P**acket **I**nspection

DN : **D**oğru **N**egatif

DP : **D**oğru **P**ozitif

DT : **D**ecision **T**ree

GBT : **G**radient **B**oosting **T**ree

GB : **G**radient **B**oosting

HTTPS : **H**yper**t**ext **T**ransfer **P**rotocol **S**ecure

IANA : **I**nternet **A**ssigned **N**umbers **A**uthority

ICQ : **I** **S**eek **Y**ou

IMAPS : **I**nternet **M**essage **A**ccess **P**rotocol **S**ecure

IP : **I**nternet **P**rotocol

IPsec : **I**nternet **P**rotocol **Sec**urity

KNN : **K**-**N**earest **N**eighbors

LGBM : **L**ight **G**radient **B**oosting **M**achine

LR : **L**ogistic **R**egression

LSTM : **L**ong **S**hort-**T**erm **M**emory

MLP : **M**ulti **L**ayer **P**erceptron

NB : **N**aive **B**ayes

P2P : **P**eer to **P**eer

POP3 : **P**ost **O**ffice **P**rotocol **3**

QIM : **Q**uick **I**nstant **M**essenger

RF : **R**andom **F**orest

SFTP : **S**ecure **F**ile **T**ransfer **P**rotocol

S : **S**aniye

SMOTE : **S**ynthetic **M**inority **O**versampling **Te**chnique

SMTPS : **S**imple **M**ail **T**ransfer **P**rotocol **S**ecure

SSL : **S**ecure **S**ockets **L**ayer

TCP : **T**ransmission **C**ontrol **P**rotocol

TLS : **T**ransport **L**ayer **S**ecurity

URL : **U**niform **R**esource **L**ocator

VoIP : **V**oice **o**ver **I**nternet **P**rotocol

VPN : **V**irtual **P**rivate **N**etwork

WWW : **W**orld **W**ide **W**eb

XGBoost : E**x**treme **G**radient **Boost**ing

YN : **Y**anlış **N**egatif

YP : **Y**anlış **P**ozitif

ŞEKİL LİSTESİ

TABLO LİSTESİ

1. GİRİŞ

İnternet, günümüzde bilgiye erişimin ve iletişimin sağlanmasında kullanılan temel araçlardan biri haline gelmiştir. Dünyada milyarlarca kişi istediği anda internetten sunulan geniş bir bilgi yelpazesine erişebilmektedir [1]. İnternetin yaygın olarak kullanılması, başta bilgiye erişimin hızlanması ve iletişimin küresel çapta yapılabilmesi olmak üzere sağlık, güvenlik, sanat, eğlence, teknoloji ve ticaret gibi birçok alanda da önemli avantajlar sağlamaktadır. Ancak bu avantajların yanı sıra, internet üzerinden kişisel, finansal, askeri vb. pek çok hassas verinin iletimi de gerçekleştirildiğinden bu verilerin güvenliği kişi ve kurumlar için giderek artan bir endişe kaynağı haline gelmiştir [2]. Bu noktada internet üzerinden aktarılan verilerin güvenliğinin sağlanması önem arz etmektedir. Güvenliğin sağlanamadığı durumlarda veriler, veriye erişim yetkisi olmayan saldırganlar tarafından kötüye kullanılabilmekte ve siber saldırılara maruz kalınabilmektedir [3]. İnternet üzerinden iletilen verilerin şifrelenmesi, hassas verilerin transferi sırasında izinsiz erişime karşı korunması maksadıyla uygulanan güvenlik önlemlerinden biridir. Öte yandan saldırganlar şifreli trafiği kullanarak kendilerini ve ağdaki izlerini gizleyerek zararlı faaliyetlerde bulunabilmektedirler [4].

Şifreli iletişim kanallarının yaygınlaşması bilgi güvenliği açısından olumlu bir gelişme sağlasa da bu şifreleme çabaları siber güvenlik uzmanlarının ağ trafiğini analiz etme ve güvenlik tehditlerini tanımlamaya yönelik süreçlerini karmaşık bir hale getirmektedir. Bu durum şifreli trafiğin sınıflandırılması konusunun önemini artırmıştır.

Şifreli trafiği doğru bir şekilde sınıflandırmak, tehditlerin tespit edilebilmesi ve önlem alınabilmesi için kritik bir gerekliliktir. İnternet üzerindeki veri trafiği sürekli olarak artmaya devam ederken bu trafiği etkili bir şekilde yönetmek ve güvenliğini sağlamak, işletme ve kurumlar için büyük bir öneme sahiptir. Makine öğrenmesi algoritmaları, büyük veri setleri üzerinde etkin olarak çalışabilmeleri nedeniyle bu alandaki yönetimsel zorluğu ve güvenlik sorunlarını hafifletmek için kullanılan çözümlerden birisi olup şifreli trafiği sınıflandırmada büyük bir potansiyele sahiptir [5]. Bu kapsamda makine öğrenmesi modellerinin kullanıldığı yapay zeka uygulamalarının desteği ile siber güvenlik uzmanlarına, siber tehditleri daha hızlı ve doğru bir şekilde tespit etme ve saldırıları önleme fırsatı sağlanmış olacaktır.

Ağ trafiğinin şifrelenmesinde Virtual Private Network (VPN) ve Hypertext Transfer Protocol Secure (HTTPS) en yaygın kullanılan yöntemlerdir. VPN ile şifreleme işlemi, mevcut paketin başka bir paketin içine alınması ve şifrelenmesi şeklinde gerçekleştirilir [6]. HTTPS ile şifrelemede ise şifreleme algoritmasına göre kullanıcıların siteye girdiği bilgileri içeren web trafiği şifrelenmektedir. Mevcut ticari güvenlik çözümleri, şifrelenmiş trafiğin deşifrelemesini yapmadan trafiği analiz edemediğinden şifrelenmiş veri trafiğinin sınıflandırılması noktasında yetersiz kalabilmektedir. Makine öğrenmesi algoritmaları, şifreli trafiği analiz ederken veri paketinin içeriğini görememesine rağmen farklı özellikleri ve örüntüleri tanımlayabilmekte ve kategorize edilmesini sağlayabilmektedir. Makine öğrenmesi algoritmaları bu özellikleri kullanarak normal ve anormal trafiği sınıflandırabilmektedir. Böylelikle, siber güvenlik uzmanlarının siber tehditleri daha hızlı ve doğru bir şekilde tespit edebilmesinde ve saldırıları önlemelerini sağlamada etkili bir araç haline gelmektedir [7].

Bu çalışmada makine öğrenmesi algoritmaları kullanılarak deşifreleme işlemi yapılmadan şifreli paketler üzerinden, ağ trafiğinde akan verilerin analiziyle; paketlerin durum, boyut ve süre değişkenleri kullanılarak trafiğin etkin bir şekilde sınıflandırılması amaçlanmıştır. Yapılan çalışmada Kanada Siber Güvenlik Enstitüsü tarafından yayımlanmış olan ISCXVPN2016 veri seti [8] kullanılmıştır.

Çalışmanın ikinci bölümünde, ağ trafiğinin şifrelenmesi ve sınıflandırılmasına ilişkin kavramlar açıklanmıştır. Trafik sınıflandırma yöntemlerinden bahsedilmiş ve şifreli trafiğin sınıflandırılmasında kullanılan topluluk öğrenmesi ve derin öğrenme gibi çeşitli algoritmaları da içeren makine öğrenmesi tabanlı sınıflandırma yöntemleri anlatılmıştır. Sonrasında şifreli trafiğin sınıflandırılmasına ilişkin yapılan çalışmalara yer verilmiş ve tablo halinde bir özeti sunulmuştur. Üçüncü bölümde, makine öğrenmesi türleri, çalışmada kullanılan algoritmalar, çalışılan veri seti, geliştirilen model ve model performans metrikleri açıklanmıştır. Dördüncü bölümde, modelin test edilmesi sonrasında elde edilen sonuçlar belirtilmiş ve karşılaştırmalı analizler yapılmıştır. Beşinci ve son bölümde ise çalışmanın genel değerlendirmesine ve tezin ana katkılarında ifade edilen karşılaştırma sonuçlarına yer verilmiş, hem akademik hem de sektörel katkısı incelenmiş ve gelecek çalışmalar için önerilerde bulunulmuştur.

1.1. Problem Durumu

Mevcut ticari güvenlik çözümleri, şifrelenmiş trafiğin deşifrelemesini yapmadan ağ trafiğini analiz edememektedir. Bu nedenle ağ üzerinde iletilen paketleri deşifre etmeye ihtiyaç duymadan analiz yapabilen, yüksek doğruluk oranına sahip, aynı zamanda analizi kısa sürede yaparak bir tehdit durumunda hızlı reaksiyon alınabilmesine imkan veren yapay zeka temelli sistemlere ihtiyaç duyulmaktadır.

1.2. Araştırmanın Amacı

Bu tez çalışmasında, siber güvenlik uzmanlarının ağ üzerindeki veri trafiğini analiz etme ve izleme kabiliyeti güçlendirilerek güvenlik altyapısının ileri seviyelere taşınması ve kullanıcı verilerinin güvenliğinin artırılması hedeflenmektedir.

Çalışmada, şifreli trafiğin sınıflandırılmasında makine öğrenmesi algoritmalarının etkinliğinin test edilmesi maksadı ile denetimli makine öğrenmesi algoritmaları ve makine öğrenmesinin bir alt dalı olan topluluk öğrenmesi algoritmaları ile analizler yapılarak, yüksek düzeyde başarılı ve aynı zamanda hızlı çalışan algoritmalar tespit edilmeye çalışılmıştır.

1.3. Araştırmanın Ana Katkısı

Yapılan çalışmada ana katkılarımız;

- Önerilen modelin test edilebilmesi için çalışmada kullanılan veri setinde yer alan tüm senaryolara ilişkin verilerin analiz edilmesi,
- Kullanılan veri seti ile şifreli trafiğin sınıflandırılmasına ilişkin yapılan çalışmaların karşılaştırmalı bir özetinin çıkarılması,
- Seçilen algoritmaların başarı oranları ve literatürde pek rastlanmayan test süreleri açısından etkinliklerinin incelenmesi,
- Senaryo bazında yapılan diğer çalışmaların sonuçlarıyla birebir karşılaştırma ve değerlendirmelerin yapılması,

- Makine öğrenmesi algoritmaları kullanılarak elde edilen başarı oranları ile topluluk öğrenmesi algoritması kullanılarak elde edilen başarı oranının kıyaslanarak hangi makine öğrenmesi türünün daha başarılı olduğunun tespit edilmesi ve diğer çalışmalarda bulunan sonuçlarla karşılaştırılmasıdır.

1.4. Araştırmanın Sınırlılıkları

Çalışmada algoritmalar, ISCXVPN2016 veri seti ile eğitim ve teste tabi tutulmuştur.

Bölüm 2.4'de yer alan karşılaştırmalı özet tablo, ISCXVPN2016 veri setinin üretildiği yıl olan 2016 yılından itibaren yapılan çalışmalarla sınırlıdır. Ancak literatür taraması için bu sınırlılık söz konusu olmayıp şifreli trafiğin sınıflandırılmasına ilişkin yapılan ve Bölüm 2.4'de yer verilen tüm çalışmaların incelenmesiyle gerçekleştirilmiştir.

Araştırmada kullanılan algoritmaların çalışma hızları Google Colaboratory ortamının [9] sunduğu kaynakların gücü ile sınırlıdır. Öte yandan, tüm algoritmalar aynı platformda test edildiği için makine öğrenmesi yöntemleri arasındaki hız farklarının doğru tespit edilmesinde bir güvenilirlik sorunu oluşturmamaktadır.

2. KAVRAMSAL ÇERÇEVE VE İLGİLİ ARAŞTIRMALAR

İnternet üzerindeki veri trafiği her yıl artmaya devam ederken [10] bu trafiği etkili bir şekilde yönetmek ve güvenliğini sağlamak, işletmeler ve hükümetler için büyük önem arz etmektedir. Makine öğrenmesi, bu alandaki yönetimsel zorluğu ve güvenlik sorunlarını hafifletmek için kullanım potansiyeli taşıyan etkili araçlardan biridir. Böylelikle şifreli trafiği sınıflandırarak, olası tehditlere karşı korunma sağlanması ve ağ güvenliğini artırmak mümkün hale gelebilecektir [7].

2.1. İnternet Ağlarında İletişim

1960'lı yıllarda Advanced Research Projects Agency Network (ARPANET) projesi ile internetin temelleri atılmıştır. ARPANET projesi bilgisayar ağlarının birbirleriyle iletişim kurmasını sağlamak amacıyla geliştirilmiş bir projedir. 1969 yılında, farklı şehirlerde yer alan iki bilgisayar arasında ilk veri transferi gerçekleştirilmiştir [1]. Bu sayede ARPANET işleyen bir sistem olarak kullanılmaya başlanmış ve internetin temelleri atılmıştır. Zamanla diğer üniversitelerin ve kurumların kullanımı ile ARPANET daha da genişlemiştir. 1970-1980'ler boyunca, Transmission Control Protocol/Internet Protocol (TCP/IP) gibi temel iletişim protokolleri geliştirilmiş [11] ve ARPANET günümüz internetinin temellerini oluşturan bir ağ haline gelmiştir. 1990'lı yılların başında sivil kullanımına açılmış böylece yaygınlaşma süreci hız kazanmıştır. Sonrasında ise web tarayıcıları, world wide web (www) ve diğer inovasyonlar ile internet ve internet ağlarında iletişim daha fazla gelişmeye ve yaygınlaşmaya başlamıştır [1].

İnternet günümüzde hayati bir öneme sahiptir çünkü kişisel, ticari ve askeri dahil olmak üzere birçok hassas verinin paylaşıldığı bir platform haline gelmiştir. Bu verilerin güvenliği, yetkisiz erişim ve kötü niyetli faaliyetlerin artan tehdidi nedeniyle giderek daha fazla endişe kaynağı haline gelmiştir. Ağ trafiğinin şifrelenmesi, hassas verilerin transferi sırasında izinsiz erişime karşı korunması maksadıyla uygulanan güvenlik önlemlerinden biridir.

2.2. Ağ Trafiğinin Şifrelenmesi

Ağ trafiğinin şifrelenmesi kişisel veriler, şifreler vb. hassas bilgilerin internet üzerinden güvenli şekilde iletilmesini sağlamaktadır. Şifreli ağ trafiği, bir bilgisayar ağı üzerinde iletilen verilerin şifrelenmiş bir formatta olması anlamına gelmektedir [12].

Bu şifreleme, bilgilerin başta gizliliği olmak üzere güvenliğini sağlamak amacıyla kullanılmaktadır. HTTPS protokolü ve VPN teknolojisi güvenli iletişimin sağlanması için ağ trafiğinin şifrelenmesinde yaygın olarak kullanılan yöntemlerdir [13].

HTTPS, verilerin gizliliğinin ve bütünlüğünün korunması için kullanıcıların web tarayıcıları ve sunucular arasında güvenli bir bağlantı kurulmasını sağlamaktadır. Bu kapsamda HTTPS protokolü, web trafiğini şifreleyerek kullanıcıların güvenli bir şekilde web hizmetlerine erişebilmelerini sağlamaktadır. HTTPS protokolünde, bir web sayfasına erişmek isteyen kullanıcıların tarayıcısı tarafından sunucu ile bir bağlantı başlatılmaktadır. Tarayıcı ve sunucu arasında bir Secure Sockets Layer (SSL)/Transport Layer Security (TLS) el sıkışma gerçekleşmektedir [14]. Kullanılacak şifreleme algoritmasına göre SSL sertifikası, verileri bir koda çevirmekte ve kullanıcıların siteye girdiği bilgileri şifrelemektedir. Şifrelenmiş veriler, tarayıcı sunucu arasında aktarılmaktadır. HTTPS, web trafiğini şifrelemek için SSL veya TLS protokollerini kullanmaktadır [15]. Bu sayede, yetkisiz üçüncü tarafların iletilen veriye erişmeleri veya veriyi değiştirmeleri zorlaştırılmaktadır [16].

Ağ güvenliğinin sağlanması, internet kullanımının gün geçtikçe artmasıyla beraber, modern iletişim ve veri aktarım sistemlerinde kritik bir önem arz etmektedir [17]. VPN teknolojisi, saklı tutulması istenen bilgileri geniş ağlar veya internet gibi küresel ağlar üzerinden güvenli bir biçimde iletebilme imkânı sunan güvenlik protokollerini içermektedir. Bu kapsamda VPN, aktarılmak istenen verinin güvenli bir biçimde şifrelenmesi ve korunmasını sağlamaktadır. Bu teknolojik çözüm, coğrafi olarak uzak noktalarda bulunan iki farklı cihazın internet üzerinden güvenli iletişim sağlayabilmesine olanak tanımaktadır [11].

VPN, kullanıcıların coğrafi kısıtlamaları aşması ve çevrimiçi gizliliği koruyabilmesi için kullanıcıların internet üzerinden özel ağlara güvenli bir şekilde bağlanmalarını sağlamaktadır. Bu kapsamda VPN teknolojisi, kullanıcıların internet trafiğini şifrelemekte ve kişi kimlik bilgilerinin korunmasını sağlamaktadır. Özellikle kurumsal gizlilik ve kişisel gizliliğin korunması için kullanılmaktadır. VPN teknolojisinde VPN tüneli olarak bilinen şifreli bir bağlantı oluşturulmaktadır ve tüm veri trafiği bu güvenli tünelden geçmektedir. Veri paketleri kapsüllenerek farklı bir paket içerisine alınmakta ve şifrelenmektedir [18]. VPN teknolojisinde SSL, Internet Protocol Security (IPsec), OpenVPN gibi protokoller kullanılmaktadır. Bu protokoller, VPN'in cihaz ile hedef sunucu arasında nasıl bir güvenli tünel oluşturacağını belirlemektedir [19].

Bu tez çalışmasında kullanılan veri seti içerisinde *No-VPN* olarak tabir edilen HTTPS protokolüyle şifrelenmiş trafik ve *VPN* olarak tabir edilen VPN teknolojisi kullanılarak şifrelenmiş trafik verileri yer almaktadır.

2.3. Ağ Trafiğinin Sınıflandırılması

Şifreleme yöntem ve protokolleri, kullanıcılar için yararlı olsa da ağdaki faaliyetlerini gizlemek isteyen saldırganlar tarafından da kullanılabilmekte ve kullanıcıların zarar görmesine neden olabilmektedir. Ayrıca şifreli iletişim kanallarının yaygınlaşması bilgi güvenliği açısından olumlu bir gelişme sağlasa da bu şifreleme çabaları siber güvenlik uzmanlarının ağ trafiğini analiz etme ve güvenlik tehditlerini tanımlamaya yönelik süreçlerini karmaşık bir hale getirmektedir. Bütün bu durumlar şifreli trafiğin sınıflandırılması konusunu önemli hale getirmiştir [20].

Şifreli trafiği doğru bir şekilde sınıflandırmak, tehditlerin tespit edilebilmesi ve önlem alınabilmesi için kritik bir gerekliliktir. Şifreli trafiği deşifreleme yapmadan trafik içeriğini analiz etme konusunda, mevcuttaki ticari güvenlik çözümleri genellikle yetersiz kalmaktadır. Makine öğrenmesi algoritmaları ile deşifreleme işlemi yapılmadan şifreli paketler üzerinden paketlerin boyutları, süre vb. değişkenler kullanılarak trafiğin sınıflandırılması yapılabilmektedir. Ayrıca internet üzerindeki veri trafiği sürekli olarak artmaya devam ederken [10] bu trafiği etkili bir şekilde yönetmek ve güvenliğini sağlamak, işletme ve kurumlar için büyük bir öneme sahiptir.

Makine öğrenmesi algoritmaları, büyük veri setleri üzerinde etkin olarak çalışabilmeleri nedeniyle bu alandaki yönetimsel zorluğu ve güvenlik sorunlarını hafifletmek için kullanılan çözümlerden birisi olup şifreli trafiği sınıflandırmada büyük bir potansiyele sahiptir [5]. Bu kapsamda makine öğrenmesi modellerinin kullanıldığı yapay zeka uygulamalarının desteği ile siber güvenlik uzmanlarına, siber tehditleri daha hızlı ve doğru bir şekilde tespit etme ve saldırıları önleme fırsatı sağlanmış olacaktır.

Ağ trafiğinin sınıflandırılması konusunda literatürde sıklıkla kullanılan üç temel yöntem bulunmaktadır. Bunlar port tabanlı sınıflandırma, içerik tabanlı sınıflandırma ve makine öğrenmesi tabanlı sınıflandırmadır [20,21].

2.3.1. Port tabanlı sınıflandırma

Sınıflandırma işleminde en eski yöntem, internet üzerindeki kaynakları tahsis edip düzenleyen bir kuruluş olan Internet Assigned Numbers Authority (IANA) tarafından yönetilen, internetteki hizmetlerin isimlerini ve bu hizmetlere atanmış olan port bilgilerini kaydeden Service Name and Transport Protocol Port Number Registry veritabanından bağlantı noktası numaralarını sorgulamaktır [22]. Bu yöntem siber güvenlik uzmanlarının saldırıları tanımlanmaları ve önlem alarak ağlarını korumaları için bir referans kaynağıdır. Ancak saldırganlar yanıltıcı trafik yönlendirmesi, port sahtekarlığı, port atlama gibi çeşitli aldatıcı metotlar geliştirmişlerdir [16]. Bu nedenle güvenilirliği düşük bir yöntem olarak görülmektedir.

2.3.2. İçerik tabanlı sınıflandırma

Bu yöntem, trafikteki veri paketlerinin içeriğinin analiz edilerek paket başlıkları, Uniform Resource Locator (URL), içerik türü gibi özelliklerine göre ağ trafiğinin sınıflandırılmasına dayanmaktadır [23]. İçerik tabanlı sınıflandırma genelde Deep Packet Inspection (DPI) tekniklerini kullanmaktadır. Bu nedenle trafiği detaylı bir şekilde analiz edebilmekte ve diğer yöntemlere göre daha yüksek doğruluk sağlayabilmektedir. Ancak DPI, ağ trafiğinde belirli örüntüleri veya imzaları arayan ve bu imzalarla eşleşen paketleri tanımlayabilen imza tabanlı bir yaklaşım kullanmaktadır. Bu imza veri tabanının sürekli güncel tutulması gerekmektedir [20]. Aksi halde yeni ve sürekli değişmekte olan tehditlerin analizinde etkisiz kalmaktadır.

