



Ağ Trafikinin Akış Tabanlı Sınıflandırılmasında Akış Sürelerinin Makine Öğrenimi Algoritmalarına Etkisi

Ramazan Bozkır^{1*}, Murtaza Cicioğlu², Cengiz Toğay³, Ali Çalhan⁴

^{1*} Bursa Uludağ Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Bölümü, Bursa, Türkiye, (ORCID: 0000-0002-0032-4270), sr.bozkir@gmail.com

² Bursa Uludağ Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Bursa, Türkiye (ORCID: 0000-0002-5657-7402), murtazacicioğlu@uludag.edu.tr

³ Vestel Elektronik, Siber Güvenlik ve Çözüm Departmanı, İzmir, Türkiye (ORCID: 0000-0001-5739-1784), cengiz.togay@vestel.com.tr

⁴ Düzce Üniversitesi, Mühendislik Fakültesi, Bilgisayar Bölümü, Düzce, Türkiye (ORCID: 0000-0002-5798-3103), alicalhan@duzce.edu.tr

(1st International Conference on Engineering and Applied Natural Sciences ICEANS 2022, May 10-13, 2022)

(DOI: 10.31590/ejosat.1112866)

ATIF/REFERENCE: Bozkır, R., Cicioğlu, M., Toğay, C. & Çalhan, A. (2022). Ağ Trafikinin Akış Tabanlı Sınıflandırılmasında Akış Sürelerinin Makine Öğrenimi Algoritmalarına Etkisi. *Avrupa Bilim ve Teknoloji Dergisi*, (36), 276-283.

Öz

Günümüzde ağ trafiği verilerinin kontrol altında olması önemli bir gerekliliktir. Ağ operasyonlarının başarısı, belirlenen hedeflere yönelik ağ trafiği sınıflandırılmasının doğru ve performanslı bir şekilde gerçekleştirilmesine bağlıdır. Ağ trafiği sınıflandırılmasında sıklıkla istatistiksel bir yaklaşım olan akış tabanlı yöntemler kullanılmaktadır. Bu çalışmada, farklı akış sürelerinde oluşan ağ akışlarının makine öğrenimi algoritmaları üzerindeki etkileri incelenmiştir. AdaBoost, DecisionTree ve RandomForest makine öğrenimi algoritmalarının ağ trafiği sınıflandırılmasında akış tabanlı yöntem ile farklı akış sürelerinde sınıflandırma performansları analiz edilmiştir. Elde edilen sonuçlara göre makine öğrenimi algoritmalarının ağ akışı süresinden önemli ölçüde etkilendikleri tespit edilmiştir.

Anahtar Kelimeler: Ağ trafiği sınıflandırılması, Akış tabanlı yöntem, Makine öğrenmesi.

The Effect of Flow Times on Machine Learning Algorithms in Flow-Based Classification of Network Traffic

Abstract

Today, it is an important requirement to have network traffic data under control. The success of network operations depends on the accurate and performance classification of network traffic for the determined targets. Flow-based methods, which are a statistical approach, are often used in network traffic classification. In this study, the effects of network flows occurring at different flow times on machine learning algorithms are examined. The classification performances of AdaBoost, DecisionTree and RandomForest machine learning algorithms at different flow times have been analyzed with the flow-based method in network traffic classification. According to the results obtained, it has been determined that machine learning algorithms are significantly affected by the network flow time.

Keywords: Network traffic classification, Flow-based method, Machine learning

1. Giriş

Günümüzde internet kullanımının artış göstermesiyle beraber kullanıcılara yönelik geliştirilen uygulamalar da artmaktadır. İnsanların günlük aktivitelerinin birçoğunu internet üzerindeki uygulamalar ile gerçekleştirmektedir. Yapılan istatistiksel çalışmalarda 2022 yılı Ocak ayında dünya nüfusunun %62,5'ini oluşturan 49.5 milyar insan internet kullanmaktadır (Statista, 2022). Dünya genelinde bir kişi, günde ortalama 1 gigabayt (GB) ağ trafiği verisi oluşturmaktadır (World Development Report (WDR) Team, 2021). İnternet kullanıcılarının mevcut internet ağı altyapısını verimli ve güvenli kullanımının sağlanması için ağ trafiği verilerinin kontrol altında olması gerekmektedir. Büyük veriyi oluşturan ağ trafiklerinin, kontrol edilebilmesi için ağ yönetimi ve analizlerinin gerçekleştirilmesi gerekmektedir. Bir ağ trafiğinin yönetiminde hizmet kalitesinin (Quality of Service (QoS)) artırılması, saldırı tespiti (intrusion detection), zararlı yazılım analizi (malware analysis), hata tespiti (fault diagnosis), anormallik tespiti (anomaly detection) gibi operasyonlar yürütülmektedir. Ağ operasyonlarının başarısı, belirlenen hedeflere yönelik ağ trafiği sınıflandırılmasının doğru ve performanslı bir şekilde gerçekleştirilmesine bağlı olmaktadır. Bu ihtiyaçlardan dolayı geçmişten günümüze ağ trafiğinin sınıflandırılması önemli bir araştırma konusu olmaktadır.

