JOURNAL of
MATERIALS and MECHATRONICS:A

e-ISSN 2717-8811
JournalMM, 2022, 3(1), 50-62
https://doi.org/10.55546/jmm.995091

Araştırma Makalesi / Research Article

MQTT Trafiğinde DoS Saldırılarının Makine Öğrenmesi ile Sınıflandırılması ve Modelin SHAP ile Yorumlanması

Ali Cihat KELLE[1]*, Hüseyin YÜCE[2]

[1] Marmara Üniversitesi, Fen Bilimleri Enstitüsü, Bilgi Güvenliği Mühendisliği, İstanbul, Türkiye,
ORCID ID: https://orcid.org/0000-0002-7543-5717, alicihatkelle@gmail.com
[2] Marmara Üniversitesi, Fen Bilimleri Enstitüsü, Siber Güvenlik Anabilim Dalı, İstanbul, Türkiye,
ORCID ID: https://orcid.org/0000-0001-5525-7733, huseyin@marmara.edu.tr

Geliş/ Received: 14.09.2021; **Kabul / Accepted:** 14.03.2022

ÖZET: MQTT (Message Queuing Telemetry Transport), nesnelerin interneti için tasarlanmış, uygulama katmanında çalışan bir haberleşme protokolüdür. MQTT protokolünde sensörler, verileri sunucu ile paylaşırlar, sunucular konulara abone olan cihazlara bu verileri iletirler. Bu çalışmada, bir MQTT trafiğinden elde edilmiş, içinde saldırı trafiği bulunan ve oldukça yeni bir veri seti olan MQTTset içindeki hizmet reddi saldırıları (DoS) makine öğrenmesi ile sınıflandırılmıştır. Saldırının sınıflandırılmasında 3 farklı makine öğrenmesi algoritmasından faydalanılmıştır. En iyi sınıflandırmayı yapan makine öğrenmesi modeli üzerinde analizler yapılmıştır. Model üzerindeki araştırmalarımızın amacı, büyük boyutlu veriler ve karmaşık ağ paketleri üzerinden anlaşılabilir yorumlar çıkarmaktır. Oluşturulan modelin analizinde SHAP kullanılmıştır. SHAP, hesaplamalarında oyun teorisi yaklaşımını benimsemiştir ve basit anlamda bir oyuncunun oyuna katkısını ölçmektedir. SHAP ile hangi öznitelliklerin ve hangi verilerin hizmet reddi saldırısının sınıflandırılmasına ne yönde etki ettiği araştırılarak, oluşturulan makine öğrenmesi modelinden anlaşılabilir yorumlar çıkarılmıştır.

Anahtar Kelimeler: MQTT, MQTTset, XGBoost, SHAP, DoS.

*Sorumlu yazar / Corresponding author: alicihatkelle@gmail.com

Bu makaleye atıf yapmak için /To cite this article

Kelle, A. C., Yüce, H. (2022). MQTT Trafiğinde DoS Saldırılarının Makine Öğrenmesi ile Sınıflandırılması ve Modelin SHAP ile Yorumlanması. Journal of Materials and Mechatronics: A (JournalMM), 3(1), 50-62.

Classification of DoS Attacks in MQTT Network with Machine Learning and Interpretation of The Model with SHAP

ABSTRACT: MQTT (Message Queuing Telemetry Transport) is an application layer communication protocol which designed for the Internet of Things. In the MQTT protocol, the sensors share the data with the server and the servers transmit this data to the devices that subscribe to the topics. In our study, denial-of-service attacks (DoS) in MQTTset, which is a relatively new data set obtained from an MQTT traffic and containing attack traffic, has been classified by machine learning. Three different algorithms of machine learning were used to classify the attack. Analyzes were made on the machine learning model that made the best classification. The purpose of our research on the model is to extract understandable interpretations from large data and complex network packets. SHAP was used in the analysis of the created model. SHAP takes the game theory approach in its calculations and simply measures a player's contribution to the game. By investigating which features and which data affect the classification of denial-of-service attack with SHAP, understandable comments were extracted from the created machine learning model.

Keywords: MQTT, MQTTset, XGBoost, SHAP, DoS.