Ayrıca içerik tabanlı sınıflandırmada derinlemesine analiz yapıldığından, bu yöntem gizliliğin korunması konusunda endişelere neden olmaktadır. Buna ilaveten büyük miktarlardaki veri trafiğinin detaylı incelenmesi için, yüksek işlem gücü ve depolama gereksinimi gibi kaynak ihtiyaçları da ortaya çıkmaktadır. İçerik tabanlı sınıflandırma yönteminin asıl zayıflığı ise paket içeriğinin şifrelenmiş olduğu durumlarda kullanılamamasıdır [2]. Şifreleme, verilerin güvenli bir şekilde iletimini sağlamaktadır ancak şifreli trafiği izlemek ve sınıflandırmak zordur [24]. Şifreli trafiğin sınıflandırılması, bir ağ üzerinde iletilen verilerin şifrelenmiş olduğu durumda, bu verilerin içeriğini veya özelliklerini anlamaya çalışma sürecidir [25]. Bu sınıflandırma, siber güvenlik uzmanlarına önemli bilgiler sağlayabilmektedir [26]. Trafiğin şifreli olduğu durumlarda, paketlerin ağ üzerindeki hareketlerine ait şifreli olmayan birtakım veriler kullanılarak makine öğrenmesi tabanlı sınıflandırma yöntemi ile sınıflandırma işlemi yapılabilmektedir [27].

2.3.3. Makine öğrenmesi tabanlı sınıflandırma

Topluluk öğrenmesi ve derin öğrenme gibi çeşitli makine öğrenmesi tekniklerini de içeren makine öğrenmesi tabanlı sınıflandırma yönteminde, kullanılan algoritmalar ile modele ağ trafiği öğretilmekte ve modelin trafiği sınıflandırması sağlanmaktadır [28]. Bu yöntem diğer yöntemlere kıyasla daha esnek ve duyarlıdır, böylece karmaşık ve değişken ağ trafiği üzerinde etkin bir sınıflandırma yapılabilmektedir.

Makine öğrenmesi tabanlı sınıflandırma yöntemine ilişkin algoritmalar, şifreli trafiğin sınıflandırılmasında büyük bir potansiyele sahiptir. Şifreli trafiği analiz ederken içeriği görmemesine rağmen ağ paketlerine ait farklı özellikleri ve örüntüleri tanımlayarak sınıflandırabilmektedir. Ayrıca büyük miktarlardaki veri trafiği üzerinde daha iyi performans göstermektedir [23,29,30].

2.4. İlgili Araştırmalar

Şifreli ağ trafiğinin sınıflandırılması konusunda yapılan birçok çalışmada farklı makine öğrenmesi, topluluk öğrenmesi ve derin öğrenme algoritmaları ile çalışılmış ve yüksek başarı oranlarının yakalanması hedeflenmiştir. Bu çalışmada şifreli trafiğin sınıflandırılması için KNN, SVM, LR ve LGBM algoritmaları ile ISCXVPN2016 veri seti kullanılmıştır.

Kullanılan şifreli trafik veri setini oluşturan Draper-gil vd. [8], çalışmalarında yalnızca zamanla ilgili değişkenleri kullanarak şifreli ve VPN trafiği karakterize etmek için akış tabanlı bir sınıflandırma yöntemi önermişlerdir. Normal şifreli trafik için 7 etiket, VPN trafiği için 7 etiket olmak üzere toplam 14 farklı etiketle veri setini oluşturmuşlardır. Bu veri seti ve veri setinde yer alan senaryolara ilişkin detaylı bilgiler 3.2 başlığında yer almaktadır. Draper-gil vd. [8] yaptıkları çalışmada makine öğrenmesi algoritmalarından C4.5 ve KNN'yi kullanmışlardır. Çalışmanın sonucunda C4.5 algoritmasının daha iyi performans sergilediği bulunmuş, her bir senaryo için detaylı sınıflandırma yapılmış ve %80,9 ila %90,6 arasında başarı oranlarına ulaşılmıştır. Bu çalışma ve oluşturulan veri seti sonraki çalışmalar için bir rehber niteliği taşımaktadır.

Makine öğrenmesi tabanlı algoritmalar kullanılarak şifreli trafiğin sınıflandırılması ile ilgili literatürde çok sayıda çalışma bulunmaktadır. Yapılan çalışmalar incelendiğinde; genellikle sadece başarı oranlarına odaklanılan çalışmaların olduğu görülmüştür. Uğurlu vd. [2] şifreli trafiğin sınıflandırılması için yaptıkları çalışmada %94,53 doğruluk oranını elde ettiklerini belirtmişlerdir. Caicedo vd. [6] %94,42 doğruluk oranına ulaşmışlardır. Draper-gil vd. [8] yaptıkları çalışmada %90,6 başarı elde etmişlerdir. Huang vd. [12] f1 skorunda %80 oranı elde edilebileceğini belirtmişlerdir. Ergönül ve Demir [16] yaptıkları çalışmada %97,77 doğruluk, Guo vd. [17] çalışmalarında %99,87 genel doğruluk, Elmaghraby vd. [21] ise %96,8 doğruluk oranı elde etmişlerdir. Seddigh vd. [24] yüksek hızlı ağlarda şifreli trafiğin sınıflandırılması için çalışma yapmışlar ve %90 doğruluk oranı elde etmişlerdir. Obaşı [27], Yamansavaşçılar vd. [31], Bagui vd. [32], Lotfollahi vd. [33], Shapira ve Shavitt [34], Zhang vd. [35], Cheng vd. [36], Zhou vd. [37], Bu vd. [38], Majeed vd. [39], Afuwape vd. [40], Ismailaj vd. [41], Almomani [42], Bozkır vd. [43], Yiğidim [44], Alshammari vd. [45], Di Mauro ve Longo [46], Zhang vd. [47], Chari vd. [48], Yang vd. [49] ve Obaidy vd. [50] de çalışmalarında elde ettikleri başarı oranlarına değinmişler, algoritmaların test sürelerini diğer bir ifadeyle çalışma hızlarını ele almamışlardır. Günümüzde neredeyse her alanda önemli bir kriter olan test süresinin [51] bu alanda yapılan çok az sayıda çalışmada [52] ele alındığı görülmüştür. Test süresi, eğitilen algoritmanın sınıflandırma yapmak için harcadığı süreyi ifade etmektedir.

Çalışmalarında başarı oranının yanı sıra test süresini de göz önünde bulunduran Khatouni ve Heywood [52], şifreli trafiği sınıflandırmak için makine öğrenmesi algoritmalarından Gradient Descent, Naive Bayes (NB), SVM, KNN, Decision Tree (DT) ve Random Forest (RF) kullanmışlardır. Çalışma sonucunda bulunan en yüksek %85 doğruluk oranını DT algoritması ile elde etmişler ve DT algoritmasının RF algoritmasına göre yedi kat daha hızlı çalıştığını belirtmişlerdir. Bu tez kapsamında, kullanılan algoritmaların başarı oranlarının yanı sıra test süreleri de hesaplanmış ve karşılaştırmaları yapılmıştır.

ISCXVPN2016 veri seti kullanılarak yapılan birçok çalışma ile algoritmaların etkinliği analiz edilmiştir. Bu veri setinde üç senaryodan oluşan, her bir senaryoda zamana bağlı alt veri setleri yer almaktadır. Her bir senaryo için detaylı olarak karşılaştırma yapan çalışmaların az sayıda olduğu görülmüştür. Bu çalışmalardan Uğurlu vd. [2], Extreme Gradient Boosting (XGBoost) algoritmasıyla Senaryo A1'de %93,04 kesinlik, Senaryo A2 VPN'de %90,76 doğruluk, Senaryo A2 No-VPN'de %94,53 doğruluk ve Senaryo B'de %86,06 kesinlik oranına ulaşmışlardır. Caicedo vd. [6], Bagging algoritması ile A2 senaryosunda VPN ve No-VPN için sırasıyla %92,82 ve %94,42 doğruluk, B senaryosunda ise %86,94 doğruluk oranı elde etmişlerdir. Draper-gil vd. [8] yaptıkları çalışmada, C4.5 algoritması ile A1 senaryosunda VPN için %89 ve No-VPN için %90,6 kesinlik, A2 senaryosunda VPN için %84 ve No-VPN için %89 kesinlik, B senaryosunda ise %80,9 kesinlik oranı yakalamışlardır. Majeed vd. [39], şifreli trafiğin sınıflandırılması için önerdikleri Horizontal Federated Learning (HFL) yöntemi ile A1 ve A2 senaryosu için %86 doğruluk, B senaryosu için %81 doğruluk oranı elde etmişlerdir. Bozkır vd. [43], yaptıkları çalışmanın sonucunda tüm senaryolarda XGBoost algoritması ile %99 f1 skoruna ulaşmışlardır. Çalışmamızın doğru ve tutarlı bir şekilde karşılaştırılabilmesi için, genel başarı oranlarını karşılaştıran çalışmalardan ziyade "senaryolar" bazında başarı oranlarını karşılaştıran bu çalışmalar ile kıyaslaması yapılarak sonuçlar Tablo 5.1'de sunulmuş ve değerlendirilmiştir.

Şifreli trafiğin sınıflandırılması için makine öğrenmesi ve topluluk öğrenmesi algoritmaları kullanılan çalışmalarda genellikle topluluk öğrenmesi algoritmaları ile daha yüksek başarı oranlarına ulaşıldığı görülmüştür. Uğurlu vd. [2], şifreli trafiğin sınıflandırılması için XGBoost, RF ve DT algoritmalarından en yüksek başarıyı XGBoost topluluk öğrenmesi algoritması ile yakalamıştır.

Caicedo vd. [6] C4.5, Bagging ve Boosting algoritmalarından en iyi sonucu Bagging topluluk öğrenmesi algoritmasının verdiğini belirtmişlerdir. Obaşı [27] Artificial Neural Network (ANN), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), CapsNet, DT ve RF algoritmalarını kullanmış ve en yüksek doğruluk oranını RF topluluk öğrenmesi algoritması ile elde etmiştir. Bagui vd. [32] yaptıkları çalışmada NB, SVM, RF, KNN, LR ve Gradient Boosting Tree (GBT) algoritmalarını kullanmışlardır. Çalışma sonucunda en yüksek başarı oranını RF ve GBT topluluk öğrenmesi algoritmaları ile elde etmişlerdir. Zhang vd. [35] RF, KNN ve C4.5 algoritmalarıyla karşılaştırıldığında RF topluluk öğrenmesi algoritması ile en iyi sonucu elde ettiklerini belirtmişlerdir. Zhou vd. [37] yaptıkları çalışmada SVM, RF, NB, LR ve ANN algoritmasını kullanmış ve en yüksek doğruluk oranını RF topluluk öğrenmesi algoritması ile yakalamışlardır. Afuwape vd. [40] çalışmalarında AdaBoost, Gradient Boosting (GB), RF, Bagging Decision Tree (BDT), KNN, LR, Multi Layer Perceptron (MLP) ve NB algoritmalarını kullanmışlardır. Çalışmanın sonucunda topluluk algoritmalarının daha iyi performans gösterdiğini tespit etmişler ve RF topluluk öğrenmesi algoritması ile en yüksek doğruluk oranına ulaşmışlardır. Bozkır vd. [43] ise GBT, XGBoost ve LGBM algoritmalarını kullanılmışlardır. Çalışmanın sonucunda XGBoost topluluk öğrenmesi algoritması ile en iyi f1 skorunu yakalamışlardır. Yapılan çalışmalar incelendiğinde genel olarak makine öğrenme algoritmalarına kıyasla topluluk öğrenmesi algoritmalarında daha iyi başarı oranlarının elde edildiği görülmüştür. Topluluk öğrenmesi algoritmalarından en iyi sonuçlar ise XGBoost ve RF algoritmaları ile yakalanmıştır. Bu çalışmada, daha önce kullanılanlardan farklı olarak topluluk öğrenmesi algoritmalarından LGBM algoritması ve makine öğrenmesi algoritmalarından KNN, SVM ve LR seçilmiştir. Seçilen topluluk öğrenmesi algoritması ile makine öğrenmesi algoritmalarının sonuçları kıyaslanmıştır.

Makine öğrenmesi algoritmaları ile şifreli trafiğin sınıflandırılmasına ilişkin ISCXVPN2016 veri setinin ortaya çıktığı çalışma [8] dahil olmak üzere ilgili tarihten günümüze aynı veri seti ile bu konuda yapılan çalışmaların [2,6,8,12,16,17,27,31–43] sonuçlarına Tablo 2.1'de karşılaştırmalı bir şekilde yer verilmiştir.

Sonuç olarak literatür incelendiğinde şifreli trafiğin sınıflandırılmasına yönelik yapılan çalışmalarda; makine öğrenmesi algoritmalarından C4.5 ve KNN, topluluk öğrenmesi algoritmalarından XGBoost ve RF, derin öğrenme algoritmalarından CNN ile yüksek başarı oranları elde edildiği görülmüştür. Topluluk öğrenmesi algoritmaları ve hibrit modeller ile sınıflandırma problemlerinde diğer algoritmalardan daha yüksek başarı oranları sağlanmaktadır. Ayrıca yapılan incelemelerde şifreli ağ trafiğinin sınıflandırılması dışında karanlık ağ trafiğinin sınıflandırılması için makine öğrenmesi ve topluluk öğrenmesi yöntemleri kullanılarak yapılan çalışmalara da [53–56] rastlanmıştır.

Tablo 2. 1. Literatür Özeti

Kaynak	Yıl	Veri Seti	Algoritma Türü	Algoritma Adı	Sonuç
Draper-gil vd. [8]	2016	ISCXVPN2016 *(Bu veri setini kendileri hazırlamıştır.)*	Makine öğrenmesi	C4.5 ve KNN	C4.5 ile Senaryo A1 %90,6 kesinlik Senaryo A2 %89 kesinlik Senaryo B %80,9 kesinlik
Yamansavaşçılar vd. [31]	2017	ISCXVPN2016	Makine öğrenmesi Topluluk öğrenmesi	J-48, RF, KNN ve Bayes Net	KNN ile %93,94 doğruluk
Bagui vd. [32]	2017	ISCXVPN2016	Makine öğrenmesi Topluluk öğrenmesi	NB, SVM, KNN, LR, RF ve GBT	RF ve GBT algoritmaları ile %90 doğruluk
Caicedo vd. [6]	2018	ISCXVPN2016	Makine öğrenmesi Topluluk öğrenmesi	C4.5, Bagging ve Boosting	Bagging ile Senaryo A2 %94,42 doğruluk Senaryo B %86,94 doğruluk
Lotfollahi vd. [33]	2019	ISCXVPN2016	Derin öğrenme	SAE ve CNN	CNN ile %98 doğruluk
Shapira ve Shavitt [34]	2019	ISCXVPN2016	Derin öğrenme	CNN	CNN ile %99,7 doğruluk
Zhang vd. [35]	2020	ISCXVPN2016	Makine öğrenmesi Topluluk öğrenmesi	C4.5, KNN ve RF	RF ile 0,93 doğruluk
Guo vd. [17]	2020	ISCXVPN2016	Derin öğrenme Makine öğrenmesi	CAE, CNN, KNN ve C4.5	CAE ile %99,87 doğruluk
Cheng vd. [36]	2020	ISCXVPN2016	Derin öğrenme	1D-CNN, CNN-LSTM ve kendi önerdikleri yöntem	Önerilen yöntem ile %95 kesinlik
Obaşı [27]	2020	ISCXVPN2016 ve Solana Networks	Derin öğrenme Makine öğrenmesi Topluluk öğrenmesi	ANN, CNN, LSTM ve CapsNet DT ve RF	RF ile %96 doğruluk
Zhou vd. [37]	2020	ISCXVPN2016 ve ISCX-Tor/Non-Tor	Makine öğrenmesi Derin Öğrenme Topluluk öğrenmesi	SVM, NB, LR, ANN ve RF	RF ile %98 doğruluk
Bu vd. [38]	2020	ISCXVPN2016	Derin Öğrenme	CNN ve NIN sinir ağı	NIN ile %98 doğruluk

Tablo 2. 2. Literatür Özeti (devam)

Kaynak	Yıl	Veri Seti	Algoritma Türü	Algoritma Adı	Sonuç
Majeed vd. [39]	2020	ISCXVPN2016	Makine öğrenmesi	HFL	HFL ile Senaryo A için %86, Senaryo B için %81 doğruluk
Afuwape vd. [40]	2021	ISCXVPN2016	Makine öğrenmesi Derin Öğrenme Topluluk öğrenmesi	KNN, LR, NB, MLP, AdaBoost, GB, RF ve BDT	RF ile %93,80 doğruluk
Uğurlu vd. [2]	2021	ISCXVPN2016	Makine öğrenmesi Topluluk öğrenmesi	XGBoost, DT ve RF	XGBoost ile Senaryo A1 %93,04 kesinlik Senaryo A2 %94,53 doğruluk Senaryo B %86,06 kesinlik
Huang vd. [12]	2021	ISCXVPN2016	Derin öğrenme Makine öğrenmesi	Kendi önerdikleri hibrit yaklaşım	Önerilen yöntem ile %80 f1 skoru
Ismailaj vd. [41]	2021	ISCXVPN2016	Derin öğrenme Topluluk öğrenmesi	CNN ve LGBM	CNN ile %90 doğruluk
Almomani [42]	2022	ISCXVPN2016	Derin öğrenme Makine öğrenmesi	ANN, SVM ve kendi önerdikleri yöntem	Önerilen yöntem ile %98 doğruluk
Ergönül ve Demir [16]	2022	ISCXVPN2016	Derin öğrenme	LSTM	LSTM ile %97,77 doğruluk
Bozkır vd. [43]	2023	ISCXVPN2016	Topluluk öğrenmesi	GBT, LGBM ve XGBoost	XGBoost ile Senaryo A ve Senaryo B %99 f1 skoru

3. MATERYAL VE YÖNTEM

3.1. Makine Öğrenmesi Algoritmaları

Günümüzde internet ve buna paralel geliştirilen yazılım araçlarının kullanımı gün geçtikçe yaygınlaşmaktadır [10]. Bu nedenle ağ üzerinden aktarılan veri miktarı her yıl artmakta ve artık bireysel olarak incelenip yorumlanamayacak bir noktaya ulaşmış bulunmaktadır. Bu da verilerin analizinde kişilerin değil makinelerin kullanımını zorunlu hale getirmektedir [57]. Bu ihtiyacı karşılamak için yaygın olarak kullanılan ve her geçen gün daha da gelişmekte olan makine öğrenmesi alanı, bilgisayar sistemlerinin deneyim yoluyla öğrenme yeteneğine sahip olduğu bir yapay zeka dalıdır [58].

Algoritmalar, problemleri bilgisayar ortamlarında çözmek için kullanılmakta olup girdilerden çıktılar sağlayan talimat dizisinden oluşmaktadır [59]. Bu girdilerin hangi adımlardan sonra çıktıya dönüşeceği bilinmiyorsa bu durumda bilgisayarların gerekli algoritmaları çıkarma işlemini otomatik olarak yapması beklenmektedir. Makine öğrenmesinde eğer problemleri çözmek için direk bir bilgisayar programı yazılamıyorsa probleme ilişkin birtakım örneklemlerle ve geçmiş deneyimlerle bilgisayarlar eğitilmekte ve bu şekilde programlanmaktadır [60]. Makine öğrenmesi algoritmaları, problemlerin çözümü için gerekli uzmanlıkların bulunmadığı veya uzmanlığın açıklanamadığı, problemlerin ve şartların sıklıkla değiştiği ve her şart veya durum için özelleştirme yapılmasının gerektiği durumlarda kullanılmaktadır [61]. Makine öğrenmesi algoritmaları sistemin öğrenmesini ve karar alabilmesini sağlamaktadır [44].

3.1.1. Denetimsiz makine öğrenmesi

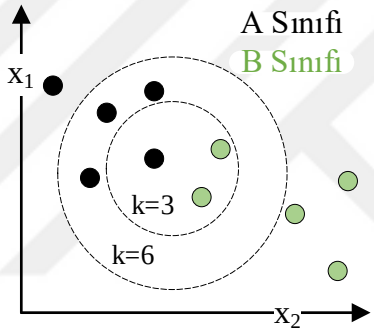
Denetimsiz makine öğrenmesi, etiketlenmemiş veri setlerini kullanarak desenleri veya yapıları belirlemek için kullanılan bir algoritma türüdür [62]. Bu algoritmalar, veri setindeki yapıları kendiliğinden keşfeder ve öğrenir, ancak herhangi bir giriş çıkış çiftiyle eğitilmezler. Denetimsiz öğrenme genellikle veri setindeki içsel yapıları anlamak, benzer örnekleri gruplandırmak veya boyut indirgeme gibi görevler için kullanılır [63].

3.1.2. Denetimli makine öğrenmesi

Denetimli makine öğrenmesi, bir modelin eğitildiği veri setinde giriş verileri ile bu verilere karşılık gelen çıkış etiketleri arasındaki ilişkiyi öğrenmeye çalışan bir algoritmadır [56]. Bu çalışmada KNN, SVM ve LR denetimli makine öğrenmesi algoritmaları kullanılmıştır.

3.1.2.1. K-Nearest Neighbors (KNN)

1951 yılında bir makalede Evelyn Fix ve Joseph Hodges tarafından tanıtılmış olup 1960'larda yaygın olarak kullanılmaya başlanan KNN, sınıflandırma problemlerinde kullanılan etkin ve basit bir algoritmadır. Bir veri noktasının sınıfını belirlemek için çevresindeki k adet en yakın veri noktasına bakılmaktadır [64].