Ağ trafiği sınıflandırmasında, geliştirilen çözüm yöntemleri üç farklı yaklaşım ile gruplanabilir. Bu yaklaşımlar bağlantı noktası tabanlı (port-based), yük tabanlı (payload-based) ve makine öğrenimi (machine learning) tabanlı yaklaşımlardır. Bağlantı noktası tabanlı yaklaşım, günümüz internet uygulamalarının dinamik bağlantı noktasına sahip olmasından dolayı uygulanabilir değildir. Yük tabanlı yaklaşım, derin paket incelemesi (DPI) olarak da bilinmektedir. DPI, ağ trafiği uygulamaları için önceden tanımlanmış tasarım kalıplarının ağ paketinin yük verisinde aranmasıyla uygulamalar sınıflandırılmaktadır. Bu yöntemin etkili uygulanabilmesi için tasarım kalıplarının çok iyi tanımlanması ve yük verisinde hızlı arama işleminin yapılması gerekmektedir. Ayrıca, ağ trafiği uygulamalarındaki değişikliklerin sürekli olarak kontrol edilerek tasarım kalıplarının güncellenmesi gerekmektedir. Bununla beraber ağ trafiğinin şifreli olması durumu, DPI yaklaşımının uygulanmasını kısıtlamaktadır. Bu sebeple yük tabanlı yaklaşımın uygulanması oldukça maliyetlidir. Makine öğrenimi tabanlı yaklaşım, istatistiksel tabanlı yaklaşım olarakta bilinmektedir. İstatistiksel tabanlı yaklaşım, ağ akışlarından çıkarılan istatistiksel özellikler ile makine öğrenimi algoritmalarının eğitilmesiyle uygulanmaktadır. Bir ağ akışını, iki cihaz arasında belirlenen akış süresi boyunca gelen ve giden ağ paketleri oluşturmaktadır.

Literatürde makine öğrenimi tekniğinin uygulandığı akış tabanlı sınıflandırma ile ilgili çalışmalar bulunmaktadır. Zhang ve arkadaşları (2013), ağ akışlarının istatistiksel özelliklerinin bir akış torbasında (Bag-of-Flows) toplanmasıyla farklı kombinasyonlarda hesaplanan korelasyon bilgileri ile En Yakın Komşu (Nearest Neighbors) algoritması tabanlı yeni bir yaklaşım önermektedirler. Zhang ve arkadaşları (2015), sıfır gün uygulamalarını sınıflandırabilen, akışlara ait istatistiksel özellikler ve korelasyon bilgileri ile Rastgele Orman (Random Forest) ve K-Ortalamlar (K-means) algoritmalarını içeren RTC (Robust Traffic Classification) olarak adlandırdıkları bir çerçeve önermektedirler. Datta ve arkadaşları (2015), paket tabanlı bir yöntem ile Hangout uygulamasındaki davranışları inceleyerek, uygulamaya özgü çıkardıkları özellikler ile Naif Bayes (Naive