1. GİRİŞ

Endüstri 4.0 ile birlikte yaygınlaşan nesnelerin internetinde teknolojik gelişim, uygulama katmanında kullanılan haberleşme protokolleri üzerinden gözlemlenebilmektedir. Sensör verilerinin işlenmek üzere iletilmesi bu protokoller üzerinden gerçekleşmektedir. Kullanım basitliği ve anlaşılması açısından MQTT (Message Queuing Telemetry Transport) protokolü diğer protokollere göre birçok artı yöne sahiptir ve kullanımı giderek yaygınlaşmaktadır. MQTT protokolünde sensörler, verileri sunucu ile paylaşırlar, sunucular konulara abone olan cihazlara bu verileri iletirler. MQTT protokolü küçük ev sistemlerinden akıllı şehir sistemlerine kadar büyük ve küçük ölçekte sistemler için kullanılabilir ve bu yüzden bilgi güvenliğinin sağlanması elzemdir.

Çalışmamızda MQTT protokolüne uygulanan hizmet reddi saldırısı makine öğrenmesi algoritmaları ile sınıflandırılmış ve sınıflandırma, makine öğrenmesi modeli üzerinden analiz edilerek, sonucun yorumlanabilir hale gelmesi sağlanmıştır. Makine öğrenmesi modelinde, MQTTset veri seti kullanılmıştır. MQTTset, MQTT ağı için özelleşmiş geniş kapsamlı güncel bir veri seti olmakla birlikte, içerisinde 5 farklı tipte saldırı trafiğini ve normal MQTT trafiğini bulundurmaktadır.

Trafiğin sınıflandırılmasında, en iyi sonuç XGBoost algoritması ile elde edilmiştir. XGBoost (eXtreme gradient boosting), karar ağacı temelli ve hesaplamalarında eğim artırma yöntemini kullanan yenilikçi bir makine öğrenmesi algoritmasıdır. XGBoost, yarışmalarda elde ettiği üstün başarılar ve hızlı işlem kabiliyeti sayesinde birçok veri bilimcinin dikkatini çekmiştir.

Makine öğrenmesi çalışmalarında yeni trend, modelin açıklanabilirliğinin araştırılmasıdır. Bir makine öğrenmesi modelinin, bir kelebeği böcek sınıfında değerlendirirken, kelebeğin bacak sayısını hangi oranda dikkate aldığının bilgisi modelin yorumlanabilirliğine bir örnektir. XGBoost ve MQTTset kullanılarak oluşturduğumuz model SHAP (SHapley Additive Explanations) ile analiz edilmiştir. SHAP, hesaplamalarında oyun teorisi yaklaşımını benimsemiştir ve basit anlamda bir oyuncunun oyuna katkısını ölçmektedir. SHAP ile özelliklerin ve verilerin sınıflandırmaya etkisi araştırılarak, oluşturulan makine öğrenmesi modelinden anlaşılabilir yorumlar çıkarılmıştır.

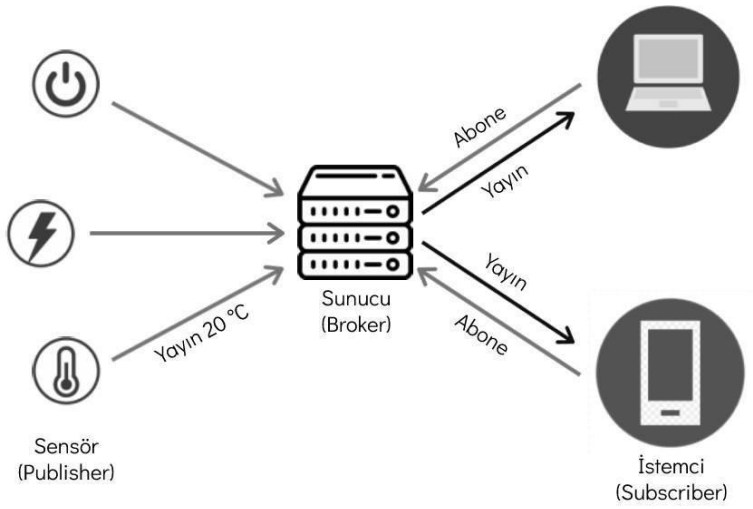
2. MATERYAL VE YÖNTEM

2.1 MQTT

MQTT, ağların ve cihazların birbiri ile haberleşmesini sağlayan açık kodlu bir mesajlaşma protokolüdür. MQTT, uygulama katmanında çalışan, nesnelerin interneti için uygulanabilir, ağ gereksinimleri düşük ve kurulumu kolay bir protokoldür. İlk olarak IBM tarafından 1999 yılında geliştirilmiş ve 2014 yılında OASIS tarafından standartlaştırılmıştır. Protokolün son versiyonu olan MQTT 5.0, OASIS tarafından 2019 yılında kabul edilmiştir.