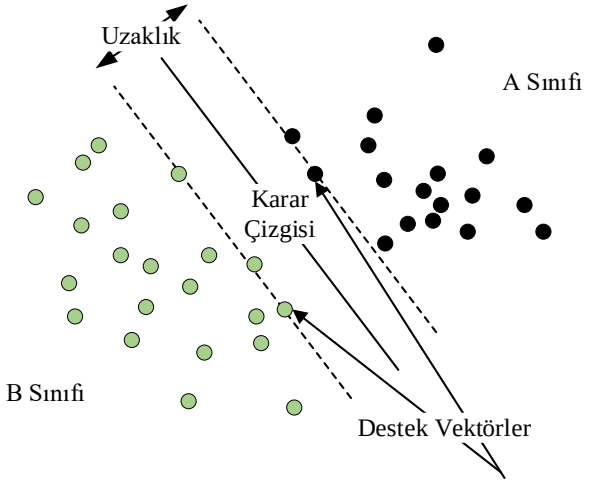


Şekil 3. 1. KNN Sınıflandırma Şeması

Şekil 3.1'de görüldüğü gibi KNN'de hiperparametre olarak komşu sayısı (k) belirlenmektedir. Belirlenen k değerine göre, en yakın komşuların sınıfları incelenir ve en çok görülen sınıf tespit edilir. Bilinmeyen veri noktasının sınıflandırılması, çoğunluğun sınıfı hangisi ise ona göre yapılmakta, en sık kullanılan etiket genellikle tahminin sınıfı olarak kullanılmaktadır [65,66].

3.1.2.2. Support Vector Machine (SVM)

1963 yılında Vladmir N. Vapnik ve Alexey Chervonenkis tarafından geliştirilen bu algoritma zamanla en önemli ve en güçlü denetimli öğrenme tekniklerinden biri haline gelmiş olup sınıflandırma için yaygın olarak kullanılan bu model, temel olarak iki sınıf arasındaki en iyi ayrımı bulmaya odaklanmaktadır [67].



Şekil 3. 2. SVM Sınıflandırma Şeması

Şekil 3.2'deki gibi veri setindeki farklı sınıfları birbirinden ayıran karar çizgileri hyperplanelerdir. Hyperplanelere en yakın veri noktalarına Support Vectors (Şekil 3.2'de destek vektörler olarak ifade edilmiştir) adı verilmektedir. Hyperplanenin iki yanındaki destek vektör noktalarının yani sınıfların arasındaki uzaklık ise margindir. SVM algoritması, veri noktaları ile bu hyperplanenin uzaklıklarını hesaplayarak noktaları sınıflandırmaktadır. SVM, veri seti içerisindeki en yüksek margine sahip hyperplane olan maksimum margin hyperplane bulmayı amaçlamaktadır. En yüksek margini veren en iyi model olarak kabul edilmektedir [65].

3.1.2.3. Logistic Regression (LR)

20. yüzyılın ikinci yarısında yaygınlaşan bu algoritma, sınıflandırma problemleri için kullanılan ve sınıflandırma problemlerine ilk uygulanan algoritmalardandır. İki veya daha fazla sınıf arasındaki ilişkiyi modellemek için kullanılmaktadır [56].

$$z = \beta 0 + \beta 1\, x1 + \beta 2\, x2 + \ldots + \beta n\, xn \qquad (3.1)$$

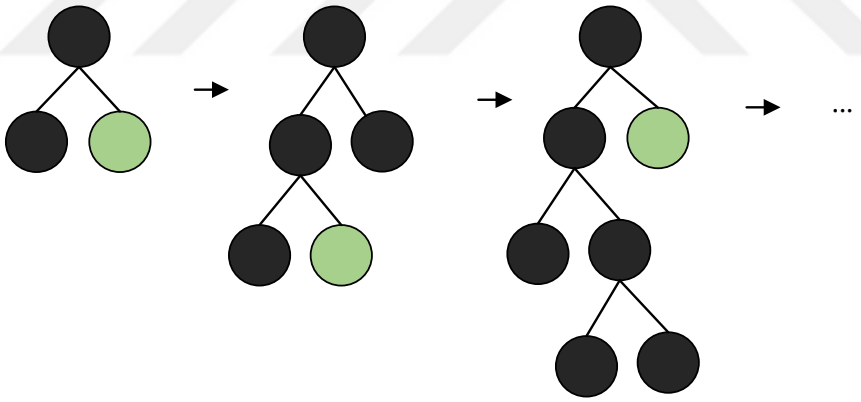
Yukarıda yer alan denklem (3.1) sonucunda ortaya çıkan değer, bir sigmoid fonksiyonu kullanılarak 0 ile 1 arasında bir olasılık değeri haline getirilmektedir. Böylece logistic regression, sınıflandırma algoritmasına dönüştürülmüş olur. Sigmoid fonksiyonu sonucunda ortaya bir olasılık değeri çıkmaktadır. Bu olasılık değerinin sınıflandırma işlemi, iki sınıf arasındaki ayrımı sağlamak için belirlenen karar sınırı değeri baz alınarak yapılmaktadır [65].

3.1.3. Topluluk öğrenmesi

Topluluk öğrenmesi, birden fazla öğrenme algoritmasını bir araya getirip farklı öğrenicileri kullanarak ve bu algoritmaların tahminlerini birleştirerek gerçekleştirilir [68]. Benzer şekilde aynı algoritmaların farklı hiperparametre ayarları birleştirilerek de topluluk öğrenmesi gerçekleştirilebilmektedir. Bu kapsamda her farklı algoritmada aynı veri seti veya veri setinin farklı alt kümeleri kullanılarak elde edilen sonuçlar birleştirilmekte ve tahmin yapılmaktadır. Kullanılan algoritmaların farklı öğrenme stratejilerinin birleşimiyle daha genelleyici ve güçlü tahminler yapılması sağlanmaktadır [69]. Bu çalışmada LGBM topluluk öğrenmesi algoritması kullanılmıştır.

3.1.3.1. Light Gradient Boosting Machine (LGBM)

2017 yılında Microsoft Research Asia tarafından geliştirilen ve Microsoft tarafından yayımlanan bu algoritma, topluluk öğrenmesi tekniklerinden Gradient Boosting'e dayanan bir sınıflandırma algoritması olup Gradient Boosting'e göre hız ve performans açısından daha etkin çalışmaktadır [70].



Şekil 3. 3. Yaprak Bazlı Büyüme [71]

LGBM Algoritması karar ağaçları üzerine kuruludur ve karar ağaçlarını kullanarak tahmin yapmaktadır [57]. Bu kapsamda Gradient Boosting kullanarak zayıf karar ağaçlarını bir araya getirmekte ve güçlü bir karar ağacı oluşturmaktadır. Şekil 3.3'teki gibi en çok kazanç sağlayan dalları genişleten yaprak bazlı büyüme tekniğini kullanmaktadır. Sonrasında model üzerinde tahminler yaparak bu tahminleri veri noktalarının özelliklerine göre sınıflandırmaktadır [72].

3.2. Çalışmada Kullanılan Veri Seti

Bu çalışmada Draper Gil vd. [8] tarafından oluşturulmuş Kanada Siber Güvenlik Enstitüsü tarafından yayımlanan şifreli trafik ve VPN trafiğini içeren kapsamlı bir etiketli veri seti olan ISCXVPN2016 kullanılmıştır. Gerçek ağ trafiğini temsil edecek şekilde hazırlanmış olan bu veri setinde Alice ve Bob adında oluşturulan iki kullanıcı arasındaki trafiğe ilişkin paketler yakalanmıştır. Wireshark ve tcp-dump uygulamaları kullanılarak üretilen verilerin toplam boyutu 28 GB'dır.

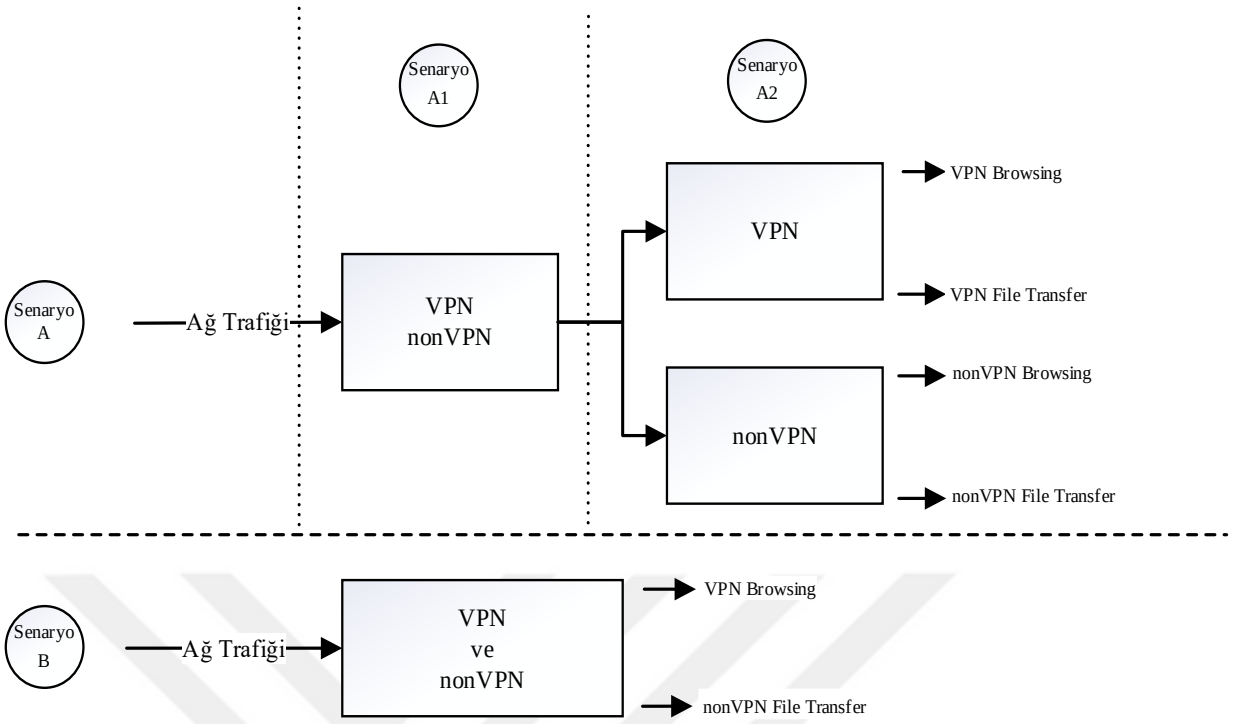
Veri setinde toplam 7 sınıf etiketi yer almaktadır. Browsing sınıf etiketi, kullanıcıların Firefox ve Chrome tarayıcılarında yaptıkları aktiviteler esnasında oluşturulan HTTPS trafik verilerini içermektedir. Email sınıf etiketi, kullanıcıların gmail hesapları kullanılarak atılan e-postalar ile oluşturulan trafik verilerini içermektedir. Bu sınıfta Simple Mail Transfer Protokol Secure (SMTPS), Post Office Protocol 3 (POP3) ve Internet Message Access Protocol Secure (IMAPS) protokolleri kullanılmıştır. Chat sınıf etiketi, anlık mesajlaşma uygulamaları I Seek You (ICQ), Quick Insant Messenger (QIM), Skype, Facebook ve Hangouts kullanılarak oluşturulan trafik verilerini içermektedir. Streaming sınıf etiketi, sürekli ve sabit veri akışı gerektiren Vimeo ve Youtube uygulamalarından izlenen videolar ile oluşturulan trafik verilerini içermektedir.

File Transfer sınıf etiketi, kullanıcıların Skype üzerinden dosya aktarım ve paylaşım işlemleri ile oluşturulan trafik verilerini içermektedir. File Transfer Protocol Secure (FTPS) ve Secure File Transfer Protocol (SFTP) protokolleri kullanılmıştır. Voice over Internet Protocol (VoIP) sınıf etiketi, kullanıcıların Facebook, Hangouts ve Skype kullanarak yaptıkları sesli görüşmeler ile oluşturulan trafik verilerini içermektedir. Peer to Peer (P2P) sınıf etiketi, uTorrent ve Transmission uygulamaları kullanılarak paylaşılan dosyalar ile oluşturulan trafik verilerini içermektedir. Veri seti içeriğindeki sınıflar, bu sınıfların hangi hizmetlerden toplandığı ve sınıfların içeriğine ilişkin bilgiler Tablo 3.1'de yer almaktadır.

Tablo 3. 1. Veri Setindeki Sınıflar

Trafik Sınıfları	Kullanılan Hizmetler	İçeriği
Browsing	Firefox ve Chrome	Kullanıcıların tarayıcısında yapılan aktiviteler esnasında oluşturulan HTTPS trafik verileri
Email	SMTPS, POP3 ve IMAPS	Kullanıcıların gmail hesapları kullanılarak atılan e-postalar ile oluşturulan trafik verileri
Chat	ICQ, QIM, Skype, Facebook ve Hangouts	Anlık mesajlaşma uygulamaları kullanılarak oluşturulan trafik verileri
Streaming	Vimeo ve Youtube	Sürekli ve sabit veri akışı gerektiren uygulamalardan izlenen videolar ile oluşturulan trafik verileri
File Transfer	Skype, FTPS ve SFTP (Filezilla ve harici bir hizmet kullanılarak)	Kullanıcıların Skype üzerinden dosya aktarım ve paylaşım işlemleri ile oluşturulan trafik verileri
VoIP	Facebook, Skype ve Hangouts sesli aramaları (1 saat süreyle)	Kullanıcıların yaptıkları sesli görüşmeler ile oluşturulan trafik verileri
P2P	uTorrent ve Transmission (Bittorrent)	Paylaşılan dosyalar ile oluşturulan trafik verileri

Bu veri setinde Şekil 3.4'teki gibi Senaryo A ve Senaryo B olmak üzere iki senaryo yer almaktadır.



Şekil 3. 4. Veri Seti Senaryoları [8]

Senaryo A'da kategorizasyon iki adımda yapılmaktadır. Senaryo A1'de VPN ile No-VPN trafik birbirinden ayrılmakta olup ikili sınıflandırma yapılmaktadır. Senaryo A2'de her bir trafik (VPN ile No-VPN) ayrı ayrı trafiğin türüne göre kategorize edilmektedir.

Senaryo B'de ise kategorizasyon tek adımda yapılmaktadır. Bu kapsamda VPN ve No-VPN trafikten oluşan ve her trafik için 7 trafik türü içeren toplam 14 sınıftan oluşan karma veri seti kullanılmaktadır.

Veri seti içerisindeki her bir senaryo; verilerin 15, 30, 60 ve 120 saniye (s) olmak üzere zamanla ilgili özelliklerden akış sürelerine göre ayrı ayrı gruplandırılmış ve her biri farklı veri setleri olarak düzenlenmiş haldedir.

Senaryo A1: A1 15s VPN, A1 30s VPN, A1 60s VPN, A1 120s VPN olmak üzere 4 veri seti,
Senaryo A2: A2 15s No-VPN, A2 15s VPN, A2 30s No-VPN, A2 30s VPN, A2 60s No-VPN, A2 60s VPN, A2 120s No-VPN, A2 120s VPN olmak üzere 8 veri seti ve
Senaryo B: B 15s, B 30s, B 60s, B 120s olmak üzere 4 veri seti yer almaktadır.

Bu kapsamda yukarıda belirtilen toplam 16 adet veri setinin her biriyle çalışılmıştır.

Kullanılan ISCXVPN2016 veri setinde bir tanesi sınıf etiketi olmak üzere toplam 24 adet özellik bulunmaktadır. Bu özellikler tanım ve açıklamaları ile birlikte Tablo 3.2'de yer almaktadır.

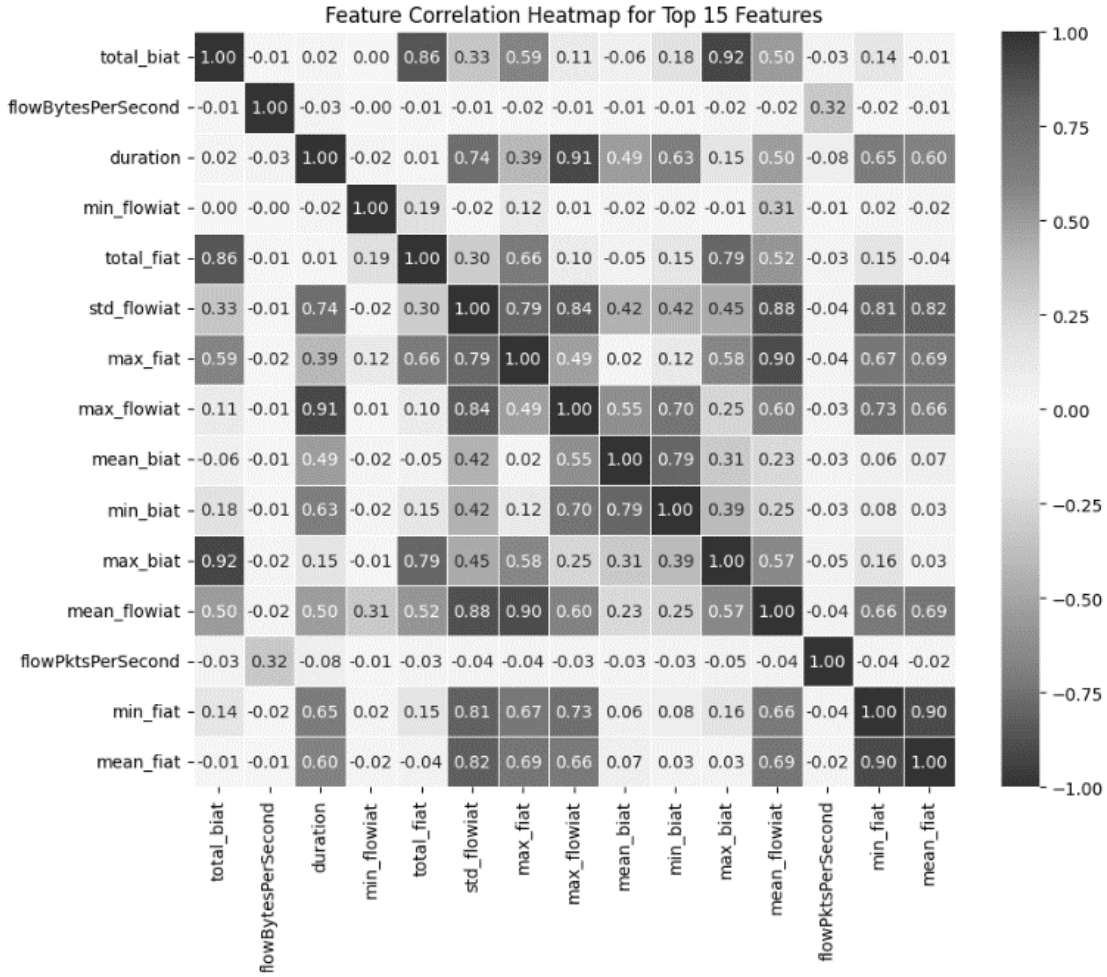
Tablo 3. 2. Özellikler Tablosu [8]

Özellik		Tanım	Açıklama
duration		Akışın süresi	-
fiat	total_fiat	Toplam ileri yönlü varış süresi	İleri yönde gönderilen iki paket arasındaki ilgili süreler
	mean_fiat	Ortalama ileri yönlü varış süresi	
	min_fiat	Minimum ileri yönlü varış süresi	
	max_fiat	Maksimum ileri yönlü varış süresi	
biat	total_biat	Toplam geri yönlü varış süresi	Geri yönde gönderilen iki paket arasındaki ilgili süreler
	mean_biat	Ortalama geri yönlü varış süresi	
	min_biat	Minimum geri yönlü varış süresi	
	max_biat	Maksimum geri yönlü varış süresi	
flowiat	mean_flowiat	Ortalama akış varış süresi	Her iki yönde gönderilen iki paket arasındaki ilgili süreler
	min_flowiat	Minimum akış varış süresi	
	max_flowiat	Maksimum akış varış süresi	
	std_flowiat	Akış varış süresi standart sapması	
active	mean_active	Ortalama aktif süre	Bir akışın boşta kalmadan önce aktif olduğu süreler
	min_active	Minimum aktif süre	
	max_active	Maksimum aktif süre	
	std_active	Aktif süre standart sapması	
idle	mean_idle	Ortalama boşta kalma süresi	Bir akışın aktif hale gelmeden önce boşta kaldığı süreler
	min_idle	Minimum boşta kalma süresi	
	max_idle	Maksimum boşta kalma süresi	
	std_idle	Boşta kalma süresi standart sapması	
fb_psec	flowBytesPerSecond	Saniyede giden bayt sayısı	-
fp_psec	flowPktsPerSecond	Saniyede giden paket sayısı	-
class	class1	Sınıf	-

Veri setinde yer alan özelliklerin arasındaki ilişkileri anlamak amacıyla özelliklerin birbirleri ile olan ilişkileri incelenmiştir. Bu kapsamda önem derecesi yüksek özelliklerin sıcaklık haritası çıkarılmış olup Şekil 3.5'te gösterilmiştir.

Sıcaklık harita analizi, özellik çiftleri arasındaki ilişkilerin ne derece güçlü olduğunu görmeye ve hangi özelliklerin birlikte kullanılması gerektiğini belirlemeye yardımcı olmaktadır [73]. Sıcaklık haritasında her bir özellik çiftinin ilişki değeri haritalandırılmaktadır. Koyu renklerden 1 değeri en güçlü pozitif ilişkiyi, -1 değeri en güçlü negatif ilişkiyi, açık renk 0 değeri ise en zayıf ilişkiyi temsil etmektedir. 1'den veya -1'den 0'a doğru özellik çifti arasındaki ilişki azalarak gitmekte ve buna uygun şekilde renk daha açık hale dönüşmektedir.

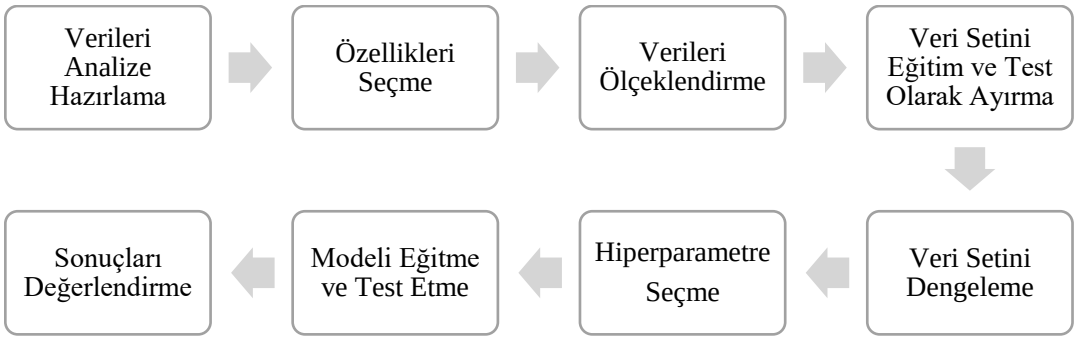
Kullanılan veriler için örnek olarak en yüksek pozitif korelasyonun total_biat ve max_biat arasında olduğu görülmekte olup, değeri 0.92'dir.