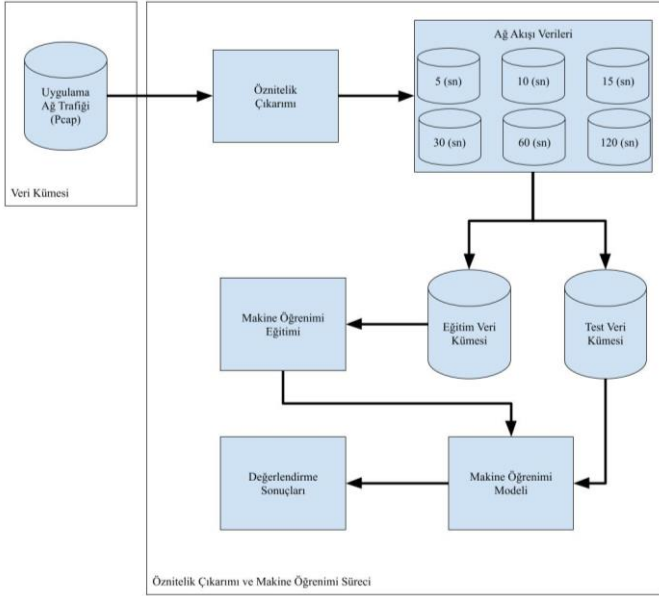
Bayes), adaptif artırma (Adaptive Boosting) ve Karar Ağacı (Decision Tree) algoritmalarının yüksek başarıya ulaştıklarını ve bu yöntemin diğer eşler-arası (peer-to-peer) uygulamaların sınıflandırılmasına uygun olduğunu belirtmektedirler. Draper-Gil ve arkadaşları (2016), ağ trafiği uygulamalarını türlerine göre sınıflandırma çalışmalarında, akış tabanlı istatistiksel özelliklerin 15, 30, 60 ve 120 saniyelik akış sürelerindeki değişimin sınıflandırma performanslarına etkilerini, En Yakın Komşu ve Karar Ağacı algoritmaları ile incelemektedirler. Yamansavascılar ve arkadaşları (2017), ağ trafiği uygulamalarının sınıflandırılmasında, akış tabanlı istatistiksel özelliklerin ki-kare testi ile öznelik seçimi sonrası sınıflandırma performansına etkilerini Karar Ağacı, Rastgele Orman, En Yakın Komşu ve Bayes Ağı algoritmaları ile incelemektedirler. Gómez ve arkadaşları (2017), akış tabanlı istatistiksel özellikleri ile ağ trafiğinin sınıflandırılması için eğitim ve sınıflandırma sürelerinin diğer topluluk öğrenme algoritmalarından daha kısa olan Uyarlanmış-Karar Ağacı Zinciri (T-DTC) algoritmasını geliştirdiler. T-DTC algoritmasının hızlı olmasından dolayı topluluk öğrenme tekniğine olan katkısının, ağ trafiğini gerçek zamanlı sınıflandırma beklentisini artırmak olacağı belirtilmektedir. Cherif ve Kortebi (2019), ağ trafiğinin sınıflandırılmasında akış tabanlı istatistiksel özellikler ile Aşırı Gradyan Artırma (Extreme Gradient Boosting) algoritmasının Naif Bayes, En Yakın Komşu ve Karar Ağacı algoritmalarından daha iyi olduğu belirtmektedirler.

Önceki çalışmalar incelendiğinde Karar Ağacı tabanlı algoritmaların ağ trafiği sınıflandırma problemi için sıklıkla tercih edilmektedir. Akış tabanlı yöntemin uygulandığı çalışmalar arasında sadece bir çalışmanın (Draper-Gil ve ark., 2016), ağ akışı süresini belirttiği ve sınıflandırma performansına olan etkilerinin incelendiği görülmektedir. Bu çalışmada, farklı akış sürelerinde oluşturan ağ akışlarının Karar Ağacı tabanlı makine öğrenimi algoritmaları üzerindeki etkileri analiz edilmektedir.

2. Materyal ve Metot

Bu çalışma kapsamında gerçekleştirilen deney, ağ trafiği verilerinden akış tabanlı özneliklerin çıkarılmasıyla makine öğrenmesi algoritmaları eğitilmektedir. Akış tabanlı özneliklerin çıkarımında akış süreleri 5, 10, 15, 30, 60 ve 120 saniye olan akışlar oluşturulmaktadır. Belirlenen her akış süresi için eğitim ve test veri kümeleri oluşturulmaktadır. Farklı akış sürelerine sahip eğitim kümeleri ile adaptif artırma (AdaBoost), karar ağacı (DecisionTree) ve rastgele orman (RandomForest) makine öğrenimi algoritmaları eğitilmektedir. Algoritmaların değerlendirilmesi için aynı akış sürelerine sahip eğitim ve test veri kümeleri ile doğruluk (Accuracy), F1, kesinlik (precision) ve duyarlılık (recall) skorları hesaplanmaktadır.

Yapılan çalışmanın yönteminde izlenen adımlar Şekil 1'de gösterildiği üzere, veri kümesi, farklı akış sürelerinde öznelik çıkarımı yapılması sonucunda oluşturulan akış verileri ile makine öğrenimi algoritmaları eğitilir ve değerlendirilir.



Şekil 1. Deneyde uygulanan yöntemle ilişkin süreçler.

Figure 1. The processes related to the methods applied in the experiment.

2.1. Veri Kümesi

Bu çalışmada ağ trafiği verilerinin pcap (paket yakalama) dosyası olarak sunulduğu UNB ISCX VPN-nonVPN veri kümesi kullanıldı (Draper-Gil ve ark., 2016). Veri kümesi Skype, Hangout, ICQ gibi uygulamalara ait VPN (sanal özel ağ) kullanımı ve normal ağ trafiklerini içeren toplamda 28 adet ayrı veri grubu barındırmaktadır.