MQTT protokolü, belli bir ağda farklı cihazlardan gelen mesajları bir merkezde toplamak ve bu mesajları ilgili cihazlara yönlendirmek üzere tasarlanmıştır (Özdoğan, 2020). MQTT sisteminde yayıncı (publisher), sunucu (broker) ve abone (subscriber) rolleri mevcuttur. Nesnelerin interneti için uygulanan bir MQTT protokolü düşünüldüğünde, sensörler yayıncı rolüne sahiptirler ve mesajları konular (topic) ile ilişkilendirerek, merkezi konumda yer alan sunucuya aktarmaktadırlar. MQTT sunucusu da gelen mesajları, ilgili konulara göre abonelere dağıtmaktadır. (Tantitharanukul ve ark., 2017).

MQTT gibi uygulama katmanında çalışan en yaygın protokoller: CoAP, HTTP, XMPP, DDS, AMQP, SMQTT, Restful protokolleridir (Saritha ve Sarasvathi, 2017). MQTT protokolü, diğer uygulama katmanı protokollere göre yaygın kullanım oranına sahiptir (Naik, 2017). MQTT protokolünün yaygın kullanım oranına sahip olmasında, basit ve hafif oluşu, işlem gücü gereksinimi ve ağ bant genişliğini minimal düzeyde tutması büyük rol oynamıştır (Nebbione ve Calzarossa, 2020).



Şekil 1. MQTT mimarisi ve mesajların iletimi

MQTT paketleri belli IP adresleri üzerinden, TCP kullanılarak iletilir. MQTT protokolü, kullanıcılara QoS0, QoS1 ve QoS2 olmak üzere 3 farklı hizmet kalitesi sunmaktadır. QoS0 seviyesinde bir mesaj, bir kez gönderilir ve mesajın karşıya ulaşıp ulaşmadığı kontrol edilmez. QoS1, bir mesajın en az bir kez teslim edildiğinin garanti edildiği hizmet kalite seviyesidir. Mesajın alıcıya iletildiğinin onayı alınana kadar, mesaj gönderici tarafından saklanır. Zaman aşımına bağlı olarak, onay mesajı gönderici tarafına ulaşmadığında saklanan mesaj tekrar alıcıya gönderilir (Gündoğan ve ark., 2018). En yüksek hizmet kalitesi QoS2 seviyesindedir. Bu seviyede mesaj kaybının ve mesaj tekrarının önlenmesi amaçlanmıştır. Mesajın tek seferde alıcıya gönderilmesi ve bir kez alınması garanti edilir.

2.2 MQTTset

MQTTset veri seti, Kasım 2020'de araştırmacılara sunulmuştur ve MQTT 3.1.1 versiyonuna ait ağ trafiğinden elde edilmiş bilgileri içermektedir. Trafikte SSL/TLS gibi şifreleme teknikleri yer almamaktadır. MQTTset veri setinde, normal ağ trafiği ile birlikte çeşitli saldırılara ait ağ trafiği de yer almaktadır. Paylaşılan MQTTset makalesinde, daha önce yayınlanmış olan MQTT trafiğine ait başka veri setleri hakkında bilgiler verilmiş, bu veri setlerinin MQTTset kadar kapsamlı olmadığı anlatılmıştır (Vaccari ve ark., 2020). MQTTset oluşturulurken, IoT-Flock aracı kullanılmıştır. IoT-Flock, nesnelerin interneti için trafik oluşturmaya yarayan açık kaynak kodlu bir araçtır.

MQTTset, 2 odaya sahip akıllı bir ev sistemine ait senaryo için hazırlanmıştır ve toplamda 10 farklı sensöre ait 1 haftalık veriyi içermektedir. Bu sensörler: sıcaklık, ışık yoğunluğu, nem, gaz, hareket, duman, kapı açma/kapama ve fan durumu hakkında bilgi veren sensörlerdir. Saldırganın MQTT mimarisinde doğrudan sunucuya (broker) bağlı olduğu varsayılır (Vaccari ve ark., 2020). Veri setinde yer alan 5 farklı saldırı çeşidi: bruteforce, malformed data, flooding, slowite ve dos saldırılarıdır. Bu saldırılar oluşturulurken çeşitli araçlardan faydalanılmıştır. Saldırı verileri, normal MQTT trafiğine ait verilerle entegre edilerek veri seti oluşturulmuştur.