Şekil 3. 5. Özelliklerin Sıcaklık Haritası

3.3. Araştırma Modeli

Bu çalışma kapsamında makine öğrenmesi algoritmaları ile şifreli trafiğin sınıflandırılması için atılan adımlar Şekil 3.6'da gösterilmiştir.



Şekil 3. 6. Araştırma Metodolojisinde İzlenen Adımlar

Makine öğrenmesi algoritmaları ile şifreli trafiğin sınıflandırılması çalışması için veriler öncelikle analize hazır hale getirilmiştir. Şifreli trafiği tanımlamak için önemli görülen özellikler tespit edilmiştir. Seçilen özellikler farklı ölçeklere sahip olduğundan ölçeklendirme işlemi yapılmıştır. Sonrasında veriler, eğitim verisi ve test verisi olarak ayrılmıştır. Model eğitim verisi ile eğitilmiş, test verisi ile modelin performansı test edilmiştir.

Optimizasyon kapsamında dengesizlik bulunan verilerde dengeleme işlemi uygulanmış ve kullanılan her bir algoritma için hiperparametre seçimi gerçekleştirilmiştir. Model tekrardan eğitim verisi ile eğitilmiş, test verisi ile test edilmiştir. Son aşamada, çıkan sonuçlar karşılaştırmalı olarak değerlendirilmiş ve yorumlanmıştır.

Bu çalışma kapsamında kullanılan algoritmaların eğitim ve test işlemleri 51.0 GB sistem belleği, 15.0 GB Graphics Processing Unit bilgisayar belleği ve 201.2 GB disk kaynaklarıyla Google Colaboratory Professional ortamında [9] gerçekleştirilmiştir. Python 3 programlama dili ve açık kaynaklı scikit-learn kütüphanesi [74] kullanılmıştır.

3.3.1. Verilerin analize hazır hale getirilmesi

Çalışmada kullanılan ISCXVPN2016 veri seti [8] etiketli verilerden oluşmaktadır. Veri seti öncelikle analize uygun hale getirilmiştir. Verilerin analize uygun hale getirilmesi için eksik verilerin kontrolü yapılmıştır. Eksik veri bulunmadığı görülmüştür.

Çalışmada kullanılan veri seti içinde yer alan sınıf etiketleri sayısal olmayan kategorik verilerden oluşmaktadır. Bu değerler label encoder ile sayısal değerlere dönüştürülmüştür. Tüm sınıfları içeren Senaryo B 15s veri seti için atanan sayısal değerler Tablo 3.3'te gösterilmiştir.

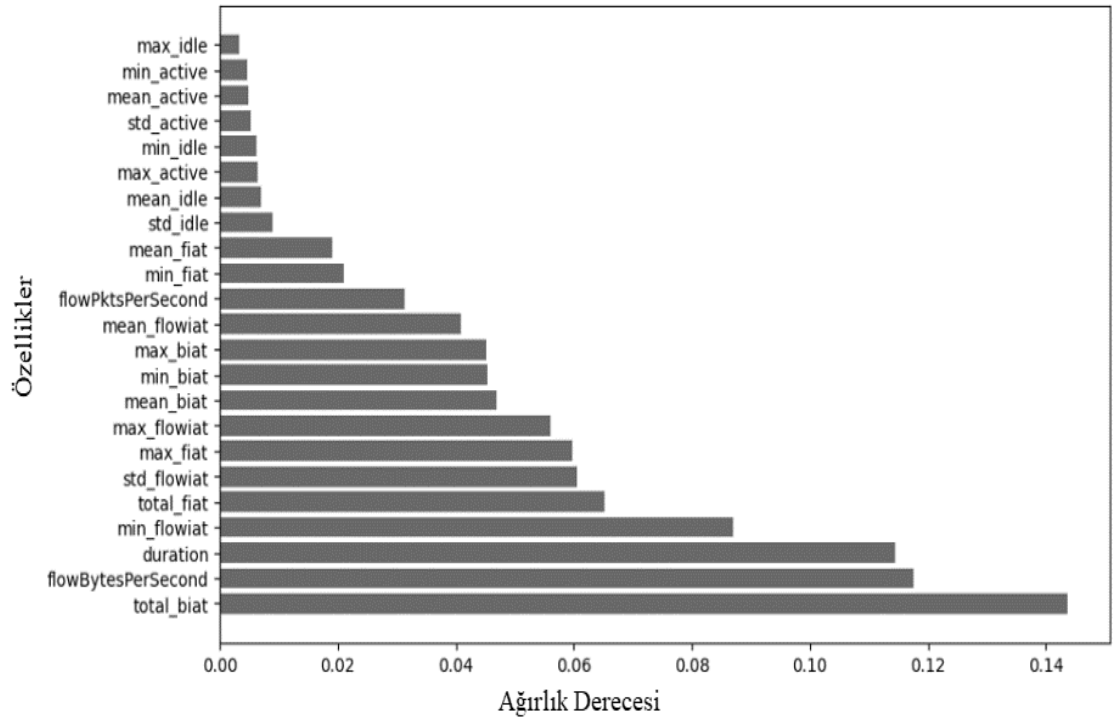
Tablo 3. 3. Label Encoder ile Sınıf Etiketlerini Sayısallaştırma Tablosu

Trafik Sınıfı Değişkeni	Atanan Sayısal Değer
Browsing	0
Chat	1
File Transfer	2
Email	3
P2P	4
Streaming	5
VoIP	6
VPN-Browsing	7
VPN-Chat	8
VPN-File Transfer	9
VPN-Email	10
VPN-P2P	11
VPN-Streaming	12
VPN-VoIP	13

3.3.2. Özelliklerin seçilmesi

Özellik seçimi, makine öğrenmesi modellerinin başarı oranını değiştirmeden hesaplama performansını artırmak, maliyetini azaltmak ve gereksiz veya zararlı özelliklerin etkisini düşürmek için yapılmaktadır [75].

Bu çalışmada karar ağacı sınıflandırıcısı kullanılarak özelliklerin önem dereceleri hesaplanmıştır. Önem derecelerine göre, tüm özelliklerin toplamı bir olacak şekilde yapılan ağırlıklandırma işlemi sonrasında özellik seçimi yapılmıştır. Tüm sınıfları içeren Senaryo B 15s veri seti için ağırlık grafiği Şekil 3.7'de gösterilmiştir.



Şekil 3. 7. Senaryo B 15s'ye Ait Özelliklerin Ağırlık Grafiği

Burada tüm özelliklerin önem dereceleri hesaplanmış ve önem derecesine göre sıralanması sağlanmıştır. Ardından ağırlıklandırma oranları analiz edilmiş, bir akışın boşta kalmadan önce aktif olduğu sürelerin (active) ve yine bir akışın aktif hale gelmeden önce boşta kaldığı sürelerin (idle) sınıflandırmada diğer özelliklere kıyasla çok önemli olmadığı görülmüştür. Bu nedenle ağırlık değeri 0.01'in altında kalan active ve idle ile ilgili toplam 8 adet özellik çıkarılmıştır. Çalışmada kullanılmak üzere 0.01'in üzerindeki diğer 15 adet özellik seçilmiştir. İlgili özellikler ve ağırlık değerleri Tablo 3.4'te gösterilmiş olup seçilen özelliklerin numaraları vurgulu olarak yazılmıştır.

Tablo 3. 4. Senaryo B 15s'ye Ait Özelliklerin Ağırlık Değerleri

No	Özellik Adı	Ağırlığı
1	total_biat	0.143661
2	flowBytesPerSecond	0.117527
3	duration	0.114499
4	min_flowiat	0.087023
5	total_fiat	0.065198
6	std_flowiat	0.060411
7	max_fiat	0.059739
8	max_flowiat	0.056023
9	mean_biat	0.046961
10	min_biat	0.045398
11	max_biat	0.045074
12	mean_flowiat	0.040779
13	flowPktsPerSecond	0.031403
14	min_fiat	0.020947
15	mean_fiat	0.018952
16	std_idle	0.008859
17	mean_idle	0.007039
18	max_active	0.006434
19	min_idle	0.006154
20	std_active	0.005152
21	mean_active	0.004819
22	min_active	0.004677
23	max_idle	0.003259

3.3.3. Verilerin ölçeklendirilmesi (normalizasyonu)

Veri setinde farklı ölçeklere sahip özelliklerin olduğu görülmüş olup tüm sınıfları içeren Senaryo B 15s veri setine ilişkin istatistiki veriler Tablo 3.5'te gösterilmiştir.

Tablo 3. 5. Senaryo B 15s'ye Ait İstatistiki Veriler

		Tanımlayıcı İstatistikler				
		Veri Sayısı (adet)	**Aritmetik Ortalama**	**Standart Sapma**	**Minimum Değer**	**Maksimum Değer**
Seçilen Özellik Adı	**total_biat (s)**	13.130	638.875	2.362.490	1	43.002.380
	flowBytesPerSecond (packets/s)	13.130	454.718	8.982.233	0	565.000.000
	duration (s)	13.130	9.788.733	14.807.250	0	601.405.000
	min_flowiat (s)	13.130	43.427	466.325	721	18.919.900
	total_fiat (s)	13.130	619.785	2.274.082	1	37.680.790
	std_flowiat (s)	13.130	1.032.781	3.595.225	0	136.000.000
	max_fiat (s)	13.130	1.016.790	3.328.565	0	152.000.000
	max_flowiat (s)	13.130	3.852.325	14.311.790	1	601.109.700
	mean_biat (s)	13.130	604.968	2.390.375	0	98.000.000
	min_biat (s)	13.130	2.871.231	10.275.070	1	600.109.700
	max_biat (s)	13.130	928.233	2.504.501	0	43.000.000
	mean_flowiat (s)	13.130	480.361	1.475.132	0	60.700.000
	flowPktsPerSecond (packets/s)	13.130	1.909	15.646	0	666.666
	min_fiat (s)	13.130	3.306.506	10.627.500	1	303.595.700
	mean_fiat (s)	13.130	8.410.637	4.509.882	0	215.000.000

Tabloda, aritmetik ortalamalar ile minimum değer ve maksimum değerler arasında büyük farklar bulunduğu ve Tablo 3.6'da yer alan miktarlarda aykırı değerlerin olduğu görülmektedir.

Tablo 3. 6. Senaryo B 15s'ye Ait Aykırı Değer Sayıları

Seçilen Özellik Adı	**Aykırı Değer (Adet)**
total_biat	646
flowBytesPerSecond	27
duration	136
min_flowiat	57
total_fiat	626
std_flowiat	131
max_fiat	110
max_flowiat	137
mean_biat	65
min_biat	34
max_biat	655
mean_flowiat	129
flowPktsPerSecond	123
min_fiat	108
mean_fiat	101

Bahsedilen özellikler arasındaki farklılık nedeniyle oluşacak dengesizliklerinin giderilmesi ve bu durumun model performansında sapmalara yol açmaması için ölçeklendirme işlemi yapılmıştır [56]. Yapılan ölçeklendirme min-max normalizasyonu ile gerçekleştirilmiştir.

$$z = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{3.2}$$

Yukarıda yer alan denklemde (3.2) gösterildiği gibi herhangi bir değerin ölçeklenmiş hali, o değerin bulunduğu veri setindeki minimum değere uzaklığının, o veri setindeki maksimum ve minimum değerlerin farkına bölünmesidir. Min-max normalizasyonunda, veri seti içinde yer alan değerlerin 0 ve 1 aralığına indirgenmesi sağlanmaktadır. Böylelikle değerler arasındaki büyük farklar ortadan kaldırılmakta, modelin daha dengeli ve daha doğru sonuçlar üretmesi sağlanmaktadır. Yapılan ölçeklendirme sonrası Senaryo B 15s veri setine ilişkin istatistiki veriler Tablo 3.7'de yer almaktadır.

Tablo 3. 7. Ölçeklendirme Sonrası Senaryo B 15s'ye Ait İstatistiki Veriler

		Tanımlayıcı İstatistikler				
		Veri Sayısı (adet)	**Aritmetik Ortalama**	**Standart Sapma**	**Minimum Değer**	**Maksimum Değer**
Seçilen Özellik Adı	**total_biat (s)**	13.130	0,015	0,055	0	1
	flowBytesPerSecond (packets/s)	13.130	0,0008	0,0159	0	1
	duration (s)	13.130	0,016	0,0246	0	1
	min_flowiat (s)	13.130	0,0023	0,0246	0	1
	total_fiat (s)	13.130	0,016	0,06	0	1
	std_flowiat (s)	13.130	0,008	0,026	0	1
	max_fiat (s)	13.130	0,007	0,0219	0	1
	max_flowiat (s)	13.130	0,006	0,0238	0	1
	mean_biat (s)	13.130	0,006	0,0244	0	1
	min_biat (s)	13.130	0,0048	0,0171	0	1
	max_biat (s)	13.130	0,0215	0,058	0	1
	mean_flowiat (s)	13.130	0,0079	0,0243	0	1
	flowPktsPerSecond (packets/s)	13.130	0,0029	0,0235	0	1
	min_fiat (s)	13.130	0,0109	0,035	0	1
	mean_fiat (s)	13.130	0,0039	0,0209	0	1

3.3.4. Verilerin eğitim ve test alt veri setlerine ayrılması

Makine öğrenmesi algoritmalarına öğrenme işlemi yaptırmak için veri seti, eğitim ve test olarak ayrılmıştır. Verilerin ayrımı %70 oranında eğitim verisi ve %30 oranında test verisi olacak şekilde gerçekleştirilmiştir. Bu işlemi gerçekleştirmek için train_test_split işlemi yapılmıştır. Train_test_split fonksiyonunda yer alan random_state parametresi sayesinde, kod her çalıştırıldığında aynı rastgele bölünmenin elde edilmesi sağlanmıştır.

Algoritmaların trafiği doğru bir şekilde sınıflandırmayı öğrenebilmesi için önce eğitim verileri ile modelin öğrenmesi sağlanmış, sonrasında test verileri ile öğrenme performansı test edilmiştir.

Optimizasyon kapsamında aşağıdaki veri dengeleme ve hiperparametre seçimi aşamaları da uygulanarak tekrar eğitim ve test gerçekleştirilmiştir.

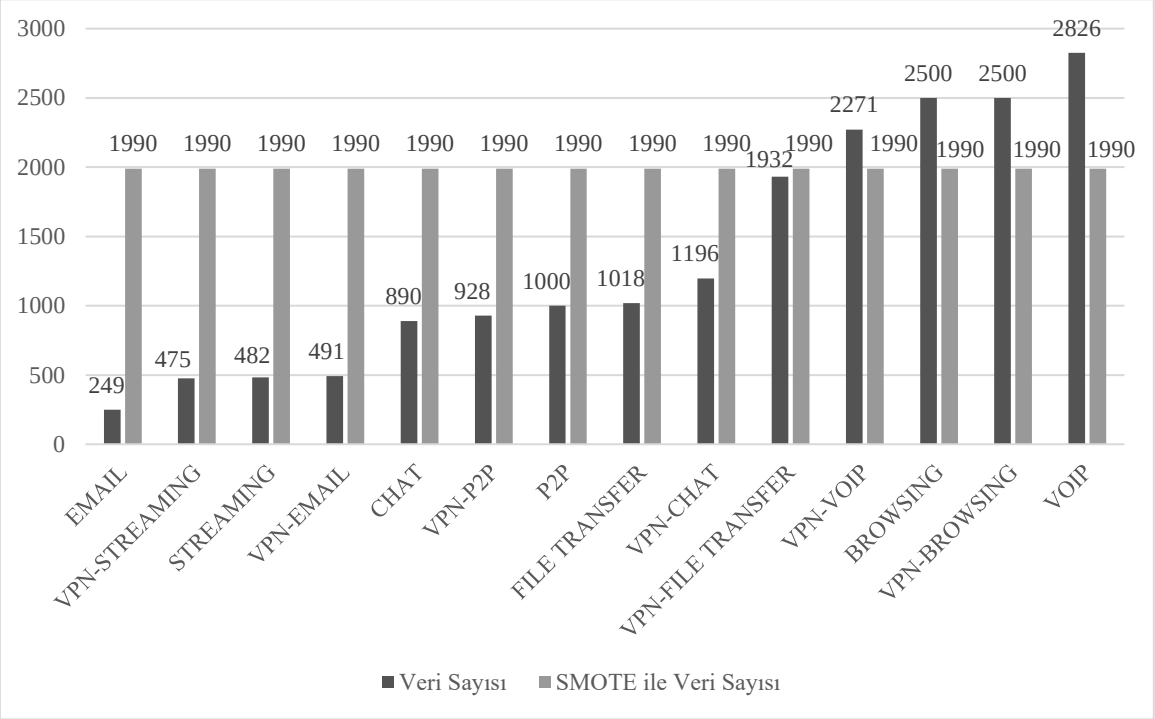
3.3.5. Verilerin dengelenmesi

Verilerin dengelenme ihtiyacı, bir sınıfa ait örneklem sayısının diğer sınıflara göre belirgin olarak fazla veya az olduğu durumlarda doğmaktadır [76]. Verilerdeki bu dengesizlik, sınıflandırma işlemlerinde algoritmaların yanlış öğrenmesine ve düşük performans göstermesine yol açabilmektedir [77].

Bu çalışmada sınıf dengesizliği A2 ve B senaryolarındaki veri setlerinde görülmektedir. Sınıf dengesizliğinin bulunduğu senaryolardaki veri setleri için dengeleme işlemi yapılmıştır. Dengeleme işlemi, yeniden örnekleme yöntemlerinden Synthetic Minority Over-sampling Technique (SMOTE) kullanılarak gerçekleştirilmiştir.

SMOTE ile az sayıdaki sınıflar için sentetik olarak örnek oluşturularak dengesizliğin giderilmesi sağlanmaktadır. Örnek oluşturulurken azınlık sayıdaki sınıflarda yer alan her bir örnek için en yakın komşular belirlenir. Bu komşular arasından rastgele bir komşu seçilerek arasındaki farkın bir katı kadar rastgele vektör oluşturulur ve böylelikle sentetik örneğin pozisyonu belirlenir. Sentetik örnekler azınlık sayıdaki sınıflarda yer alan örneklerin özelliklerini temsil eden yeni örnekler olup var olan örneklerin kombinasyonu oluşturulur [78].

Tüm sınıfları içeren Senaryo B 15s veri setinde, dengeleme öncesi ve sınıf dengesizliğini gidermek için uygulanan SMOTE sonrası veri dağılımı Şekil 3.8'deki gibidir.



Şekil 3. 8. Senaryo B 15s için SMOTE ile Verilerin Dengelenmesi

3.3.6. Hiperparametrelerin seçilmesi

Hiperparametre seçimi, algoritmaların performansını doğrudan etkileyen parametrelerin seçilmesi anlamına gelmektedir [79]. Hiperparametre seçiminde algoritmalar için kullanılacak aday parametrelerin denenmesi amacıyla modelin sürekli olarak eğitilmesi ve test edilmesi gerekmekte ve bu durum yoğun hesaplama işlemleri ve zaman gerektirmektedir [80]. Bu seçim deneme yanılma yoluyla manuel olarak da yapılabilmektedir. Ancak manuel seçimler süreci uzatmakta ve optimal sonuç garantisi de vermemektedir.

Yapılan çalışmada hem süreci uzatmamak hem de hiperparamatrelerin en iyi kombinasyonunu bulabilmek için algoritmaların parametre seçimlerinde rastgele arama tekniği kullanılmıştır. Rastgele arama tekniğinde tanımlanan belirli aralıktaki parametreler rastgele olarak denenmektedir. Böylece hesaplama süresi açısından az zaman gerektirmektedir.

Ayrıca bu teknikte, ızgara araması adı verilen ve belirlenen parametrelerin tüm kombinasyonlarının denendiği tekniğe kıyasla rastgele arama ile önemli parametrelere odaklanılmakta ve kaynakların tüm parametrelere eşit olarak dağıtılarak harcanmasından kaçınılmaktadır. Bu nedenle daha hızlı, verimli ve pratik bir yaklaşımdır [81].

Kullanılan makine öğrenmesi algoritmaları için belirlenen aralıklarda rastgele arama tekniği uygulanmış ve sınıflandırma işlemi için en iyi sonuçları veren parametreler seçilmiştir. Bu işlem her algoritma için ayrı ayrı uygulanmıştır. Parametre seçiminde belirlenen aralıklar Tablo 3.8'de gösterilmiştir.

Tablo 3. 8. Kullanılan Hiperparametre Aralıkları

Algoritma Adı	**Hiperparametre Aralıkları**
KNN	'n_neighbors': [3, 5, 7, 9], 'weights': ['uniform', 'distance'], 'metric': ['euclidean', 'manhattan', 'minkowski'].
SVM	'C': [0.1, 1, 10, 100], 'gamma': ['scale', 0.1, 1, 10], 'kernel': ['linear', 'rbf', 'sigmoid'].
LR	'C': [0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000], 'penalty': ['none', 'l1', 'l2'], 'solver': ['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'], 'max_iter': [30, 50, 100, 200, 500, 1000, 2000, 5000].
LGBM	'num_leaves': sp_randint(6, 100), 'learning_rate': [0.01, 0.05, 0.1, 0.3, 0.5, 0.7], 'max_depth': sp_randint(3, 30), 'min_child_samples': sp_randint(100, 1000), 'subsample': [0.6, 0.7, 0.8, 0.9, 1.0], 'colsample_bytree': [0.5, 0.6, 0.7, 0.8, 0.9, 1.0], 'reg_alpha': [0, 0.1, 0.5, 1, 2, 5, 10], 'reg_lambda': [0, 0.1, 0.5, 1, 2, 5, 10], 'min_child_weight': [1e-3, 1e-2, 0.1, 0.5, 1, 5, 10], 'n_estimators': sp_randint(50, 1000).