2.2. Öznitelik Çıkarımı

Akış tabanlı öznitelik çıkarımında NFStream aracı kullanıldı. NFStream aracı (Aouini & Pekar, 2022), çevrimiçi ve çevrimdışı gerçek dünya ağ trafiği analizi çalışmaları için tasarlanan performanslı ve esnek bir yapıya sahip Python kütüphanesidir. NFStream ile uygulamaların pcap dosyaları tanımlanan 5, 10, 15, 30, 60 ve 120 saniyelik akış sürelerinde ağ akışı verilerine dönüştürülmektedir. Ağ akışı verilerine ait öznitelikleri, TCP/IP temel özellikleri, temel akış özellikleri ve istatistiksel özellikler oluşturmaktadır. Tablo 1'de veri setimizdeki özniteliklerin isimleri, öznitelik türleri ve sayısı verilmiştir. Öznitelik sütununda "x" ifadesi ilgili özneliğin çarpanı (IP adresi (x2) : kaynak ve hedef IP adresi) olarak verilmektedir. Temel akış özellikleri, akış yönlerini ifade eden kaynak, hedef ve çift yönlü olarak ayrı ayrı öznitelikleri oluşturmaktadır. İstatistiksel özellikler ise akış yönlerine ek olarak ortalama, standart sapma, minimum ve maksimum değerleri ifade eden ayrı ayrı öznitelikler bulunmaktadır.

Tablo 1. Öznitelik Tablosu.

Table 1. Feature Table.

Öznitelik türü	Öznitelik	Öznitelik Sayısı
TCP/IP Temel Özellikler	IP adresi (x2) MAC adresi (x2) OUI adresi (x2) port numarası (x2) protokol ip versiyonu vlan id	11
Temel Akış Özellikleri	id (x2) zaman damgaları (x6) toplam paket sayısı (x3) toplam bayt (x3) akış süresi (x3)	17
İstatistiksel Özellikler	paket boyutu (x12) paket varış süresi (x12) TCP bayrak sayısı (x24)	48

2.3. Makine Öğrenmesi

Yapılan deneyde, makine öğrenimi algoritmalarından AdaBoost, DecisionTree ve RandomForest algoritmaları kullanılmaktadır. DecisionTree algoritması için maksimum derinlik parametresi 10 olarak belirlendi. AdaBoost ve RandomForest algoritmaları DecisionTree algoritmasını kullanan topluluk (ensemble) öğrenme tekniği ile tasarlanan algoritmalarıdır. Topluluk öğrenmesi birden fazla karar ağacı barındırmaktadır. Toplu öğrenme algoritmalarında karar ağacı sayısı n_estimators parametresinin belirlenmesi ile sağlanır. Topluluk öğrenme algoritmaları ile karar ağacı sayısı 300 olarak ayarlandı. AdaBoost algoritması topluluk öğrenme olmasıyla beraber artırma tekniğini içermektedir. Artırma tekniği karar ağaçlarının sırasıyla eğitilerek her bir karar ağacının kendisinden önceki karar ağacından hatalı sınıflandırılan verilerin katsayıları artırılarak bir sonraki karar ağacında kullanılır. Bu işlem belirlenen karar ağacı sayısı kadar tekrarlanır. AdaBoost algoritması için RandomForest algoritmasında olduğu gibi 300 adet karar ağacı belirlendi. Ayrıca AdaBoost algoritmasında önemli bir parametre olan öğrenme oranı (learning rate) 0,15 olarak belirlendi. Öğrenme oranı hata değeri hesaplanırken tahminlerin ne kadarının dikkate alınacağını ifade etmektedir.

Makine öğrenimi algoritmaları farklı akış sürelerinde oluşturulan veri kümelerinin %33 oranında test verileri geri kalanı ise eğitim verileri olarak ayrılmaktadır. Makine öğrenmesi algoritmaları eğitim verileri ile eğitilmekte ve hem eğitim hemde test verileri ile değerlendirme skorları hesaplanmaktadır. Değerlendirme skorlarını doğruluk, F1, duyarlılık ve kesinlik skorları oluşturmaktadır. Değerlendirme skorları Tablo 2'de verilen karmaşıklık matrisi (confusion matrix) gösterimi üzerinden hesaplanmaktadır. Karmaşıklık matrisinde 1 doğru sınıfı, 0 ise yanlış sınıfı temsil etmektedir.

Makine öğrenmesi modellerinin performansı denklem 1 ve 4 arasındaki hesaplamalara göre belirlenmektedir. Doğruluk skoru karmaşıklık matrisinden genel bir skoru ifade eden doğru tahmin sayısının tüm tahmin sayısı oranı ile hesaplanmaktadır. Kesinlik skoru, doğru olarak tahmin edilenlerin gerçekte ne oranda doğru olduğunu belirtmektedir. Duyarlılık skoru, doğru sınıfın ne oranda doğru tahmin edildiğini ifade etmektedir. F1 skoru, duyarlılık ve kesinlik skorlarının harmonik ortalaması ile ifade edilmektedir.