MQTTset veri setinde ham PCAP dosyalarından verilere ait özellikler çıkarılırken bazı elemeler yapılmıştır. En önemli değişiklikler: kimlik doğrulama, bağlantı süreleri, kaynak ve hedefe ait adreslerin silinmiş olmasıdır (Vaccari ve ark., 2020).

MQTTset içinde dengelenmemiş ve dengeli olmak üzere 2 farklı veri seti bulunmaktadır. Yazarların ilk oluşturdukları veri setinde saldırı trafiği, toplam trafiğin %1'i kadardır. Normal trafiğin sınıflandırmalar yapılırken sonucu domine ettiği düşünülerek, veri setinin dengelenmesi yazarlar tarafından uygun görülmüştür. Yeni veri setinde saldırı trafiğinin sayısı artırılarak dengeli bir veri seti oluşturulmak amaçlanmıştır. Dengelenmiş versiyonda saldırıların karakteristiğinin daha iyi anlaşılacağı düşünülmüştür. Çalışmamızda, trafiğin sınıflandırılması için yalnızca normal ve DoS saldırılarını içeren dengeli veri seti kullanılmıştır.

2.3 Makine Öğrenimi

Makine öğrenimi, belirlenen bir görevi yerine getirmek için tanımlanan verileri kullanarak, sonuç tahmin etme yöntemidir. Sonuç tahmin etme işlemi için çeşitli algoritmalardan faydalanılır. Algoritmalar üzerinde değişiklikler yapılmaz, öğrenme işlemi tamamen veriler üzerinden elde edilen bir kazanımdır. Çalışmamızda 3 farklı makine öğrenmesi algoritması ile veri seti üzerinde normal ve DoS trafiği sınıflandırılmıştır.

2.3.1 Naive Bayes

Naive Bayes algoritmasının işleyişi Bayes teoremine dayanmaktadır ve ismini de bir matematikçi olan Thomas Bayes'ten almaktadır. Naive Bayes algoritması sınıflandırma yaparken her özelliği birbirinden bağımsız olarak değerlendirir. Örneğin bir meyveyi sınıflandırırken, meyvenin rengi ve boyutları bir arada değerlendirilmez. Algoritmada her bir eleman için tüm durumların olasılıkları hesaplanır. Bu hesaplamalardan olasılığı en yüksek olan duruma göre sınıflandırma yapılır.

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

Şekil 2. Bayes Teoremi

Şekil 2.'de yer alan formülde;

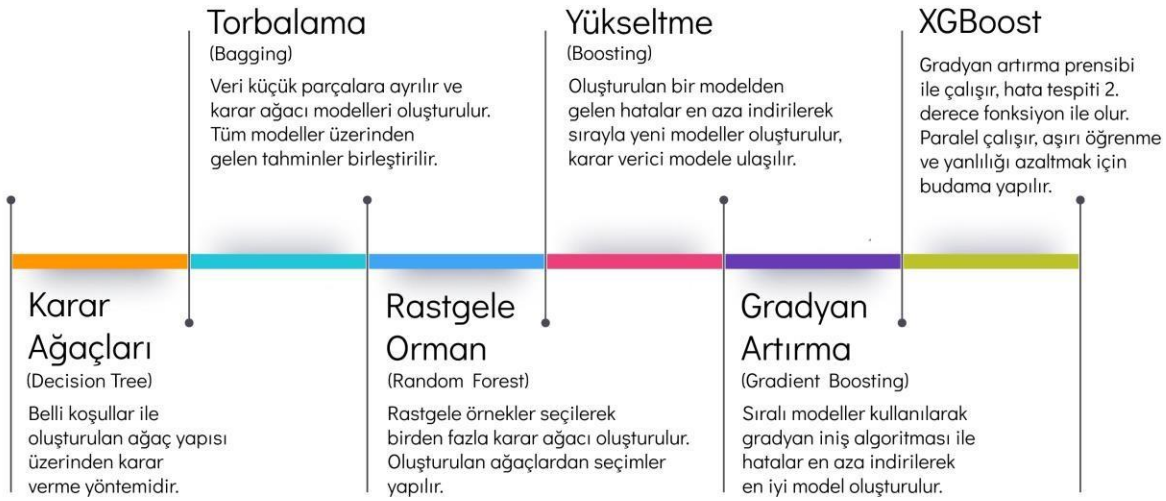
- P (A | B) = B olayı gerçekleştiğinde A olayının gerçekleşme olasılığını
- P (A) = A olayının gerçekleşme olasılığını
- P (B | A) = A olayı gerçekleştiğinde B olayının gerçekleşme olasılığını
- P (B) = B olayının gerçekleşme olasılığını temsil etmektedir.