Rastgele arama işlemi ile hiperparametre seçiminde çapraz doğrulama da yapılmıştır. Çapraz doğrulama işlemi sayesinde veriler daha verimli şekilde değerlendirilmekte ve modelin daha güvenilir tahmin yapması sağlanmaktadır. Bu işlemde veri setinin her bir parçası hem eğitim hem de test verisi olarak kullanılmaktadır. Çapraz doğrulamada veri seti k sayıda alt kümeye bölünmekte, her alt küme için model eğitilmekte ve modelin performansı değerlendirilmektedir [82]. Bu işlem k kez tekrarlanmakta ve sonuçların ortalamaları alınarak bir performans değeri elde edilmektedir. Bu çalışmada k değeri 3 olarak kullanılmıştır.

3.4. Model Performans Metrikleri

Makine öğrenme algoritmalarının performanslarını ölçebilmek ve analizlerini yapabilmek için kullanılan çok sayıda performans metriği bulunmaktadır. Bunlardan en çok kullanılan performans metriği doğruluktur. Modelin karmaşık veriler arasındaki ilişkileri öğrenemediği durumlarda, çok dengeli olduğu ancak sınıfların önem derecelerinin farklı olduğu durumlarda veya sınıf dengesizliğinin olduğu durumlarda doğruluk metriği tek başına yeterli olmayabilir. Bu gibi durumlarda hassasiyet, duyarlılık, f1 skoru gibi performans metrikleri de kullanılmaktadır [83]. Bu performans metriklerinin hesaplanmasında, makine öğrenmesi modelinin doğru ve yanlış yaptığı sınıflandırma sayılarına ihtiyaç duyulmaktadır. Bu kapsamda gerçek ve tahmin değerlerini içeren karmaşıklık matrisi kullanılmaktadır. Karmaşıklık matrisi yapısı Tablo 3.9'da yer almaktadır [84].

Tablo 3. 9. Karmaşıklık Matrisi Yapısı

Karmaşıklık Matrisi		**Tahmin**	
		C_1 (Pozitif)	**C_2 (Negatif)**
Gerçek	**C_1 (Pozitif)**	Doğru Pozitif (DP)	Yanlış Negatif (YN)
	C_2 (Negatif)	Yanlış Pozitif (YP)	Doğru Negatif (DN)

Karmaşıklık matrisinde yer alan ölçütlere ilişkin açıklamalar aşağıdaki gibidir. DP ve DN ölçütleri modelin başarılı olduğu durumları, YP ve YN ölçütleri ise modelin başarısız olduğu durumları temsil etmektedir [85].

DP: Gerçekte pozitif olan örneklerin, kaçının pozitif olarak sınıflandırıldığını ifade etmektedir.

YP: Gerçekte negatif olan ancak pozitif olarak sınıflandırılan örnek sayısını ifade etmektedir.

DN: Gerçekte negatif olan örneklerin, kaçının negatif olarak sınıflandırıldığını ifade etmektedir.

YN: Gerçekte pozitif olan ancak negatif olarak sınıflandırılan örnek sayısını ifade etmektedir.

Bu çalışma kapsamında kullanılan makine öğrenmesi algoritmalarının performanslarını ölçebilmek ve değerlendirebilmek için test işlemleri sonucunda doğruluk, f1 skoru ve test süresi metrikleri dikkate alınmıştır.

3.4.1. Doğruluk

Doğruluk metriği, sınıflandırma işlemi yapılırken algoritma tarafından doğru sınıflandırılan tahminlerin oranını bulmak için kullanılmaktadır. Doğruluk, aşağıda yer alan 3.3 denklemi ile hesaplanmaktadır.

$$\text{Doğruluk} = \frac{\text{DP} + \text{DN}}{\text{DP} + \text{YP} + \text{DN} + \text{YN}} \tag{3.3}$$

3.4.2. Kesinlik

Kesinlik metriği, sınıflandırma işlemi yapılırken algoritma tarafından pozitif olarak tahmini yapılan örneklemlerin gerçekte ne kadarının pozitif olduğunu ölçmek için kullanılmaktadır. Kısaca doğru pozitif oranıdır. Kesinlik, 3.4 denklemi ile hesaplanmaktadır.

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

3.4.3. Duyarlılık

Duyarlılık metriği, sınıflandırma işlemi yapılırken gerçek pozitif örneklerin ne kadarının doğru tahmin edildiğini ölçmek için kullanılmaktadır. Duyarlılık, 3.5 denklemi ile hesaplanmaktadır.

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

3.4.4. F1 skoru

F1 skoru, kesinlik ve duyarlılığın harmonik ortalaması alınarak modelin ne kadar dengeli bir ölçüm sağladığını ölçmek için kullanılmaktadır. F1 skoru, 3.6 denklemi ile hesaplanmaktadır.

$$\text{F1 skoru} = 2 * \frac{\text{Kesinlik} * \text{Duyarlılık}}{\text{Kesinlik} + \text{Duyarlılık}} \tag{3.6}$$

3.4.5. Test süresi

Test süresi, bir sınıflandırma sürecinde algoritmanın sınıflandırma işlemini ne kadar sürede gerçekleştirdiğini ifade etmektedir. Bu süre, bir sınıflandırma algoritmasının ne kadar hızlı veya ne kadar yavaş çalıştığının belirlenmesine yardımcı olmaktadır. Algoritmanın hızlılığı, sınıflandırma işleminde tespitin daha hızlı yapılması anlamına geldiğinden, daha etkin bir sınıflandırma gerçekleştirilmesi ve kaynakların daha verimli kullanılması açısından fayda sağlayacaktır [51]. Yapılan çalışma kapsamında göz önünde bulundurulan noktalardan birisi de makine öğrenmesi modellerinin sınıflandırmadaki hızlarının karşılaştırılması olmuştur.

4. BULGULAR VE YORUMLAR

Senaryo A1'de yer alan 4 adet, Senaryo A2'de yer alan 8 adet ve Senaryo B'de yer alan 4 adet zaman bazlı veri setlerinin her birinin orijinal halleri ile ayrı ayrı denetimli makine öğrenmesi ve topluluk öğrenmesi algoritmaları kullanılarak çalışma yapılmıştır. Bu kapsamda yapılan çalışmanın ilk aşamasında, veri dengeleme işlemi ve hiperparametre seçimi yapılmamıştır. Algoritmaların varsayılan parametre değerleri kullanılmıştır.

Birinci aşamadaki işlemlere ilaveten veri setlerinde dengeleme ve hiperparametre seçimi ile optimizasyon yapılan ikinci aşama ile ilgili olarak; Senaryo A1'de VPN ve No-VPN olarak yapılan ikili sınıflandırmada verilerin dengeli olduğu görülmüştür. Bu nedenle Senaryo A1'deki zaman bazlı 4 adet veri seti için veri seti dengeleme işlemi yapılmamıştır. Senaryo A2 ve Senaryo B'de yer alan toplam 12 adet veri setinde SMOTE ile dengeleme işlemi gerçekleştirilmiştir. Tüm veri setlerinde rastgele arama ile elde edilen algoritmaların en iyi parametreleri kullanılmıştır.

4.1. Denetimli Makine Öğrenmesi Algoritmaları ile Elde Edilen Sonuçlar

KNN, SVM ve LR algoritmaları kullanılarak orijinal veri seti ile elde edilen sonuçlar sırasıyla Tablo 4.1, Tablo 4.2 ve Tablo 4.3'te yer almaktadır.

Tablo 4. 1. KNN Algoritmasının Orijinal Veri Seti ile Başarı Oranları

Algoritma / **Senaryolar**	**KNN**	
	Doğruluk	F1 Skor
Senaryo A1 15s	**0,8211***	**0,8212***
Senaryo A1 30s	0,8144	0,8140
Senaryo A1 60s	0,7889	0,7891
Senaryo A1 120s	0,7935	0,7930
Senaryo A2 No-VPN 15s	0,8539	0,8521
Senaryo A2 VPN 15s	0,8043	0,8028
Senaryo A2 No-VPN 30s	***0,8613****	***0,8578****
Senaryo A2 VPN 30s	0,8311	0,8290
Senaryo A2 No-VPN 60s	0,8050	0,8013
Senaryo A2 VPN 60s	0,7967	0,7915
Senaryo A2 No-VPN 120s	0,8551	0,8480
Senaryo A2 VPN 120s	0,7834	0,7791
Senaryo B 15s	0,7166	0,7147
Senaryo B 30s	**0,7232***	**0,7286***
Senaryo B 60s	0,7187	0,7245
Senaryo B 120s	0,6754	0,6681

* Kalın yazılan değerler, ilgili senaryoda elde edilen en yüksek başarı oranlarıdır.
**Kalın ve italik yazılan değerler, tüm senaryolarda elde edilen en yüksek başarı oranlarıdır.

Tablo 4. 2. SVM Algoritmasının Orijinal Veri Seti ile Başarı Oranları

Algoritma / Senaryolar	SVM	
	Doğruluk	F1 Skor
Senaryo A1 15s	0,6075	0,5924
Senaryo A1 30s	0,6078	0,5859
Senaryo A1 60s	0,6123	0,6108
Senaryo A1 120s	**0,6173***	**0,6125***
Senaryo A2 No-VPN 15s	***0,6874****	***0,6310****
Senaryo A2 VPN 15s	0,5752	0,5106
Senaryo A2 No-VPN 30s	0,6460	0,5791
Senaryo A2 VPN 30s	0,5890	0,5277
Senaryo A2 No-VPN 60s	0,5812	0,5262
Senaryo A2 VPN 60s	0,5320	0,4448
Senaryo A2 No-VPN 120s	0,6578	0,5767
Senaryo A2 VPN 120s	0,5811	0,4841
Senaryo B 15s	**0,4394***	**0,3987***
Senaryo B 30s	0,3938	0,3595
Senaryo B 60s	0,3767	0,3443
Senaryo B 120s	0,4049	0,3655

* Kalın yazılan değerler, ilgili senaryoda elde edilen en yüksek başarı oranlarıdır.
**Kalın ve italik yazılan değerler, tüm senaryolarda elde edilen en yüksek başarı oranlarıdır.

Tablo 4. 3. LR Algoritmasının Orijinal Veri Seti ile Başarı Oranları

Algoritma / Senaryolar	LR	
	Doğruluk	F1 Skor
Senaryo A1 15s	**0,5721***	**0,5678***
Senaryo A1 30s	0,5612	0,5389
Senaryo A1 60s	0,5603	0,5231
Senaryo A1 120s	0,5518	0,5156
Senaryo A2 No-VPN 15s	0,5825	0,4790
Senaryo A2 VPN 15s	0,4847	0,3503
Senaryo A2 No-VPN 30s	0,5742	0,4801
Senaryo A2 VPN 30s	0,5050	0,3862
Senaryo A2 No-VPN 60s	0,5190	0,4460
Senaryo A2 VPN 60s	0,4786	0,3827
Senaryo A2 No-VPN 120s	***0,6255****	***0,5380****
Senaryo A2 VPN 120s	0,5053	0,3644
Senaryo B 15s	0,2601	0,1882
Senaryo B 30s	0,3219	0,2420
Senaryo B 60s	0,3289	0,2619
Senaryo B 120s	**0,3354***	**0,2634***

* Kalın yazılan değerler, ilgili senaryoda elde edilen en yüksek başarı oranlarıdır.
**Kalın ve italik yazılan değerler, tüm senaryolarda elde edilen en yüksek başarı oranlarıdır.

Senaryo A1 göz önünde bulundurulduğunda 15s, 30s, 60s ve 120s veri setlerinde KNN ve LR algoritmaları kullanıldığında başarı oranlarının gittikçe azaldığı, SVM algoritmasında ise başarı oranının gittikçe arttığı görülmüştür. Bu senaryoda en yüksek başarı oranlarına KNN algoritması ile ulaşılmıştır. En yüksek başarı oranları, 0,8211 doğruluk ve 0,8212 f1 skor ile Senaryo A1 15s veri setinde elde edilmiştir.

Senaryo A2 göz önünde bulundurulduğunda 15s, 30s, 60s ve 120s VPN ve No-VPN veri setlerinde kullanılan KNN, SVM ve LR algoritmalarının hepsinin başarı oranlarında azalma eğilimi görülmüştür. Bu senaryoda en yüksek başarı oranlarına KNN algoritması ile ulaşılmıştır. En yüksek başarı oranları, 0,8613 doğruluk ve 0,8578 f1 skor ile Senaryo A2 No-VPN 30s veri setinde elde edilmiştir.

Senaryo B göz önünde bulundurulduğunda 15s, 30s, 60s ve 120s veri setlerinde KNN algoritması kullanıldığında başarı oranlarında azalma eğilimi görülmüştür. Bu senaryoda en yüksek başarı oranlarına KNN algoritması ile ulaşılmıştır. En yüksek başarı oranları, 0,7232 doğruluk ve 0,7286 f1 skor ile Senaryo B 30s veri setinde elde edilmiştir. Bu senaryoda SVM ve LR algoritmalarında başarı yakalanamamıştır (ortalama oran 0,33).

Tüm senaryolar için Tablo 4.4'teki varsayılan parametre değerlerinin kullanıldığı orijinal veri seti ile en yüksek başarı oranlarına denetimli makine öğrenmesi algoritmalarından KNN ile ulaşılmıştır. KNN algoritması ile ulaşılan en yüksek başarı oranları, 0,8613 doğruluk ve 0,8578 f1 skor ile Senaryo A2 No-VPN 30s veri setinde elde edilmiştir.

Tablo 4. 4. KNN Algoritması için Varsayılan Parametre Değerleri

Parametrenin Adı	**Değeri**
weights	uniform
n_neighbors	5
metric	minkowski

Orijinal veri seti ile eğitime ve teste tabi tutulan KNN, SVM ve LR algoritmalarının test süreleri Tablo 4.5'te yer almaktadır. Test sürelerinin karşılaştırması, tüm veri setlerindeki kategorizasyon sürelerinin aritmetik ortalaması alınarak gerçekleştirilmiştir. Bu kapsamda test süreleri; KNN için girdi başına 0,000098 saniye olup toplam 0,33 saniye, SVM için girdi başına 0,000385 saniye olup toplam 1,56 saniye ve LR için girdi başına 0,000004 saniye olup toplam 0,01 saniye olarak hesaplanmıştır.

Tablo 4. 5. KNN, SVM ve LR Algoritmalarının Test Süreleri

Algoritma / Senaryolar	Test Edilen Veri Sayısı (adet)	KNN		SVM		LR		
		Test Süresi (s)	Birim Test Süresi (s/adet)	Test Süresi (s)	Birim Test Süresi (s/adet)	Test Süresi (s)	Birim Test Süresi (s/adet)	
Senaryo A1 15s	5.628	0,34	0,000060	2,73	0,000485	0,04	0,000007	Orijinal veri seti ile
Senaryo A1 30s	4.397	0,43	0,000098	2,09	0,000475	0,01	0,000002	
Senaryo A1 60s	4.655	0,56	0,000120	1,80	0,000387	0,01	0,000002	
Senaryo A1 120s	3.236	0,61	0,000189	1,13	0,000349	0,01	0,000003	
Senaryo A2 No-VPN 15s	2.690	0,25	0,000093	0,63	0,000234	0,01	0,000004	
Senaryo A2 VPN 15s	2.938	0,16	0,000054	0,96	0,000327	0,01	0,000003	
Senaryo A2 No-VPN 30s	2.076	0,09	0,000043	0,41	0,000197	0,01	0,000005	
Senaryo A2 VPN 30s	2.321	0,13	0,000056	0,59	0,000254	0,01	0,000004	
Senaryo A2 No-VPN 60s	2.574	0,09	0,000035	0,69	0,000268	0,01	0,000004	
Senaryo A2 VPN 60s	2.081	0,22	0,000106	0,46	0,000221	0,01	0,000005	
Senaryo A2 No-VPN 120s	1.546	0,07	0,000045	0,18	0,000116	0,01	0,000006	
Senaryo A2 VPN 120s	1.690	0,39	0,000231	0,33	0,000195	0,01	0,000006	
Senaryo B 15s	5.704	0,72	0,000126	6,31	0,001106	0,01	0,000002	
Senaryo B 30s	4.543	0,49	0,000108	2,97	0,000654	0,01	0,000002	
Senaryo B 60s	4.825	0,45	0,000093	2,03	0,000421	0,01	0,000002	
Senaryo B 120s	3.501	0,39	0,000111	1,62	0,000463	0,01	0,000003	
Ortalama Test Süreleri		**0,33**	**0,000098**	**1,56**	**0,000385**	**0,01**	**0,000004**	

Bu sonuçlar, kategorizasyonu gerçekleştirme süresinin en kısa olması nedeniyle en hızlı çalışan algoritmanın LR, kategorizasyonu gerçekleştirme süresinin en uzun olması nedeniyle en yavaş çalışan algoritmanın ise SVM algoritması olduğunu göstermektedir.

Çalışmada, veri dengeleme ve hiperparametre seçimi işlemleri ile yapılan optimizasyon sonrasında veriler tekrar eğitim ve teste tabi tutulmuştur. KNN, SVM ve LR algoritmaları kullanılarak optimizasyon sonrası elde edilen sonuçlar sırasıyla Tablo 4.6, Tablo 4.7 ve Tablo 4.8'de yer almaktadır. Bu tablolarda, Senaryo A2 ve Senaryo B'deki toplam 12 adet veri seti için SMOTE ile dengeleme, rastgele arama ve çapraz doğrulama ile hiperparametre seçimi yapıldıktan sonra elde edilen sonuçlar yer almaktadır. Dengeleme işlemi yapılmayan Senaryo A1'de yer alan 4 adet veri seti için rastgele arama ve çapraz doğrulama ile hiperparametre seçimi yapıldıktan sonra elde edilen sonuçlara da aynı tablolarda yer verilmiştir.

Tablo 4. 6. KNN Algoritmasının Optimizasyon Sonrası Veri Seti ile Başarı Oranları

Algoritma / **Senaryolar**	**KNN**		
	Doğruluk	F1 Skor	
Senaryo A1 15s	**0,8277***	**0,8421***	Hiperparametre seçimi
Senaryo A1 30s	0,8279	0,8289	
Senaryo A1 60s	0,8089	0,8191	
Senaryo A1 120s	0,8127	0,8209	
Senaryo A2 No-VPN 15s	***0,9126****	***0,8591****	Dengelenmiş veri setleri ile Hiperparametre seçimi
Senaryo A2 VPN 15s	0,8695	0,8203	
Senaryo A2 No-VPN 30s	0,9109	0,8505	
Senaryo A2 VPN 30s	0,8989	0,8454	
Senaryo A2 No-VPN 60s	0,8675	0,8186	
Senaryo A2 VPN 60s	0,8739	0,7999	
Senaryo A2 No-VPN 120s	0,8993	0,8433	
Senaryo A2 VPN 120s	0,8725	0,7972	
Senaryo B 15s	0,8219	0,7334	
Senaryo B 30s	**0,8379***	**0,7357***	
Senaryo B 60s	0,8187	0,7245	
Senaryo B 120s	0,8214	0,7140	

* Kalın yazılan değerler, ilgili senaryoda elde edilen en yüksek başarı oranlarıdır.
**Kalın ve italik yazılan değerler, tüm senaryolarda elde edilen en yüksek başarı oranlarıdır.

Tablo 4. 7. SVM Algoritmasının Optimizasyon Sonrası Veri Seti ile Başarı Oranları

Algoritma / Senaryolar	SVM		
	Doğruluk	F1 Skor	
Senaryo A1 15s	0,6167	0,6113	Hiper-parametre seçimi
Senaryo A1 30s	0,6445	0,6367	
Senaryo A1 60s	0,6323	0,6308	
Senaryo A1 120s	**0,6576***	**0,6388***	
Senaryo A2 No-VPN 15s	0,7403	0,7633	Dengelenmiş veri setleri ile Hiperparametre seçimi
Senaryo A2 VPN 15s	0,6511	0,6411	
Senaryo A2 No-VPN 30s	0,6427	0,6866	
Senaryo A2 VPN 30s	***0,7602****	***0,7176****	
Senaryo A2 No-VPN 60s	0,6372	0,6374	
Senaryo A2 VPN 60s	0,6866	0,6293	
Senaryo A2 No-VPN 120s	0,6748	0,6711	
Senaryo A2 VPN 120s	0,6969	0,6049	
Senaryo B 15s	**0,5914***	**0,5696***	
Senaryo B 30s	0,4245	0,4234	
Senaryo B 60s	0,3767	0,3443	
Senaryo B 120s	0,4128	0,3810	

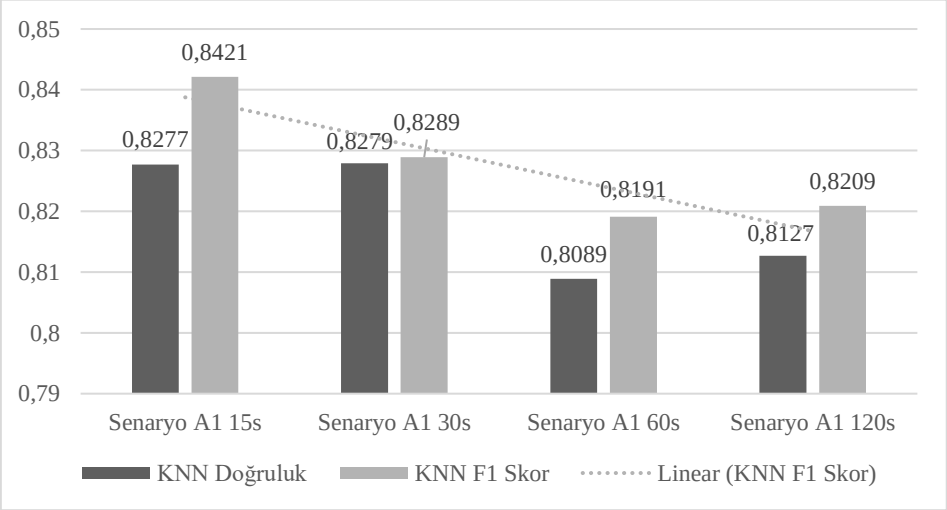
* Kalın yazılan değerler, ilgili senaryoda elde edilen en yüksek başarı oranlarıdır.
**Kalın ve italik yazılan değerler, tüm senaryolarda elde edilen en yüksek başarı oranlarıdır.