Tablo 2. Karmaşıklık Matrisine Genel Bakış.

Table 2. Overview of the Confusion matrix.

		Gerçek Sınıf	
		1	0
Tahmin Sınıfı	1	Doğru Pozitif (DP)	Yanlış Pozitif (YP)
	0	Yanlış Negatif (YN)	Doğru Negatif (DN)

$$\text{Doğruluk Skoru} = \frac{DP + DN}{(DP + DN + YP + YN)} \quad (1)$$

$$\text{Kesinlik Skoru} = \frac{DP}{(DP + YP)} \quad (2)$$

$$\text{Duyarlılık Skoru} = \frac{DP}{(DP + YN)} \quad (3)$$

$$\text{F1 Skoru} = 2 \times \frac{(\text{Kesinlik} \times \text{Duyarlılık})}{(\text{Kesinlik} + \text{Duyarlılık})} \quad (4)$$

3. Araştırma Sonuçları ve Tartışma

Yapılan çalışma kapsamında, ağ trafiklerinin farklı sürelerde ağ akışlarına dönüştürülmesi sonucunda oluşturulan veri kümeleri eğitim ve test veri kümesi olmak üzere iki gruba ayrıldı. Eğitim veri kümesi makine öğrenmesi algoritmalarıyla eğitildi. Bu eğitim sonucunda elde edilen model test veri kümesi ile değerlendirildi. Bu bölümde, makine öğrenmesi algoritmalarının sonuçları, genel ve uygulama sınıflarına yönelik hesaplanan değerlendirme skorları ile incelenmektedir.

Makine öğrenmesi algoritmalarının eğitim ve test verileri ile 5, 10, 15, 30, 60 ve 120 saniyelik akış sürelerindeki değerlendirme skorları Şekil 2'de gösterilmektedir. Değerlendirme sonuçlarından F1 skoru dikkate alındığında AdaBoost algoritması tüm akış sürelerinde yaklaşık %100'e yakın bir skor ile DecisionTree ve RandomForest algoritmalarından daha iyi sonuç vermektedir. AdaBoost algoritması başarısı için akış sürelerindeki değişimin bir etkisinin olmadığı görülmektedir. DecisionTree algoritması için F1 skoru dikkate alındığında en iyi sonucun yaklaşık %73 ile 15 saniyelik akış süresinde en kötü sonucun ise yaklaşık %69 ile 120 saniyelik akış süresinde olduğu görülmektedir. RandomForest algoritmasında F1 skoru dikkate alındığında en iyi skora yaklaşık %87 ile 5 saniyelik akış süresinde, en kötü skora ise yaklaşık %80 ile 60 saniyelik akış süresinde ulaşmaktadır. Elde edilen sonuçlara göre, akış süresinin AdaBoost makine öğrenmesi algoritmasına etkisi olmadığını ancak diğer taraftan RandomForest ve DecisionTree algoritmaları için farklılıklar barındırdığı gözlenmiştir.