Naive Bayes, olasılıksal hesaplamaya dayandığı için az veri ile elde edilen modellerden bile güzel sonuçlar elde edilebilmektedir. Basit bir formüle dayanıyor olmasından dolayı algoritma çok hızlı çalışmaktadır. Gaussian Naive Bayes, Multinominal Naive Bayes, Bernoulli Naive Bayes olmak üzere 3 çeşittir. Bu çalışmada sınıflandırmada yapılırken Gaussian Naive Bayes algoritması kullanılmıştır.

2.3.2 Gradient Boosting

Gradient Boosting algoritması, karar ağacı temelli bir algoritmadır. Karar ağaçlarının çalışma mantığının anlaşılması, Gradient Boosting hakkında fikir elde etme açısından önemlidir. Karar ağaçları, sınıflandırma ve regresyon tahminleri için yaygın kullanıma sahip yöntemlerden bir tanesidir. Kullanılan birçok yönteme göre, kolayca anlaşılabilir ve yorumlanabilir olması bu yöntemin sağladığı avantajlardandır. Karar ağacı, kök noktasından başlar, verileri yorumlayarak aşağıya doğru dallanmalar yapar ve yapraklar oluşturur.

Gradinet Boosting, gradyan artırma yöntemi ile zayıf öğrenicileri güçlü öğreniciye dönüştürme algoritmasıdır (Feng ve ark., 2018). Gradient Boosting algoritmasında oluşturulan her yeni karar ağacı, bir önceki ağaçta hesaplanan hataların en aza indirilmesi prensibine dayanır. Algoritmada ilk olarak oluşturulan karar ağacı ile bir tahmin elde edilir. Elde edilen tahmin ile hedef arasındaki fark hesaplanır. Her yeni iterasyonda, hesaplanan fark ile birlikte yeni bir ağaç oluşturulur. Sonuç olarak tahmin ile hedef arasındaki farkı sıfırlamak amaçlanır. Gradient Boosting algoritması hata oranını belli iterasyonlarla en aza indirmesiyle, yapay sinir ağlarına benzemektedir. Gradient Boosting'in yapay sinir ağlarına sağladığı en büyük avantaj ise açıklanabilirliğinin kolay olmasıdır.



Şekil 3. Karar ağacı algoritmalarının zaman içindeki gelişimi

2.3.3 XGBoost

XGBoost, ilk olarak 2016 yılında Tianqi Chen ve Carlos Guestrin tarafından makalesi yayınlan yenilikçi bir makine öğrenmesi algoritmasıdır. Yayınlanan makale, veri bilimciler tarafından ilgiyle karşılanmış, Kaggle yarışmalarında kullanımı gün geçtikçe artmıştır. XGBoost makalesinde yer alan

bilgiye göre, 2015 yılında 29 Kaggle yarışmasının 17 tanesi XGBoost ile kazanılmıştır (Chen ve Guestrin, 2016). XGBoost'un geliştirilmesi, veri bilimciler tarafından halen devam etmektedir ve Github'ta 500'den fazla katılımcısı, 5400'den fazla kod eklemesi bulunmaktadır.

XGBoost algoritması Gradient Boosting algoritmasının optimize edilmiş bir türüdür. Önceki versiyonlara göre sağladığı avantajları XGBoost kullanımının yaygınlaşmasındaki en önemli nedendir. XGBoost, ağacı oluştururken maksimum derinlik değerini kullanır. Oluşturulan ağaç aşağı yönde aşırı ilerleme gösterirse, budama gerçekleştirilir. Aşırı öğrenmenin önüne geçilir. Gradient Boosting algoritması, kayıp fonksiyonun hesaplanmasında birinci dereceden fonksiyon kullanırken, XGBoost bu hesaplamaları ikinci dereceden fonksiyonlar kullanarak gerçekleştirir. Paralel çalışma özelliği, diğer algoritlamalara göre sonuca daha kısa sürede ulaşılmasını sağlar.