Tablo 4. 8. LR Algoritmasının Optimizasyon Sonrası Veri Seti ile Başarı Oranları

Algoritma / Senaryolar	LR		
	Doğruluk	F1 Skor	
Senaryo A1 15s	**0,5963***	**0,5877***	Hiper-parametre seçimi
Senaryo A1 30s	0,5956	0,5874	
Senaryo A1 60s	0,5603	0,5231	
Senaryo A1 120s	0,5904	0,5802	
Senaryo A2 No-VPN 15s	***0,6646****	***0,7197****	Dengelenmiş veri setleri ile Hiperparametre seçimi
Senaryo A2 VPN 15s	0,5692	0,5486	
Senaryo A2 No-VPN 30s	0,6489	0,6746	
Senaryo A2 VPN 30s	0,6278	0,5735	
Senaryo A2 No-VPN 60s	0,5537	0,6009	
Senaryo A2 VPN 60s	0,5430	0,5044	
Senaryo A2 No-VPN 120s	0,6171	0,6293	
Senaryo A2 VPN 120s	0,5518	0,4746	
Senaryo B 15s	0,3608	0,4008	
Senaryo B 30s	**0,3950***	**0,3778***	
Senaryo B 60s	0,3289	0,2619	
Senaryo B 120s	0,3705	0,3281	

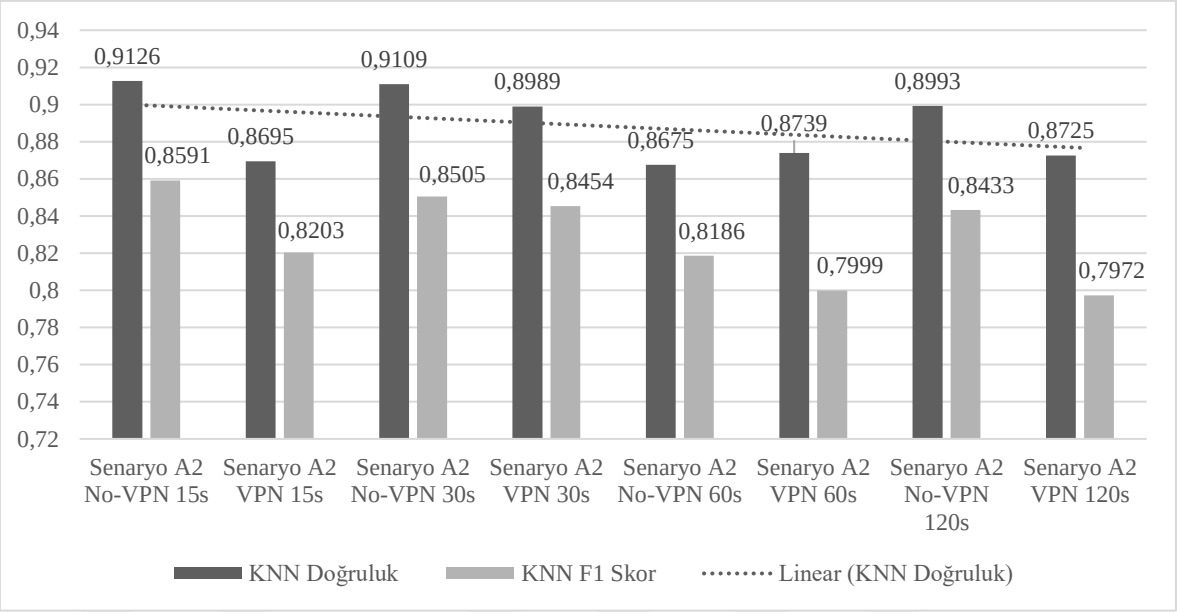
* Kalın yazılan değerler, ilgili senaryoda elde edilen en yüksek başarı oranlarıdır.
**Kalın ve italik yazılan değerler, tüm senaryolarda elde edilen en yüksek başarı oranlarıdır.

Senaryo A1'de hiperparametre seçimi sonrasındaki sonuçlar göz önünde bulundurulduğunda 15s, 30s, 60s ve 120s veri setlerinde KNN ve LR algoritmaları kullanıldığında başarı oranlarının gittikçe azaldığı, SVM algoritmasında ise başarı oranının gittikçe arttığı görülmüştür. Senaryo A1'in en yüksek başarı oranları KNN algoritmasında 0,8277 doğruluk ve 0,8421 f1 skor ile Senaryo A1 15s veri setinde elde edilmiştir. Senaryo A1 veri setlerinde elde edilen başarı oranlarının grafiği Şekil 4.1'de gösterilmiştir.



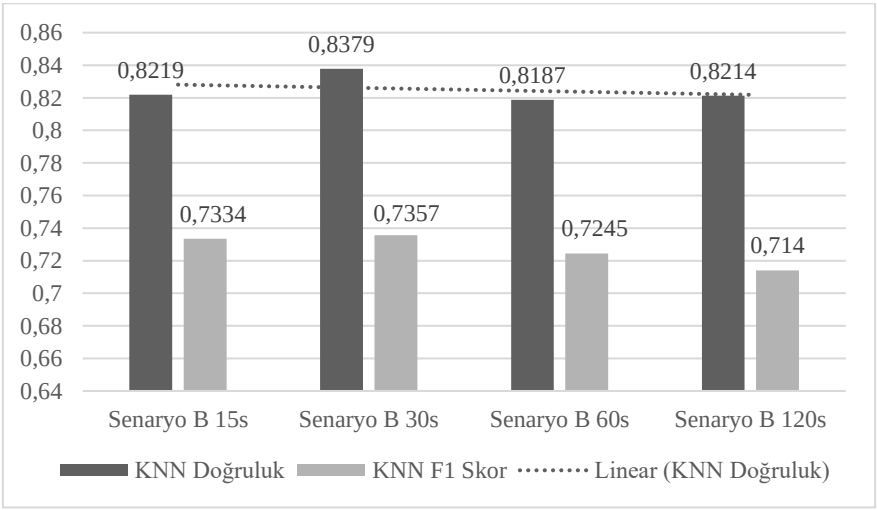
Şekil 4. 1. KNN Algoritmasının Optimizasyon Sonrası Senaryo A1 Başarı Oranları

Senaryo A2'de dengeleme işlemi ve hiperparametre seçimi sonrasındaki sonuçlar göz önünde bulundurulduğunda 15s, 30s, 60s ve 120s VPN ve No-VPN veri setlerinde kullanılan KNN, SVM ve LR algoritmalarının hepsinin başarı oranlarında azalma eğilimi görülmüştür. Senaryo A2'nin en yüksek başarı oranları KNN algoritmasında 0,9126 doğruluk ve 0,8591 f1 skor ile Senaryo A2 No-VPN 15s veri setinde elde edilmiştir. Senaryo A2 veri setlerinde elde edilen başarı oranlarının grafiği Şekil 4.2'de gösterilmiştir.



Şekil 4. 2. KNN Algoritmasının Optimizasyon Sonrası Senaryo A2 Başarı Oranları

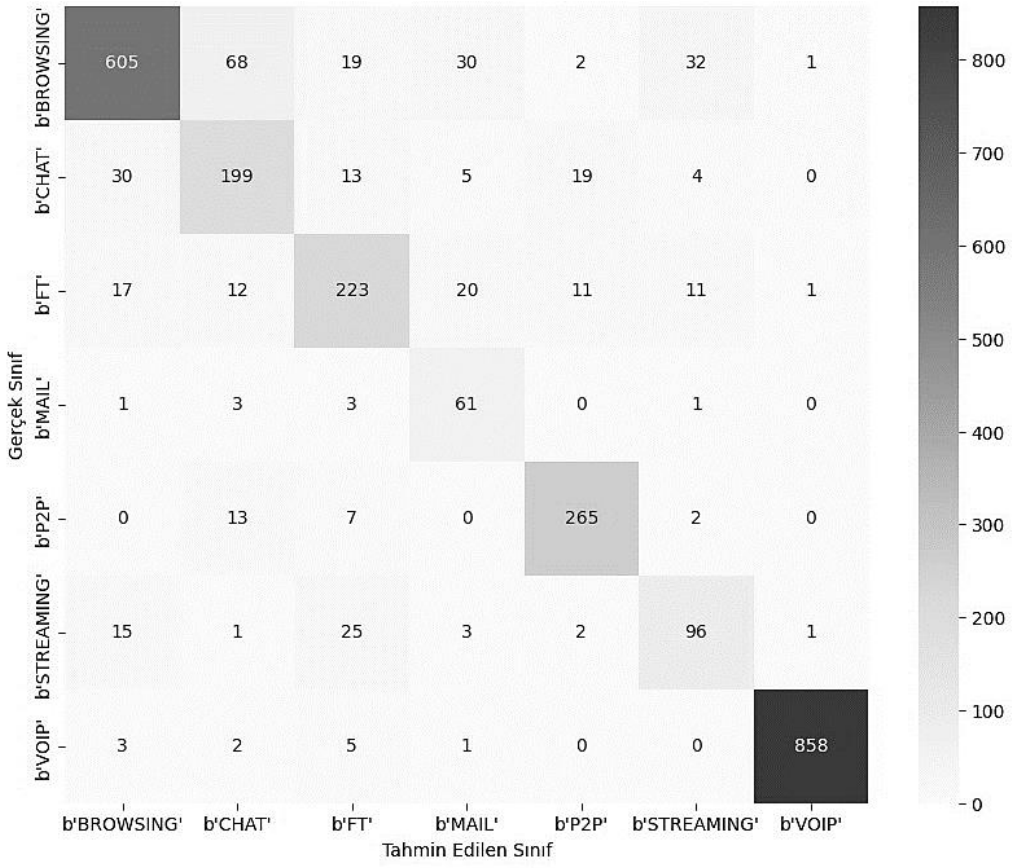
Senaryo B'de dengeleme işlemi ve hiperparametre seçimi sonrasındaki sonuçlar göz önünde bulundurulduğunda SVM ve LR algoritmalarında başarı yakalanamamıştır (ortalama başarı oranı 0,40). 15s, 30s, 60s ve 120s veri setlerinde KNN algoritması kullanıldığında başarı oranlarında azalma eğilimi görülmüştür. Senaryo B'nin en yüksek başarı oranları KNN algoritmasında 0,8379 doğruluk ve 0,7357 f1 skor ile Senaryo B 30s veri setinde elde edilmiştir. Senaryo B veri setlerinde elde edilen başarı oranlarının grafiği Şekil 4.3'te gösterilmiştir.



Şekil 4. 3. KNN Algoritmasının Optimizasyon Sonrası Senaryo B Başarı Oranları

Tüm senaryolar için optimizasyon sonrasında en yüksek başarı oranlarına denetimli makine öğrenmesi algoritmalarından KNN ile ulaşılmıştır. KNN algoritması ile ulaşılan en yüksek başarı oranları, 0,9126 doğruluk ve 0,8591 f1 skor ile Senaryo A2 No-VPN 15s veri setinde elde edilmiştir.

Senaryo A2 No-VPN 15s veri seti ile KNN algoritması kullanılarak elde edilen Şekil 4.4'te yer alan karmaşıklık matrisine göre yapılan analizde sınıf türlerini tahmin etmedeki en belirgin hatanın chat ile browsing sınıfları arasında olduğu görülmektedir. Bu durum, chat sınıfına ait örneklerin browsing olarak ya da browsing sınıfına ait örneklerin chat sınıfı olarak tahmin edildiğini göstermektedir.



Şekil 4. 4. KNN Algoritması ile Senaryo A2 No-VPN 15s Veri Seti Karmaşıklık Matrisi

KNN algoritması ile elde edilen başarı oranları için kullanılan parametreler Tablo 4.9'da yer almaktadır.

Tablo 4. 9. KNN Algoritması için Parametre Değerleri

Parametrenin Adı	**Değeri**
weights	distance
n_neighbors	7
metric	manhattan

Optimizasyon sonrası tekrardan eğitime ve teste tabi tutulan KNN, SVM ve LR algoritmalarının test süreleri Tablo 4.10'da yer almaktadır. Test sürelerinin karşılaştırması, tüm veri setlerindeki kategorizasyon sürelerinin aritmetik ortalaması alınarak gerçekleştirilmiştir. Bu kapsamda test süreleri; KNN için girdi başına 0,000140 saniye olup toplam 0,48 saniye, SVM için girdi başına 0,000906 saniye olup toplam 3,35 saniye ve LR için girdi başına 0,000003 saniye olup toplam 0,01 saniye olarak hesaplanmıştır.

Tablo 4. 10. KNN, SVM ve LR Algoritmalarının Optimizasyon Sonrası Test Süreleri

Algoritma / **Senaryolar**	**Test Edilen Veri Sayısı (adet)**	**KNN**		**SVM**		**LR**		
		Test Süresi (s)	Birim Test Süresi (s/adet)	Test Süresi (s)	Birim Test Süresi (s/adet)	Test Süresi (s)	Birim Test Süresi (s/adet)	
Senaryo A1 15s	5.628	0,43	0,000076	2,56	0,000455	0,01	0,000002	Hiper-parametre optimizasyonu
Senaryo A1 30s	4.397	0,31	0,000071	1,82	0,000414	0,01	0,000002	
Senaryo A1 60s	4.655	0,44	0,000095	1,30	0,000279	0,01	0,000002	
Senaryo A1 120s	3.236	0,60	0,000185	0,96	0,000297	0,01	0,000003	
Senaryo A2 No-VPN 15s	2.690	1,50	0,000558	3,45	0,001283	0,01	0,000004	Dengelenmiş veri setleri ile Hiperparametre optimizasyonu
Senaryo A2 VPN 15s	2.938	0,17	0,000058	1,79	0,000609	0,01	0,000003	
Senaryo A2 No-VPN 30s	2.076	0,10	0,000048	1,45	0,000698	0,01	0,000005	
Senaryo A2 VPN 30s	2.321	0,15	0,000065	1,31	0,000564	0,01	0,000004	
Senaryo A2 No-VPN 60s	2.574	0,11	0,000043	2,94	0,001142	0,01	0,000004	
Senaryo A2 VPN 60s	2.081	0,14	0,000067	1,31	0,000630	0,01	0,000005	
Senaryo A2 No-VPN 120s	1.546	0,06	0,000039	0,74	0,000479	0,01	0,000006	
Senaryo A2 VPN 120s	1.690	0,45	0,000266	1,10	0,000651	0,01	0,000006	
Senaryo B 15s	5.704	1,46	0,000256	12,75	0,002235	0,01	0,000002	
Senaryo B 30s	4.543	0,65	0,000143	7,70	0,001695	0,01	0,000002	
Senaryo B 60s	4.825	0,60	0,000124	6,50	0,001347	0,01	0,000002	
Senaryo B 120s	3.501	0,54	0,000154	6,00	0,001714	0,01	0,000003	
Ortalama Test Süreleri		**0,48**	**0,000140**	**3,35**	**0,000906**	**0,01**	**0,000003**	

Bu sonuçlar yine kategorizasyonu gerçekleştirme süresinin en kısa olması nedeniyle en hızlı çalışan algoritmanın LR, kategorizasyonu gerçekleştirme süresinin en uzun olması nedeniyle en yavaş çalışan algoritmanın ise SVM algoritması olduğunu göstermektedir.

4.2. Topluluk Öğrenmesi Algoritması ile Elde Edilen Sonuçlar

LGBM algoritması kullanılarak orijinal veri seti ile elde edilen sonuçlar Tablo 4.11'de yer almaktadır.

Tablo 4. 11. LGBM Algoritmasının Orijinal Veri Seti ile Başarı Oranları

Algoritma / Senaryolar	LGBM	
	Doğruluk	F1 Skor
Senaryo A1 15s	**0,9087***	**0,9085***
Senaryo A1 30s	0,8924	0,8922
Senaryo A1 60s	0,8901	0,8912
Senaryo A1 120s	0,8927	0,8926
Senaryo A2 No-VPN 15s	***0,9468****	***0,9461****
Senaryo A2 VPN 15s	0,8999	0,8988
Senaryo A2 No-VPN 30s	0,9383	0,9373
Senaryo A2 VPN 30s	0,9130	0,9121
Senaryo A2 No-VPN 60s	0,9169	0,9166
Senaryo A2 VPN 60s	0,8813	0,8785
Senaryo A2 No-VPN 120s	0,9353	0,9341
Senaryo A2 VPN 120s	0,8728	0,8684
Senaryo B 15s	**0,8658***	**0,8657***
Senaryo B 30s	0,8576	0,8562
Senaryo B 60s	0,8335	0,8234
Senaryo B 120s	0,8250	0,8222

* Kalın yazılan değerler, ilgili senaryoda elde edilen en yüksek başarı oranlarıdır.
**Kalın ve italik yazılan değerler, tüm senaryolarda elde edilen en yüksek başarı oranlarıdır.

Senaryo A1 göz önünde bulundurulduğunda 15s, 30s, 60s ve 120s veri setlerinde LGBM algoritması kullanıldığında başarı oranlarının gittikçe azaldığı görülmüştür. Bu senaryoda en yüksek başarı oranları, 0,9087 doğruluk ve 0,9085 f1 skor ile Senaryo A1 15s veri setinde elde edilmiştir.

Senaryo A2 göz önünde bulundurulduğunda 15s, 30s, 60s ve 120s veri setlerinde LGBM algoritması kullanıldığında başarı oranlarının gittikçe azaldığı görülmüştür. Bu senaryoda en yüksek başarı oranları, 0,9468 doğruluk ve 0,9461 f1 skor ile Senaryo A2 No-VPN 15s veri setinde elde edilmiştir.

Senaryo B göz önünde bulundurulduğunda 15s, 30s, 60s ve 120s veri setlerinde LGBM algoritması kullanıldığında başarı oranlarının gittikçe azaldığı görülmüştür. Bu senaryoda en yüksek başarı oranları, 0,8658 doğruluk ve 0,8657 f1 skor ile Senaryo B 15s veri setinde elde edilmiştir.

Tüm senaryolar için Tablo 4.12'deki varsayılan parametre değerlerinin kullanıldığı orijinal veri seti ile LGBM topluluk öğrenme algoritmasında ulaşılan en yüksek başarı oranları, 0,9468 doğruluk ve 0,9461 f1 skor ile Senaryo A2 No-VPN 15s veri setinde elde edilmiştir.

Tablo 4. 12. LGBM Algoritması için Varsayılan Parametre Değerleri

Parametrenin Adı	**Değeri**
colsample_bytree	1.0
learning_rate	0.1
max_depth	-1
min_child_samples	20
min_child_weight	0.001
n_estimators	100
num_leaves	31
reg_alpha	0.0
reg_lambda	0.0
subsample	1.0

Orijinal veri seti ile eğitime ve teste tabi tutulan LGBM algoritmasının test süreleri Tablo 4.13'te yer almaktadır. LGBM algoritmasının test süresi girdi başına 0,000022 saniye olup toplam 0,07 saniye olarak hesaplanmıştır. En başarılı sonuçların elde edildiği topluluk öğrenmesi algoritması LGBM'nin ortalama test süresi, denetimli makine öğrenmesi algoritmalarından en başarılı sonuçları veren KNN algoritmasının Tablo 4.5'te gösterilmiş olan ortalama test süresinden (0,33 saniyeden) daha az çıkmıştır. Burada LGBM algoritmasının kategorizasyon hızının KNN'den yaklaşık beş kat daha yüksek olduğu görülmektedir.

Tablo 4. 13. LGBM Algoritmasının Test Süreleri

Algoritma / Senaryolar	Test Edilen Veri Sayısı (adet)	LGBM		
		Test Süresi (s)	Birim Test Süresi (s/adet)	
Senaryo A1 15s	5.628	0,01	0,000002	Orijinal veri seti ile
Senaryo A1 30s	4.397	0,01	0,000002	
Senaryo A1 60s	4.655	0,03	0,000006	
Senaryo A1 120s	3.236	0,13	0,000040	
Senaryo A2 No-VPN 15s	2.690	0,04	0,000015	
Senaryo A2 VPN 15s	2.938	0,05	0,000017	
Senaryo A2 No-VPN 30s	2.076	0,05	0,000024	
Senaryo A2 VPN 30s	2.321	0,04	0,000017	
Senaryo A2 No-VPN 60s	2.574	0,04	0,000016	
Senaryo A2 VPN 60s	2.081	0,03	0,000014	
Senaryo A2 No-VPN 120s	1.546	0,02	0,000013	
Senaryo A2 VPN 120s	1.690	0,12	0,000071	
Senaryo B 15s	5.704	0,17	0,000030	
Senaryo B 30s	4.543	0,14	0,000031	
Senaryo B 60s	4.825	0,12	0,000025	
Senaryo B 120s	3.501	0,12	0,000034	
Ortalama Test Süreleri		**0,07**	**0,000022**	

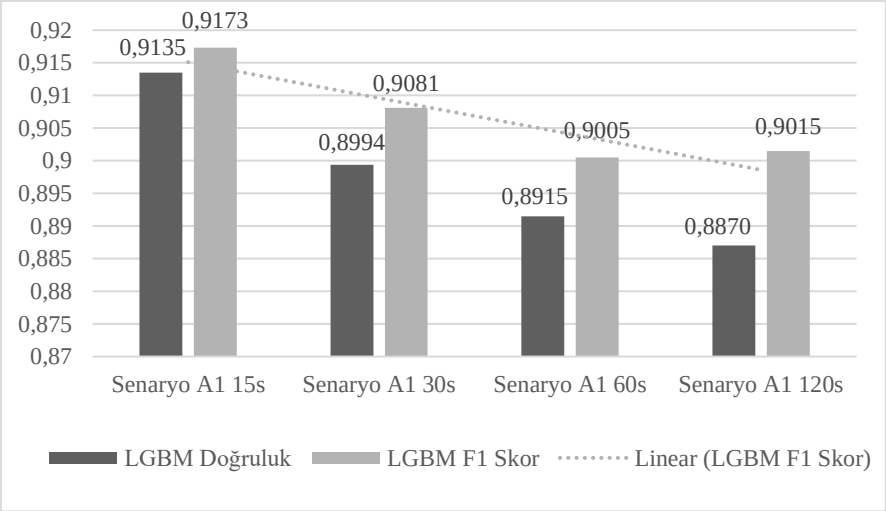
Çalışmada veri dengeleme ve hiperparametre seçimi işlemleri ile yapılan optimizasyon sonrasında veriler tekrar eğitim ve teste tabi tutulmuştur. LGBM algoritması kullanılarak Senaryo A2 ve Senaryo B'deki 12 adet veri seti için SMOTE ile dengeleme işlemi, rastgele arama ve çapraz doğrulama ile hiperparametre seçimi yapıldıktan sonra elde edilen sonuçlar Tablo 4.14'te yer almaktadır. Dengeleme işlemi yapılmayan Senaryo A1'de yer alan 4 adet veri seti için rastgele arama ve çapraz doğrulama ile hiperparametre seçimi yapıldıktan sonra elde edilen sonuçlara da aynı tabloda yer verilmiştir.