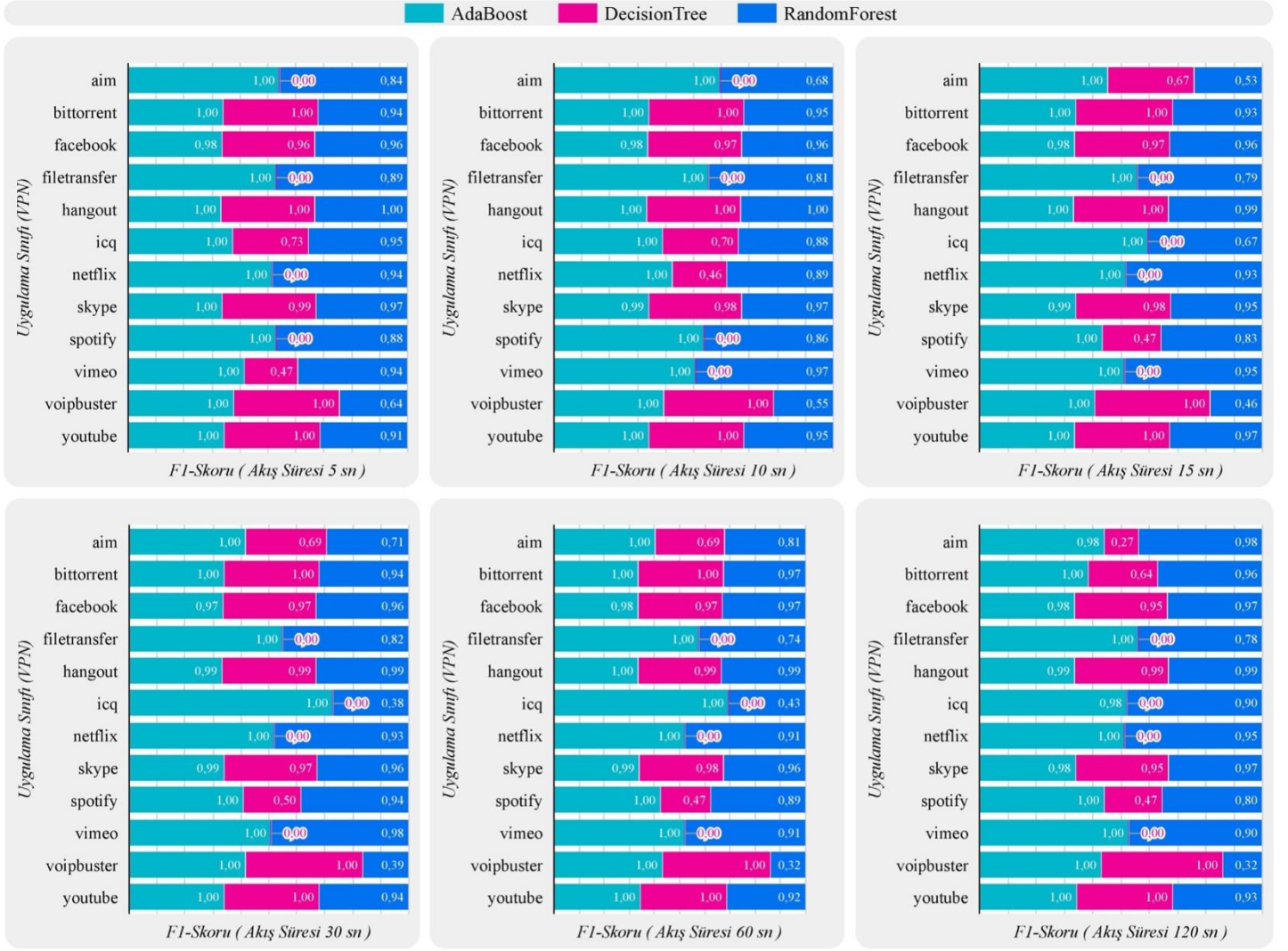


Şekil 2. Farklı akış sürelerinde makine öğrenimi algoritmalarının değerlendirme skorları.

Figure 2. Evaluation scores of machine learning algorithms at different flow times.

Farklı akış sürelerinde VPN kullanımını içeren ağ trafiği uygulama sınıfları için F1 skorları Şekil 3'te gösterilmektedir. AdaBoost algoritması tüm akış sürelerinde iyi skorlara ulaşmaktadır. Ancak, akış süresinin artmasıyla bazı uygulamalarda F1 skoru azalmaktadır. Örneğin, 120 saniyelik akış süresinde AIM, ICQ ve Skype uygulamaları 5 saniyelik akış süresine kıyasla %2 oranında F1 skoru azalmaktadır. DecisionTree algoritması, tüm akış sürelerinde YouTube, VoipBuster, Skype ve Facebook uygulamalarında ortalama olarak yaklaşık %100 oranında F1 skoruna ulaşmıştır. BitTorrent uygulamasında ise sadece 120 saniyelik akış süresinde %64, diğer akış sürelerinde ise %100 oranında F1 skoruna ulaşmaktadır. DecisionTree algoritması, farklı akış

sürelerinde AIM, ICQ, FileTransfer, Netflix ve Vimeo uygulamalarında başarısız olduğu gözlenmiştir. Bu uygulamalar DecisionTree algoritmasında akış süresinden en fazla etkilenen uygulamalar olduğu görülmektedir. Random Forest algoritması AIM ve ICQ uygulamaları dışında diğer tüm uygulamalarda %3 oranında tolerans ile incelendiğinde genelde akış süresinin artmasıyla F1 skorunda düşüş görülmektedir. AIM ve ICQ uygulamaları için ise sırasıyla 5 ve 120 saniyelik akış sürelerinde en iyi skora ulaşılmaktadır. Ancak, AIM ve ICQ uygulamaları 10, 15, 30 ve 60 saniyelik akış sürelerinde önce F1 skoru azalmakta daha sonra yükselmektedir. Elde edilen sonuçlar incelendiği en etkili algoritmanın AdaBoost olduğu gözlenmiştir.