2.3.4 Makine öğrenmesi değerlendirme metrikleri

Veriler kullanılarak, birçok farklı makine öğrenmesi algoritması ile farklı modeller oluşturmak mümkündür. Oluşturulan modellerden hangisinin daha iyi sonuç vereceğini ölçmek için değerlendirme metriklerine ihtiyaç duyulmaktadır. Değerlendirme metrikleri modelden elde edilen tahminler ile gerçek sonuçları karşılaştırarak bize rakamsal sonuçlar vermektedirler. Bu çalışmada makine öğrenmesi modellerinin doğruluğu, 4 farklı değerlendirme metriği ile ölçülmüştür. Değerlendirme metriklerinin açıklanması, karışıklık matrisleri üzerinden anlatılmıştır.

Karışıklık matrisi, sınıflandırma yapan uygulamalarda, gerçek ve tahmin edilen değerleri bir tablo üzerinden kolayca kıyaslayabilmek için kullanılmaktadır.

Çizelge 1. Karışıklık matrisi örneği

		Tahmin	
		Pozitif	Negatif
Gerçek	Pozitif	Gerçek Pozitif	Yanlış Negatif
	Negatif	Yanlış Pozitif	Gerçek Negatif

Çizelge 1.'de yer alan matriste;

- Gerçek Pozitif (GP): Doğru olarak tahmin edilen, gerçekte de doğru olan değerler,
- Gerçek Negatif (GN): Yanlış olarak tahmin edilen, gerçekte de yanlış olan değerler,
- Yanlış Pozitif (YP): Doğru olarak tahmin edilen, gerçekte yanlış olan değerler,
- Yanlış Negatif (YN): Yanlış olarak tahmin edilen, gerçekte doğru olan değerleri ifade etmektedir.

2.3.4.1 Doğruluk (Accuracy)

Doğru tahmin edilen değerlerin, tüm değerlere bölünmesi ile elde edilmektedir. Doğruluk değeri, 0 ile 1 arasındadır. Doğruluk değeri 1'e yaklaştıkça başarı artmaktadır.

$$\text{Doğruluk: } (GP + GN) / (GP + GN + FP + FN)$$

2.3.4.2 Duyarlılık (Recall)

Doğru olarak tahmin etmemiz gereken değerlerin, ne kadarını doğru tahmin ettiğimizi belirtmektedir. Duyarlılık değeri, gerçekte doğru olan ve doğru olarak tahmin edilen değerlerin, tüm doğru değerlere bölünmesi ile elde edilmektedir.

$$\text{Duyarlılık: } GP / (GP + FN)$$

2.3.4.3 Kesinlik (Precision)

Doğru olarak tahmin ettiğimiz değerlerin, ne kadarının gerçekte doğru olduğunu göstermektedir. Kesinlik değeri, gerçekte doğru olan ve doğru olarak tahmin edilen değerlerin, doğru olarak tahmin edilen tüm değerlere bölünmesi ile elde edilmektedir.

$$\text{Kesinlik: } GP / (GP + FP)$$

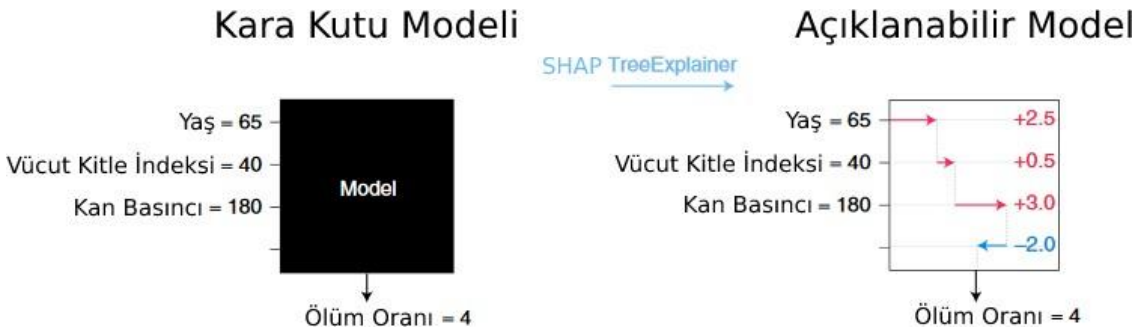
2.3.4.4 F1 Skoru (F1 Score)