Tablo 4. 14. LGBM Algoritmasının Optimizasyon Sonrası Veri Seti ile Başarı Oranları

Algoritma / Senaryolar	LGBM		
	Doğruluk	F1 Skor	
Senaryo A1 15s	**0,9135***	**0,9173***	Hiperparametre seçimi
Senaryo A1 30s	0,8994	0,9081	
Senaryo A1 60s	0,8915	0,9005	
Senaryo A1 120s	0,8870	0,9015	
Senaryo A2 No-VPN 15s	***0,9653****	***0,9459****	Dengelenmiş veri setleri ile Hiperparametre seçimi
Senaryo A2 VPN 15s	0,9267	0,8989	
Senaryo A2 No-VPN 30s	0,9468	0,9302	
Senaryo A2 VPN 30s	0,9299	0,9076	
Senaryo A2 No-VPN 60s	0,9294	0,9169	
Senaryo A2 VPN 60s	0,9228	0,8678	
Senaryo A2 No-VPN 120s	0,9437	0,9271	
Senaryo A2 VPN 120s	0,9149	0,8614	
Senaryo B 15s	**0,9005***	**0,8614***	
Senaryo B 30s	0,8989	0,8561	
Senaryo B 60s	0,8335	0,8234	
Senaryo B 120s	0,8869	0,8253	

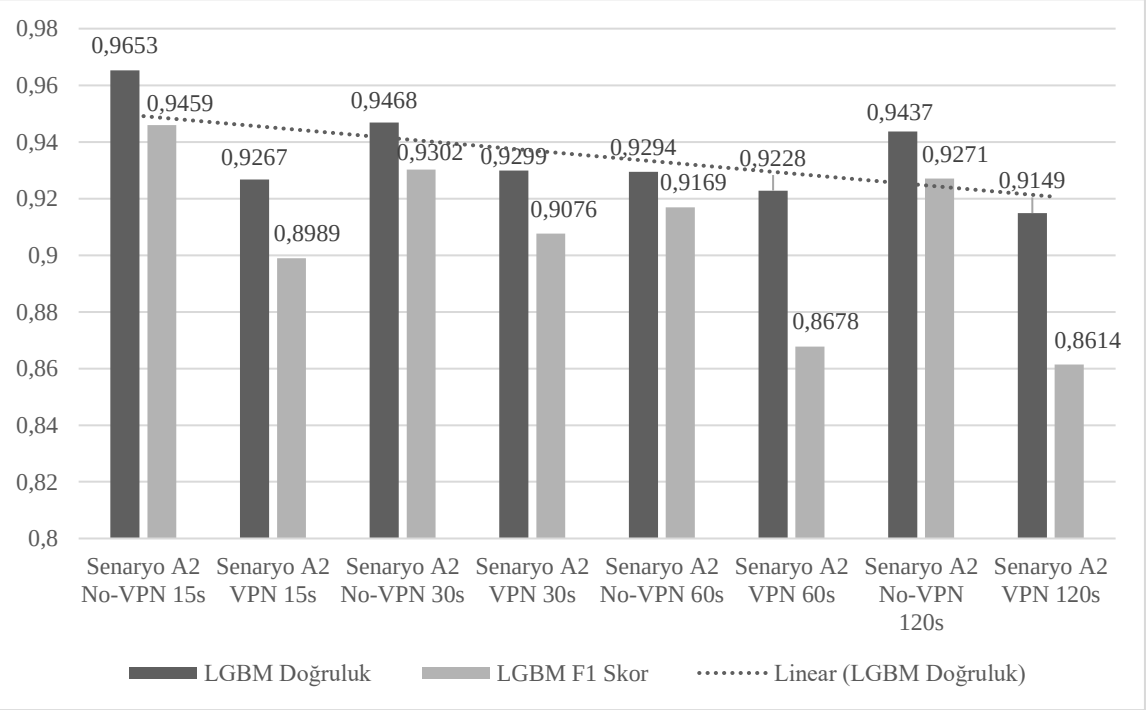
* Kalın yazılan değerler, ilgili senaryoda elde edilen en yüksek başarı oranlarıdır.
**Kalın ve italik yazılan değerler, tüm senaryolarda elde edilen en yüksek başarı oranlarıdır.

Senaryo A1'de hiperparametre seçimi sonrasındaki sonuçlar göz önünde bulundurulduğunda 15s, 30s, 60s ve 120s veri setlerinde LGBM algoritması kullanıldığında başarı oranlarının gittikçe azaldığı görülmüştür. LGBM algoritması ile Senaryo A1 için en yüksek başarı oranları, 0,9135 doğruluk ve 0,9173 f1 skor ile Senaryo A1 15s veri setinde elde edilmiştir. Senaryo A1 veri setlerinde elde edilen başarı oranlarının grafiği Şekil 4.5'te gösterilmiştir.



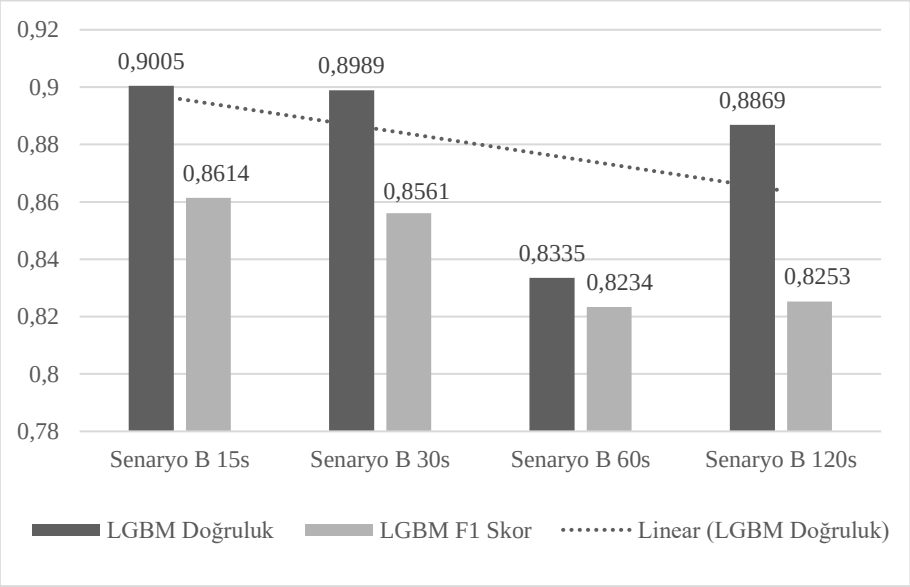
Şekil 4. 5. LGBM Algoritmasının Optimizasyon Sonrası Senaryo A1 Başarı Oranları

Senaryo A2'de dengeleme işlemi ve hiperparametre seçimi sonrasındaki sonuçlar göz önünde bulundurulduğunda 15s, 30s, 60s ve 120s VPN ve No-VPN veri setlerinde LGBM algoritması kullanıldığında başarı oranlarının gittikçe azaldığı görülmüştür. LGBM algoritması ile Senaryo A2 için en yüksek başarı oranları, 0,9653 doğruluk ve 0,9459 f1 skor ile Senaryo A2 No-VPN 15s veri setinde elde edilmiştir. LGBM algoritması ile Senaryo A2 veri setlerinde elde edilen başarı oranlarının grafiği Şekil 4.6'da gösterilmiştir.



Şekil 4. 6. LGBM Algoritmasının Optimizasyon Sonrası Senaryo A2 Başarı Oranları

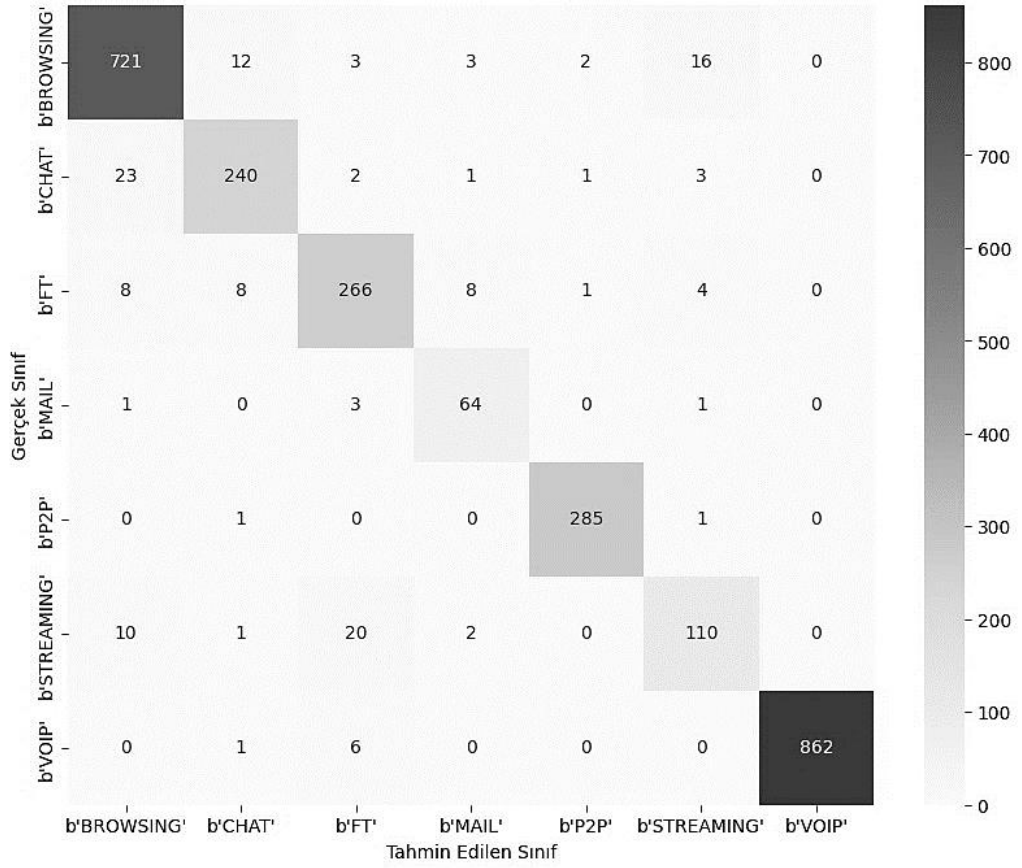
Senaryo B'de dengeleme işlemi ve hiperparametre seçimi sonrasındaki sonuçlar göz önünde bulundurulduğunda 15s, 30s, 60s ve 120s veri setlerinde LGBM algoritması kullanıldığında başarı oranlarının gittikçe azaldığı görülmüştür. LGBM algoritması ile Senaryo B için en yüksek başarı oranları, 0,9005 doğruluk ve 0,8614 f1 skor ile Senaryo B 15s veri setinde elde edilmiştir. LGBM algoritması ile Senaryo B veri setlerinde elde edilen başarı oranlarının grafiği Şekil 4.7'de gösterilmiştir.



Şekil 4. 7. LGBM Algoritmasının Optimizasyon Sonrası Senaryo B Başarı Oranları

Tüm senaryolar için optimizasyon sonrasında LGBM topluluk öğrenmesi algoritmasında ulaşılan en yüksek başarı oranları, 0,9653 doğruluk ve 0,9459 f1 skor ile Senaryo A2 No-VPN 15s veri setinde elde edilmiştir.

Senaryo A2 No-VPN 15s veri seti ile LGBM algoritması kullanılarak elde edilen Şekil 4.8'de yer alan karmaşıklık matrisine göre yapılan analizde sınıf türlerini tahmin etmedeki en belirgin hatanın chat ile browsing sınıfları arasında olduğu görülmektedir. Bu durum, chat sınıfına ait örneklerin browsing olarak ya da browsing sınıfına ait örneklerin chat sınıfı olarak tahmin edildiğini göstermektedir.



Şekil 4. 8. LGBM Algoritması ile Senaryo A2 No-VPN 15s Veri Seti Karmaşıklık Matrisi

LGBM algoritması ile elde edilen başarı oranları için kullanılan parametreler Tablo 4.15'te yer almaktadır.

Tablo 4. 15. LGBM Algoritması için En İyi Parametre Değerleri

Parametrenin Adı	**Değeri**
colsample_bytree	0.8
learning_rate	0.5
max_depth	13
min_child_samples	257
min_child_weight	0.01
n_estimators	604
num_leaves	18
reg_alpha	0
reg_lambda	1
subsample	0.9

Senaryo A1, A2 ve B'de yer alan toplamda 16 adet veri seti ile optimizasyon sonrası tekrardan eğitime ve teste tabi tutulan LGBM algoritmasının test süreleri Tablo 4.16'da yer almaktadır. LGBM algoritmasının test süresi girdi başına 0,000074 saniye olup toplam 0,25 saniye olarak hesaplanmıştır. Optimizasyon sonrasında en başarılı sonuçların elde edildiği topluluk öğrenmesi algoritması LGBM'nin ortalama test süresi, denetimli makine öğrenmesi algoritmalarından en başarılı sonuçları veren KNN algoritmasının Tablo 4.10'da gösterilmiş olan ortalama test süresinden (0,48 saniyeden) daha az çıkmıştır. Burada LGBM algoritmasının kategorizasyon hızının yaklaşık iki kat daha yüksek olduğu görülmektedir.

Tablo 4. 16. LGBM Algoritmasının Optimizasyon Sonrası Test Süreleri

Algoritma / Senaryolar	Test Edilen Veri Sayısı (adet)	LGBM		
		Test Süresi (s)	Birim Test Süresi (s/adet)	
Senaryo A1 15s	5.628	0,06	0,000011	Hiper-parametre optimizasyonu
Senaryo A1 30s	4.397	0,10	0,000023	
Senaryo A1 60s	4.655	0,25	0,000054	
Senaryo A1 120s	3.236	0,34	0,000105	
Senaryo A2 No-VPN 15s	2.690	0,27	0,000100	Dengelenmiş veri setleri ile Hiperparametre optimizasyonu
Senaryo A2 VPN 15s	2.938	0,93	0,000317	
Senaryo A2 No-VPN 30s	2.076	0,03	0,000014	
Senaryo A2 VPN 30s	2.321	0,04	0,000017	
Senaryo A2 No-VPN 60s	2.574	0,03	0,000012	
Senaryo A2 VPN 60s	2.081	0,03	0,000014	
Senaryo A2 No-VPN 120s	1.546	0,02	0,000013	
Senaryo A2 VPN 120s	1.690	0,11	0,000065	
Senaryo B 15s	5.704	0,16	0,000028	
Senaryo B 30s	4.543	0,64	0,000141	
Senaryo B 60s	4.825	0,23	0,000048	
Senaryo B 120s	3.501	0,79	0,000226	
Ortalama Test Süreleri		**0,25**	**0,000074**	

Bu sonuçlar, optimizasyon sonrasında da hem en başarılı hem de en hızlı algoritmanın LGBM topluluk öğrenmesi algoritması olduğunu göstermektedir.

5. SONUÇ, TARTIŞMA VE ÖNERİLER

5.1. Sonuçlar ve Tartışma

Yapılan şifreli trafiğin sınıflandırılması çalışmasında siber güvenlik uzmanlarının ağ üzerindeki tehditleri tespit etme ve saldırıları önleme yeteneklerini artırmak ve sistemlerin daha güvenli hale getirilmesini sağlamak amaçlanmıştır. Bu sınıflandırma ile siber güvenlik uzmanları belirledikleri politikalar kapsamında istenmeyen sınıflara ilişkin trafikleri tespit edip engelleyebileceklerdir. Çalışma kapsamında ağ trafiğinde yaygın olarak kullanılan sınıfları içeren ve gerçek ağ trafiği verilerinden elde edilmiş olan ISCXVPN2016 veri seti kullanılmıştır. Bu veri seti için denetimli makine öğrenmesi algoritmalarından KNN, SVM ve LR ile topluluk öğrenmesi algoritmalarından LGBM algoritması kullanılmıştır. Kullanılan algoritmalar için öncelikle orijinal veri seti ile eğitim ve test gerçekleştirilmiştir. Daha sonra dengesiz dağılımlı veri setleri SMOTE ile dengelenip rastgele arama ve çapraz doğrulama ile parametre optimizasyonu yapılarak tekrar eğitim ve test gerçekleştirilmiştir. Yapılan optimizasyon sonucunda elde edilen sonuçlar incelenmiş ve başarı oranlarının arttığı görülmüştür.

Bu çalışmada her bir senaryo ve ilgili veri setleri için, topluluk öğrenme algoritması LGBM ile elde edilen başarı oranlarının, kullanılan denetimli makine öğrenmesi algoritmalarından KNN, SVM ve LR ile elde edilen başarı oranlarından daha yüksek olduğu görülmüştür. Bu sonuç; Uğurlu vd. [2], Caicedo vd. [6], Obaşı [27], Bagui vd. [32], Zhang vd. [35], Zhou vd. [37], Afuwape vd. [40] ve Bozkır [43] tarafından yapılan çalışmalardaki topluluk öğrenmesi algoritmalarında (genelde XGBoost ve RF ile) makine öğrenmesi algoritmalarına kıyasla daha yüksek başarı oranlarına ulaşıldığı sonucunu desteklemektedir.

En yüksek başarı oranlarına ulaşılan LGBM algoritması ile elde edilen sonuçların birebir karşılaştırması, ISCXVPN2016 veri seti kullanılarak tüm senaryoları ele alan çalışmalarda (Uğurlu vd. [2], Caicedo vd. [6], Draper-gil vd. [8], Majeed vd. [39] ve Bozkır vd. [43]) elde edilen en yüksek başarı oranları ile yapılmış olup sonuçlar Tablo 5.1'de sunulmuştur.

Tablo 5. 1. Yapılan Çalışmanın Diğer Tüm Senaryoları İçeren Çalışmalarla Karşılaştırması

Senaryo	Yazar(lar)	Algoritma	Doğruluk	F1 skor	Kesinlik
Senaryo A1	Uğurlu vd. [2]	XGBoost	%93,04	-	-
	Draper-gil vd. [8]	C4.5	-	-	%90,60
	Majeed vd. [39]	HFL	%86	-	-
	Bozkır vd. [43]	XGBoost	-	%99	-
	Kendi çalışmamız	LGBM	%91,35	%91,73	%91,99
Senaryo A2	Uğurlu vd. [2]	XGBoost	%94,53	-	-
	Caicedo vd. [6]	Bagging	%94,42	-	-
	Draper-gil vd. [8]	C4.5	-	-	%89
	Majeed vd. [39]	HFL	%86	-	-
	Bozkır vd. [43]	XGBoost	-	%99	-
	Kendi çalışmamız	LGBM	%96,53	%94,59	%94,86
Senaryo B	Uğurlu vd. [2]	XGBoost	%86,06	-	-
	Caicedo vd. [6]	Bagging	%86,94	-	-
	Draper-gil vd. [8]	C4.5	-	-	%80,90
	Majeed vd. [39]	HFL	%81	-	-
	Bozkır vd. [43]	XGBoost	-	%99	-
	Kendi çalışmamız	LGBM	%90,05	%86,14	%86,83

Senaryo A1'de en yüksek başarı oranları 15s veri setinde elde edilmiş olup doğruluk %91,35, f1 skor %91,73 ve kesinlik %91,99'dur. Bu sonuç ile Draper-gil vd. [8] tarafından yapılan çalışmadaki %90,6 kesinlik oranı ile karşılaştırıldığında %1,5 daha başarılıdır. Majeed vd. [39] tarafından yapılan çalışmadaki %86 doğruluk oranı ile karşılaştırıldığında %5,5 daha başarılıdır. Ancak Uğurlu vd. [2] tarafından yapılan çalışmadaki %93,04 doğruluk oranından %2 ve Bozkır vd. [43] tarafından yapılan çalışmadaki %99 f1 skoru'ndan %7 daha az başarılıdır.

Senaryo A2'de en yüksek başarı oranları 15s veri setinde elde edilmiş olup doğruluk %96,53, f1 skor %94,59 ve kesinlik %94,86'dır. Bu sonuç ile Draper-gil vd. [8] tarafından yapılan çalışmadaki %89 kesinlik oranı ile karşılaştırıldığında %6 daha başarılıdır. Caicedo vd. [6] tarafından yapılan çalışmadaki %94,42 doğruluk oranı ile karşılaştırıldığında %2 daha başarılıdır. Majeed vd. [39] tarafından yapılan çalışmadaki %86 doğruluk oranı ile karşılaştırıldığında %10 daha başarılıdır. Uğurlu vd. [2] tarafından yapılan çalışmadaki %94,53 doğruluk oranı ile karşılaştırıldığında %2 daha başarılıdır. Ancak Bozkır vd. [43] tarafından yapılan çalışmadaki %99 f1 skoru'ndan %4 daha az başarılıdır.

Senaryo B'de en yüksek başarı oranları da yine 15s veri setinde elde edilmiş olup doğruluk %90,05, f1 skor %86,14 ve kesinlik %86,83'dir. Bu sonuç ile Draper-gil vd. [8] tarafından yapılan çalışmadaki %80,9 kesinlik oranı ile karşılaştırıldığında %6 daha başarılıdır. Caicedo vd. [6] tarafından yapılan çalışmadaki %86,94 doğruluk oranı ile karşılaştırıldığında %3 daha başarılıdır. Majeed vd. [39] tarafından yapılan çalışmadaki %81 doğruluk oranı ile karşılaştırıldığında %9 daha başarılıdır. Uğurlu vd. [2] tarafından yapılan çalışmadaki %86,06 doğruluk oranı ile karşılaştırıldığında %4 daha başarılıdır. Ancak Bozkır vd. [43] tarafından yapılan çalışmadaki %99 f1 skoru'ndan %4 daha az başarılıdır.