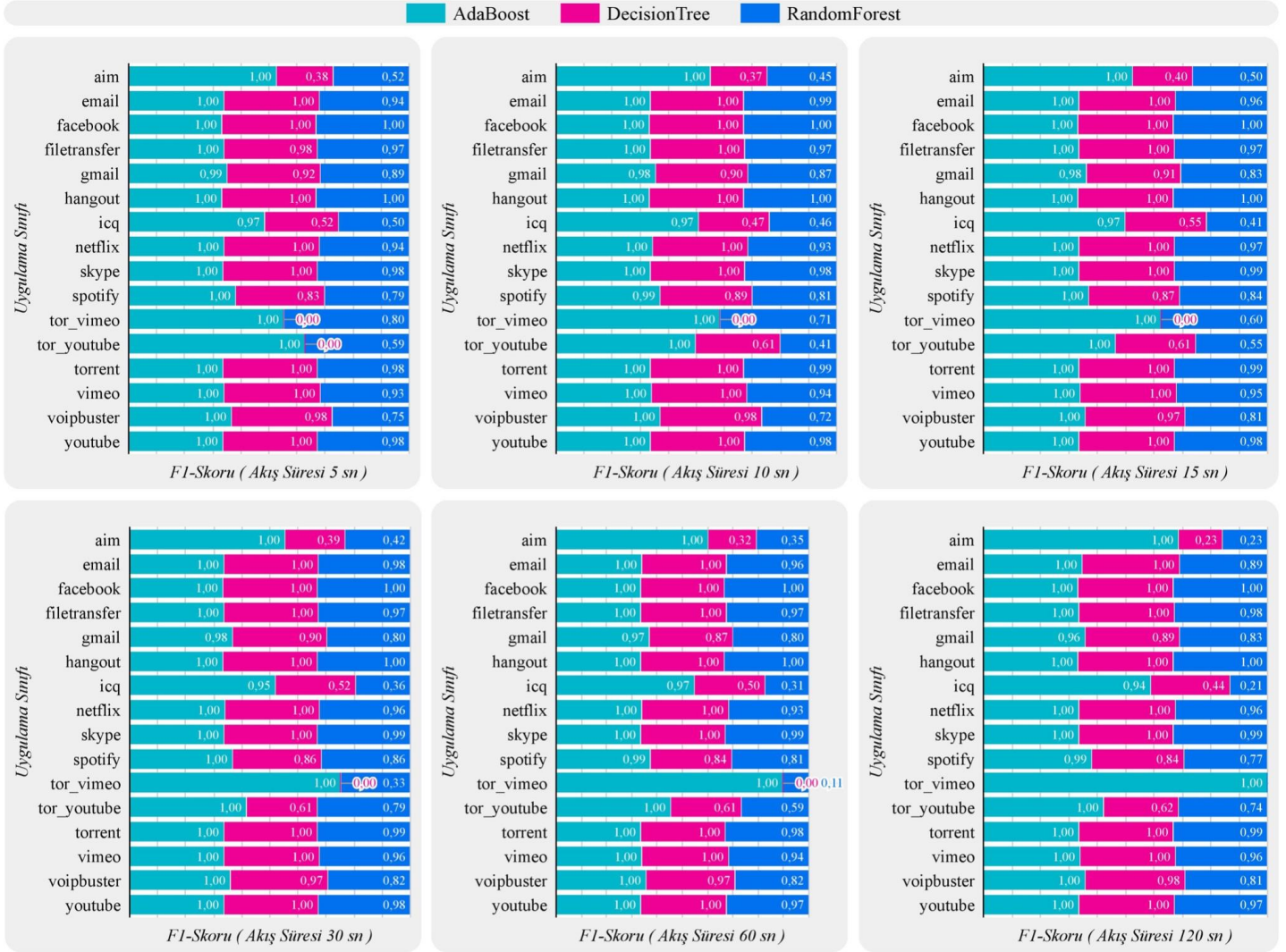


Şekil 3. Farklı akış sürelerinde VPN kullanımını içeren ağ trafiklerine ait uygulama sınıflarının F1 skorları.

Figure 3. F1 scores of application classes of network traffic involving VPN usage at different flow times.

Normal trafiğe sahip uygulama sınıflarının farklı akış sürelerine ilişkin F1 skorları Şekil 4'te gösterilmektedir. AdaBoost algoritması, tüm akış sürelerinde Gmail ve ICQ uygulamaları haricinde %100 oranın F1 skoruna ulaşmaktadır. Gmail ve ICQ uygulamalarında ise en iyi F1 skoruna 5 saniyelik akış süresinde ulaşmaktadır. DecisionTree algoritması, Tor üzerinden gerçekleştirilen Vimeo uygulaması için tüm akış sürelerinde, YouTube uygulaması için 10 ve 15 saniyelik akış süreleri haricindeki diğer tüm akış sürelerinde F1

skorlarının başarısız olduğu görülmektedir. Diğer uygulamalar için DecisionTree algoritması en iyi skora 15 saniyelik akış süresinde ulaşmaktadır. RandomForest algoritması Tor üzerinden gerçekleştirilen YouTube uygulaması dışında en iyi skorlara 5 ve 15 saniyelik akış sürelerinde ulaştığı görülmektedir. Tor üzerinde yapılan YouTube ağ trafiğinde ise 30 saniyelik akış süresinde RandomForest algoritması %79 oranında F1 skoru ile en iyi sonuca ulaşmaktadır.