Şifreli ağ trafiğinin sınıflandırılmasına ilişkin yapılan bu çalışmada, başarı oranlarının yanı sıra günümüzde neredeyse her alanda önemli bir kriter olan test süreleri de göz önünde bulundurulmuştur. En iyi başarı oranlarının elde edildiği LGBM topluluk öğrenmesi algoritmasının ortalama birim test süresi 0,000074 saniye olup KNN makine öğrenmesi algoritmasının ortalama birim test süresi 0,000140 saniyeden daha az çıkmıştır. Bu, LGBM'nin kategorizasyon işlemini daha kısa sürede yaptığı ve KNN algoritmasından yaklaşık iki kat daha hızlı çalıştığı anlamına gelmektedir. Bu sonuçlar Khatouni ve Heywood [52] tarafından yapılan çalışmanın aksine, şifreli trafiğin sınıflandırılmasında topluluk öğrenmesi algoritmasının makine öğrenmesi algoritmasından daha hızlı çalıştığını göstermektedir.

Ayrıca çalışmanın optimizasyon öncesi ve sonrasındaki başarı oranları ve test süreleri incelendiğinde SVM algoritmasının başarı oranı ve hız açısından etkin kategorizasyon gerçekleştiremediği, LR algoritmasının en hızlı kategorizasyonu sağlamasına karşın başarı oranının yüksek olmadığı, KNN algoritmasının başarı oranının yüksek olduğu ancak hız açısından LR algoritmasından daha geride kaldığı, LGBM algoritmasının ise en yüksek başarı oranlarını verdiği ve hız açısından da etkin kategorizasyon sağlayabildiği görülmüştür. Bu veriler, kategorizasyon hızının kritik olduğu durumlarda LR ve LGBM algoritmalarının tercih edilebileceğine işaret etmektedir.

5.2. Çalışmanın Akademik Katkısı

Yapılan çalışmada hem doğruluk hem de hız açısından yüksek performans gösteren bir LGBM topluluk öğrenmesi modeli geliştirilmiştir. ISCXVPN2016 veri seti kullanılarak yapılan çalışmada, daha önce LGBM topluluk öğrenmesi algoritması ile KNN, SVM ve LR denetimli makine öğrenmesi algoritmalarını kıyaslayan bir çalışmaya rastlanmamıştır.

Ayrıca bir sınıflandırma probleminin etkinliğinden bahsedebilmek için modelin başarısı kadar hızı da önemlidir. Ancak bu açıdan ele alınan çalışmalar görülmemiş olup ilgili algoritmaların test süreleri hesaplanarak sınıflandırma hızı açısından da karşılaştırma yapan bir çalışma gerçekleştirilmiştir.

Kullanılan veri setinin büyüklüğü ve çeşitliliği ise elde edilen sonuçların genelleme ve geçerlilik açısından daha güçlü olmasını sağlamaktadır. Önerilen modelin test edilebilmesi için çalışmada kullanılan veri setinde yer alan tüm senaryolara ilişkin veri setleri analiz edilmiştir. Bu veri seti kullanılarak yapılan diğer çalışmalarda bu kadar detaylı bir karşılaştırmanın yapılmadığı da dikkatimizi çekmiştir.

5.3. Çalışmanın Sektöre Katkısı

Şifreli trafiğin sınıflandırılmasına ilişkin yapılmış olan bu tez çalışmasında önerilen modeli kullanacak siber güvenlik uzmanları, şifreli trafiği daha etkin bir şekilde sınıflandırarak tehditleri yüksek doğruluk ve hızla tespit edebilir, istenmeyen türden trafikleri engelleyebilirler. Güvenlik şirketleri, bu çalışmanın bulgularını kullanarak mevcut tehdit tespiti ve engelleme sistemlerini güçlendirebilirler. Örneğin, yeni nesil güvenlik duvarları veya ağ izleme sistemleri ile önerilen modelin entegrasyonu yapılarak daha etkin ve adaptif güvenlik modülleri oluşturulabilir. Bu ürünlerle, ağ trafiği analiz edilerek istenmeyen türden trafikler sınıflandırılabilir ve tehditlere karşı otomatik önlemler alınabilir.

5.4. Öneriler

Şifreli trafiğin sınıflandırılması, güvenli ağ yönetimi açısından kritik bir konudur. Bu konuda yapılan yapay zeka temelli çalışmalar 2000'li yıllarda yoğunluk kazanmış olup günümüzde de artan önemiyle devam etmektedir.

Gelecek çalışmalarda;

- Farklı veri setleri oluşturulup modele entegre edilerek geliştirilen model daha genellenebilir hale getirilebilir.
- Veri bütünlüğünün korunması için şifreli trafiğin sınıflandırılmasında geliştirilen modelin zaman damgası ile entegrasyonu sağlanabilir.

Bu alandaki çalışmalar geliştirildikçe siber güvenlik uzmanlarının ağ üzerindeki tehditleri tespit etme ve saldırıları önleme yetenekleri artırılacak, sistemsel güvenliğin daha ileri seviyelere ulaşması sağlanacaktır.

KAYNAKÇA

[1] Kleinrock, L., Kahn, R.E., Clark, D.D., Cerf, V.G., Leiner, B.M., Lynch, D.C., Roberts, L.G., Wolff, S. (2005) Internet (The History).

[2] Uğurlu, M., Doğru, İ.A., Arslan, R.S. (2021) A New Classification Method for Encrypted Internet Traffic Using Machine Learning. *Turkish Journal of Electrical Engineering and Computer Sciences*. 25 (9), 2450–2468.

[3] Cengiz, E., Gök, M. (2023) Reinforcement Learning Applications in Cyber Security: A Review. *Sakarya University Journal of Science*. 27 (2), 481–503.

[4] Kurt Pehlivanoğlu, M., Atay, R., Odabaş, D.E. (2019) İki Seviyeli Hibrit Makine Öğrenmesi Yöntemi ile Saldırı Tespiti. *Gazi Journal of Engineering Sciences*. 5 (3), 258–272.

[5] Khorram, T. (2019) Network Anomaly Detection Using Optimized Machine Learning Algorithms.

[6] Caicedo-Muñoz, J.A., Ledezma Espino, A., Corrales, J.C., Rendón, A. (2018) QoS-Classifier for VPN and Non-VPN Traffic Based on Time-Related Features. *Computer Networks*. 144 271–279.

[7] Uğurlu, M. (2020) Şifrelenmiş İnternet Trafiğinin Makine Öğrenmesi Yaklaşımı ile Sınıflandırılması, Yüksek Lisans Tezi, Gazi Üniversitesi / Fen Bilimleri Enstitüsü, 2020.

[8] Draper-Gil, G., Lashkari, A.H., Mamun, M.S.I., Ghorbani, A.A. (2016) Characterization of Encrypted and VPN Traffic Using Time-Related Features. in: ICISSP 2016 - Proceedings of the 2nd International Conference on Information Systems Security and Privacy, SciTePress, pp. 407–414.

[9] Welcome to Colab. Date accessed: 17/05/2024. https://colab.research.google.com/.

[10] Kemp, S. (2024) Internet Use In 2024. Date accessed: 29/04/2024. https://datareportal.com/reports/digital-2024-deep-dive-the-state-of-internet-adoption

[11] Fırlar, T. (2003) Ağ Güvenliği. *Sakarya University Journal of Science*.

[12] Huang, Y.F., Lin, C.B., Chung, C.M., Chen, C.M. (2021) Research on QoS Classification of Network Encrypted Traffic Behavior based on Machine Learning. *Electronics (Switzerland)*. 10 (12).

[13] Frankel, S. (NIST), Hoffman, P. (Virtual Private Network Consortium), Orebaugh, A. (BAH), Park, R. (BAH). (2008) NIST SP 800-113 Guide to SSL VPNs.

[14] Nguyen, N.H. (2021) SSL/TLS Interception Challenge from the Shadow to the Light. *SANS Institute*.

[15] What is an SSL Certificate? Date accessed: 26/03/2024. https://www.digicert.com/what-is-an-ssl-certificate.

[16] Ergönül, D.T., Demir, O. (2022) Real-Time Encrypted Traffic Classification with Deep Learning. *Sakarya University Journal of Science*. 26 (2), 313–332.

[17] Guo, L., Wu, Q., Liu, S., Duan, M., Li, H., Sun, J. (2020) Deep Learning-Based Real-Time VPN Encrypted Traffic Identification Methods. in: J Real Time Image Process, Springer, pp. 103–114.

[18] Frankel, S.E., Hoffman, P., Orebaugh, A.D., Park, R. (2008) Guide to SSL VPNs. *NIST-National Institute of Standards and Technology*.

[19] Bischoff, P. (2023) VPN Encryption Explained: IPsec vs SSL.

[20] Cisco (2008) Chapter: Traffic Classification https://www.cisco.com/c/en/us/td/docs/nsite/enterprise/wan/wan_optimization/wan_opt_sg/chap05.html.

[21] Elmaghraby, R.T., Abdel Aziem, N.M., Sobh, M.A., Bahaa-Eldin, A.M. (2023) Encrypted Network Traffic Classification Based on Machine Learning. *Ain Shams Engineering Journal*.

[22] Internet Assigned Numbers Authority (2024) Service Name and Transport Protocol Port Number Registry. *Www.Iana.Org*.

[23] Azab, A., Khasawneh, M., Alrabaee, S., Choo, K.K.R., Sarsour, M. (2023) Network Traffic Classification: Techniques, Datasets, and Challenges. *Digital Communications and Networks*.

[24] Seddigh, N., Nandy, B., Bennett, D., Ren, Y., Dolgikh, S., Zeidler, C., Knoetze, J., Solana, N.S.M. (2019) A Framework & System for Classification of Encrypted Network Traffic Using Machine Learning.

[25] Muliukha V.A, Laboshin L.U, Lukashin A.A (2020) Analysis and Classification of Encrypted Network Traffic Using Machine Learning.

[26] Lichy, A., Bader, O., Dubin, R., Dvir, A., Hajaj, C. (2022) When a RF Beats a CNN and GRU, Together -- A Comparison of Deep Learning and Classical Machine Learning Approaches for Encrypted Malware Traffic Classification.

[27] Obaşı, T.C. (2020) Encrypted Network Traffic Classification Using Ensemble Learning Techniques.

[28] Elmaghraby, R.T., Abdel Aziem, N.M., Sobh, M.A., Bahaa-Eldin, A.M. (2023) Encrypted network traffic classification based on machine learning. *Ain Shams Engineering Journal*.

[29] Abbas, G., Farooq, U., Singh, P., Khurana, S.S., Singh, P. (2023) Feature Engineering and Ensemble Learning-Based Classification of VPN and Non-VPN-Based Network Traffic over Temporal Features. *SN Computer Science*. 4 (5).

[30] Beşiktaş, C. (2012) İnternet Trafiğinin Gerçek Zamanlı Sınıflandırılması.

[31] Yamansavaşçılar, B., Güvensan, M.A., Yavuz, A.G., Karslıgil, M.E. (2017) Application Identification via Network Traffic Classification.

[32] Bagui, S., Fang, X., Kalaimannan, E., Bagui, S.C., Sheehan, J. (2017) Comparison of Machine-Learning Algorithms for Classification of VPN Network Traffic Flow Using Time-Related Features. *Journal of Cyber Security Technology*. 1 (2), 108–126.

[33] Lotfollahi, M., Jafari Siavoshani, M., Shirali Hossein Zade, R., Saberian, M. (2019) Deep Packet: A Novel Approach for Encrypted Traffic Classification using Deep Learning. *Soft Computing*. 24 (3), 1999–2012.

[34] Shapira, T., Shavitt, Y. (2019) Encrypted Internet Traffic Classification is as Easy as Image Recognition. *IEEE Communications Society*. 680–687.

[35] Zhang, F., Tao, S., Liu, J. (2020) Imbalanced Encrypted Traffic Classification Scheme Using Random Forest.

[36] Cheng, J., He, R., Yuepeng, E., Wu, Y., You, J., Li, T. (2020) Real-Time Encrypted Traffic Classification via Lightweight Neural Networks. in: 2020 IEEE Global Communications Conference, GLOBECOM 2020 - Proceedings, Institute of Electrical and Electronics Engineers Inc.

[37] Zhou, K., Wang, W., Wu, C., Hu, T. (2020) Practical Evaluation of Encrypted Traffic Classification Based on a Combined Method of Entropy Estimation and Neural Networks. *ETRI Journal*. 42 (3), 311–323.

[38] Bu, Z., Zhou, B., Cheng, P., Zhang, K., Ling, Z.H. (2020) Encrypted Network Traffic Classification Using Deep and Parallel Network-in-Network Models. *IEEE Access*. 8 132950–132959.

[39] Majeed, U., Khan, L.U., Seon Hong, C. (2020) Cross-Silo Horizontal Federated Learning for Flow-based Time-Related-Features Oriented Traffic Classification.

[40] Afuwape, A.A., Xu, Y., Anajemba, J.H., Srivastava, G. (2021) Performance Evaluation of Secured Network Traffic Classification using a Machine Learning Approach. *Computer Standards and Interfaces*. 78.

[41] Ismailaj, K., Camelo, M., Latre, S. (2021) When Deep Learning May Not Be The Right Tool for Traffic Classification. IFIP Working Group 6.6 on Network Management, IEEE Communications Society, Institute of Electrical and Electronics Engineers.

[42] Almomani, A. (2022) Classification of Virtual Private Networks Encrypted Traffic Using Ensemble Learning Algorithms. *Egyptian Informatics Journal*. 23 (4), 57–68.

[43] Bozkır, R. (2022) Şifreli Ağ Trafiğinin İçerik Açısından Sınıflandırılması, Yüksek Lisans Tezi, Bursa Uludağ Üniversitesi / Fen Bilimleri Enstitüsü.

[44] Yiğidim, H. (2012) Makine Öğrenme Algoritmalarını Kullanarak Ağ Trafiğinin Sınıflandırılması.

[45] Alshammari, R., Heywood, A.N.Z. (2009) Machine Learning Based Encrypted Traffic Classification-Identifying SSH and Skype. Institute of Electrical and Electronics Engineers., and IEEE Computational Intelligence Society.

[46] Di Mauro, M., Longo, M. (2015) Revealing Encrypted WebRTC Traffic via Machine Learning Tools. in: SECRYPT 2015 - 12th International Conference on Security and Cryptography, Proceedings; Part of 12th International Joint Conference on e-Business and Telecommunications, ICETE 2015, SciTePress, pp. 259–266.

[47] Zhang, Y., Zhao, S., Zhang, J., Ma, X., Huang, F. (2019) STNN: A Novel TLS/SSL Encrypted Traffic Classification System Based on Stereo Transform Neural Network. in: Proceedings of the International Conference on Parallel and Distributed Systems - ICPADS, IEEE Computer Society, pp. 907–910.

[48] Madhusoodhana Chari, S., Srinidhi, H., Somu, T.E. (2019) Network Traffic Classification by Packet Length Signature Extraction. in: 2019 5th IEEE International WIE Conference on Electrical and Computer Engineering, WIECON-ECE 2019 - Proceedings, Institute of Electrical and Electronics Engineers Inc.

[49] Yang, Y., Kang, C., Gou, G., Li, Z., Xiong, G. (2019) TLS/SSL Encrypted Traffic Classification with Autoencoder and Convolutional Neural Network. in: Proceedings - 20th International Conference on High Performance Computing and Communications, 16th International Conference on Smart City and 4th International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2018, Institute of Electrical and Electronics Engineers Inc., pp. 362–369.

[50] Obaidy, F. Al, Momtahen, S., Hossain, Md.F., Mohammadi, F. (2019) Encrypted Traffic Classification Based ML for Identifying Different Social Media Applications. Institute of Electrical and Electronics Engineers.

[51] Aydın, A. (2023) Topluluk Öğrenmesi Kullanılarak Zararlı Alan Adlarının Sınıflandırılması, Yüksek Lisans Tezi, Marmara Üniversitesi / Fen Bilimleri Enstitüsü , 2023.

[52] Khatouni, A.S., Heywood, N.Z. (2019) Integrating Machine Learning with Off-the-Shelf Traffic Flow Features for HTTP/HTTPS Traffic Classification, Institute of Electrical and Electronics Engineers.

[53] Eker, K., Eker, A.G., Mandal, D., Pehlivanoğlu, M.K., Duru, N. (2022) Makine Öğrenmesi Yaklaşımları ile Ağ Trafik Sınıflandırılması. in: Proceedings - 7th International Conference on Computer Science and Engineering, UBMK 2022, Institute of Electrical and Electronics Engineers Inc., pp. 393–397.

[54] Alaca, Y. (2023) Siber Güvenlikte CIC-Darknet2020 Veri Seti Kullanarak VPN/NoVPN ve Tor/NoTor Sınıflandırması: Basit ve Karmaşık Modellerin Kullanımı. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*.

[55] Uğurlu, M., Doğru, İ.A., Arslan, R.S. (2023) Detection and Classification of Darknet Traffic Using Machine Learning Methods. *Journal of the Faculty of Engineering and Architecture of Gazi University*. 38 (3), 1737–1746.

[56] Gül İlgün, E., Sönmez, Y., Dener, M. (2023) DarkWEB Traffic Detection and Classification Using Machine Learning Method. *Gazi Journal of Engineering Sciences*. 9 (4), 126–140.

[57] Müsevitoğlu, H., Öztürk, A., Başünal, F.N. (2023) Detection of Personality Features From Handwriting By Machine Learning Methods. *Gazi Journal of Engineering Sciences*. 9 (2), 200–212.

[58] Papadogiannaki, E., Ioannidis, S. (2021) A Survey on Encrypted Network Traffic Analysis Applications, Techniques and Countermeasures. *ACM Computing Surveys*. 54 (6).

[59] Shams, M. (2020) Ağ Anomalisi Tespitinde Makine Öğrenmesi Algoritmalarının Kullanımı ve Karşılaştırmalı Analizi.

[60] Çelikten, H. (2023) Otomatik Makine Öğrenmesi (AUTOML) Yöntemlerinin Karşılaştırılması.

[61] Bhuyan, M.H., Bhattacharyya, D.K., Kalita, J.K. (2014) Network Anomaly Detection: Methods, Systems and Tools. *IEEE Communications Surveys and Tutorials*. 16 (1), 303–336.

[62] Jo, T. (2021) Machine Learning Foundations: Supervised, Unsupervised and Advanced Learning. Springer International Publishing.

[63] Livari, A. (2022) Anomaly Detection Techniques for Unsupervised Machine Learning.

[64] Nahayo, Y., Arı, S. (2015) Performance of SVM, KNN and NBC Classifiers for Text-Independent Speaker Identification with and Without Modelling Through Merging Models. *Sakarya University Journal of Science*.

[65] Arda, M. (2021) Makine Öğrenmesi (Machine Learning) Eğitimi, https://www.udemy.com/tr/. *Https://Www.Udemy.Com/Tr/*.

[66] Shibah, A., Ahmed, A. (2022) Makine Öğrenme Yöntemleri ile Network Data Analizi.

[67] Armağan, H. (2023) Hyperparameter Optimization in Polynomial Kernel SVM Algorithms and its Application. *Gazi Journal of Engineering Sciences*. 9 (4), 220–229.

[68] Sewell, M. (2011) Ensemble Learning.

[69] Kurt, A. (2021) Ağ Tabanlı Saldırı Tespit Sistemlerinde Topluluk Öğrenme Yöntemlerinin Karşılaştırmalı Performans Analizi.

[70] Hu, Y., Zou, F., Li, L., Yi, P. (2020) Traffic Classification of User Behaviors in Tor, I2P, ZeroNet, Freenet.

[71] Yu, T.J. (2020) GitHub-denistanjingyu/LTA-Mobility-Sensing-Project: Location Information About Commuter Activities is Vital for Planning for Travel Disruptions and Infrastructural Development.

[72] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T. (2017) LightGBM: A Highly Efficient Gradient Boosting Decision Tree. 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

[73] Murphy, K.P. (2012) Machine Learning: A Probabilistic Perspective. The MIT Press.

[74] Scikit-Learn Machine Learning in Python. Date accessed: 17/05/2024. https://scikit-learn.org/stable/.

[75] Zheng, A., Casari, A. (2018) Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists. Almanya: O'Reilly Media.

[76] López, V., Fernández, A., García, S., Palade, V., Herrera, F. (2013) An Insight into Classification with Imbalanced Data: Empirical Results and Current Trends on Using Data Intrinsic Characteristics. *Information Sciences*. 250 113–141.

[77] Brodersen, K.H., Ong, C.S., Stephan, K.E., Buhmann, J.M. (2010) The Balanced Accuracy and Its Posterior Distribution. in: Proceedings - International Conference on Pattern Recognition, pp. 3121–3124.

[78] Chawla, N. V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P. (2002) SMOTE: Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence Research*. 16 321–357.

[79] Yang, L., Shami, A. (2020) On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice. *Neurocomputing*. 415 295–316.

[80] Tran, N., Schneider, J.G., Weber, I., Qin, A.K. (2020) Hyper-Parameter Optimization in Classification: To-Do or Not-To-Do. *Pattern Recognition*. 103.

[81] Bergstra, J., Bengio, Y. (2012) Random Search for Hyper-Parameter Optimization.

[82] Schaffer, C., Edu, S.A.H.C. (1993) Selecting a Classification Method by Cross-Validation.

[83] Uddin, M.F. (2019) Addressing Accuracy Paradox Using Enhanched Weighted Performance Metric in Machine Learning, Institute of Electrical and Electronics Engineers, 2019.

[84] Deng, X., Liu, Q., Deng, Y., Mahadevan, S. (2016) An Improved Method to Construct Basic Probability Assignment Based on the Confusion Matrix for Classification Problem. *Information Sciences*. 340–341 250–261.

[85] Beşiktaş, C., Mantar, H.A. (2011) Traffic Classification of User Behaviors in Tor, I2P, ZeroNet, Freenet (Ağ ve bilgi sempozyumu: Bildiriler kitabı sf 87). TMMOB Elektrik Mühendisleri Odası Ankara Şubesi.