Şekil 4. Farklı akış sürelerinde normal kullanımı içeren ağ trafiği uygulama sınıflarının F1 skorları.

Figure 4. F1 scores of application classes of network traffic involving normal usage at different flow times.

4. Sonuç

Yapılan çalışma ile AdaBoost, DecisionTree ve RandomForest algoritmalarının ağ trafiği sınıflandırılmasında kullanılan istatistiksel bir yaklaşım olan akış tabanlı yöntem ile 5, 10, 15, 30, 60 ve 120 saniyelik akış sürelerinde sınıflandırma performansları değerlendirildi. Değerlendirme sonuçlarında AdaBoost algoritması tüm akış sürelerinde %100'e yakın bir F1 skoruna ulaştığı tespit edildi. RandomForest algoritması için en iyi F1 skoruna 5 saniyelik akış süresinde ulaşmasıyla beraber akış süresinin artış göstermesinden olumsuz etkilendiği gözlemlendi. DecisionTree algoritması ise 15 saniyelik akış süresinde en iyi F1 skoruna ulaştığı ve diğer akış sürelerinde performansının düştüğü görüldü. Bunun yanında uygulama sınıflarına yönelik hesaplanan değerlendirme skorlarında akış süresindeki değişiklik uygulamalar üzerinde de farklı etkilere yol açmaktadır.

Sonuç olarak, algoritmaların ağ akışı süresinden önemli ölçüde etkilendikleri görülmektedir. Akış sürelerinin algoritmaların performansa olan etkileri, algoritmaların yapılarına göre değişiklik göstermektedir. Bu sebeple ağ trafiği sınıflandırılmasında tercih edilen algoritma için farklı akış sürelerinde değerlendirilmeleri gerekmektedir.

Kaynakça

Aouini, Z., & Pekar, A. (2022). NFStream. *Computer Networks*, 204, 108719. <https://doi.org/10.1016/j.comnet.2021.108719>

Cherif, I. L., & Kortebi, A. (2019). On using extreme gradient boosting (XGBoost) machine learning algorithm for home network traffic classification. *2019 Wireless Days (WD)*. <https://doi.org/10.1109/wd.2019.8734193>

Datta, J., Kataria, N., & Hubballi, N. (2015). Network traffic classification in encrypted environment: A case study of Google hangout. *2015 Twenty First National Conference on Communications (NCC)*. <https://doi.org/10.1109/ncc.2015.7084879>

Draper-Gil, G., Lashkari, A. H., Mamun, M. S., & A. Ghorbani, A. (2016). Characterization of encrypted and VPN traffic using time-related features. *Proceedings of the 2nd International Conference on Information Systems Security and Privacy*. <https://doi.org/10.5220/0005740704070414>

Gómez, S. E., Martínez, B. C., Sánchez-Esguevillas, A. J., & Hernández Callejo, L. (2017). Ensemble network traffic classification: Algorithm comparison and novel ensemble scheme proposal. *Computer Networks*, 127, 68-80. <https://doi.org/10.1016/j.comnet.2017.07.018>

- Statista. (2022, April). *Global digital population as of April 2022* (in billions).
<https://www.statista.com/statistics/617136/digital-population-worldwide/>
- World Development Report (WDR) Team. (2021). *How big are global data flows?*
<https://wdr2021.worldbank.org/stories/crossing-borders/>
- Yamansavascular, B., Guvensan, M. A., Yavuz, A. G., & Karsligil, M. E. (2017). Application identification via network traffic classification. *2017 International Conference on Computing, Networking and Communications (ICNC)*.
<https://doi.org/10.1109/icnc.2017.7876241>
- Zhang, J., Chen, X., Xiang, Y., Zhou, W., & Wu, J. (2015). Robust network traffic classification. *IEEE/ACM Transactions on Networking*, 23(4), 1257-1270.
<https://doi.org/10.1109/tnet.2014.2320577>
- Zhang, J., Xiang, Y., Wang, Y., Zhou, W., Xiang, Y., & Guan, Y. (2013). Network traffic classification using correlation information. *IEEE Transactions on Parallel and Distributed Systems*, 24(1), 104-117.
<https://doi.org/10.1109/tpds.2012.98>