F1 skoru, duyarlılık ve kesinlik değerlerinin harmonik ortalamasının hesaplanması ile elde edilmektedir. Her iki değerinde hesaplamaya katılarak dengeli bir değer elde edilmesi amaçlanmaktadır. Eşit dağılıma sahip olmayan veri setlerinde başarılı sonuçlar elde etmek için kullanılmaktadır.

$$\text{F1 Skoru: } 2 * \text{Kesinlik} * \text{Duyarlılık} / (\text{Kesinlik} + \text{Duyarlılık})$$

2.4 SHAP

Makine öğrenimi insan hayatını kolaylaştırma adına çok büyük gelişmeler gösterdi. Bu gelişime rağmen şirketler ve araştırmacılar algoritmaların bir kara kutu olması zorluğu ile karşılaşmaktadırlar. Kara kutu algoritmaları yoruma kapalıdır. Algoritmaya verilen bir girdiden, bir çıktı alınmakta fakat nedeni anlaşılamamaktadır. SHAP (SHapley Additive ExPlanations) ile makine öğrenmesi modelini kara kutu olmaktan çıkararak, yorumlanabilir bir hale getirmek mümkündür.



Şekil 4. Kara kutu modeli, açıklanabilir model

SHAP, makine öğrenmesi modelinin yorumlanmasında oyun teorisi yaklaşımını kullanır ve makalesi ilk olarak 2017 yılında, Lundberg and Lee tarafından yayınlanmıştır. SHAP, özelliklerin model oluşumuna katkısını ölçerken, Shapley değerlerini kullanır (Lundberg ve Lee, 2017). Shapley değeri, ilk olarak 1953 yılında Lloyd Shapley tarafından makalesi yayınlan ve her oyuncunun oyuna katkısını ölçmeye yarayan değerdir (Hausken ve Mohr, 2001). Veri setinde yer alan her bir değer için ayrı Shapley değeri hesaplanır. Her Shapley değeri, ilişkilendirildiği verinin tahminde yarattığı etkiyi belirtmektedir. Belli bir özelliğe ait tüm Shapley değerlerinin toplamı, o özelliğin tahmin için toplam katkısını belirtmektedir. SHAP kütüphanesi kullanılarak oluşturulan grafiklerde yer alan SHAP değerleri, Shapley değerlerinin toplamını temsil etmektedir.

3. BULGULAR VE TARTIŞMA

3.1 Trafiğin sınıflandırılması

Normal ve DoS trafiği içeren dengeli veri setinde 3 farklı makine öğrenmesi algoritması ile sınıflandırma yapılmıştır. Algoritmalar üzerinde herhangi bir hiper parametre optimizasyonu yapılmamıştır. Sınıflandırmaya ait sonuçlar Çizelge 2'de aktarılmıştır.

Çizelge 2. Farklı algoritmalar ile trafiğin sınıflandırılmasında elde edilen sonuçlar

Algoritma	Doğruluk (Accuracy)	Duyarlılık (Recall)	Kesinlik (Precision)	F1 Skoru
Naive Bayes	0.951	0.998	0.946	0.948
Gradient Boosting	0.969	0.999	0.964	0.968
XGBoost	0.980	0.995	0.981	0.979

Karışıklık matrisi değerlerine bakarak, 3 algoritma için de, trafik tiplerine göre başarı oranı gözlemlenebilmektedir.

Çizelge 3. XGBoost ile yapılan sınıflandırmanın karışıklık matrisi üzerindeki sonuçları

		Tahmin	
		DoS	Normal
Gerçek	DoS	543,375	56,625
	Normal	14,560	2,985,440

Çizelge 4. Gradient Boosting ile yapılan sınıflandırmanın karışıklık matrisi üzerindeki sonuçları

		Tahmin	
		DoS	Normal
Gerçek	DoS	491,141	108,859
	Normal	1,112	2,998,888

Çizelge 5. Naive Bayes ile yapılan sınıflandırmanın karışıklık matrisi üzerindeki sonuçları

		Tahmin	
		DoS	Normal
Gerçek	DoS	430,275	169,725
	Normal	3,722	2,996,278

Eğitim verisinde yer alan 8.4 milyon trafiğin, 1.4 milyonu DoS saldırısına aittir. Eşit dağılım söz konusu olmadığından dolayı, algoritmaların başarılarını değerlendirirken F1 skorunu dikkate almak önemlidir. XGBoost, 4 farklı değerlendirme metriği dikkate alındığında 3 değerlendirme metriğinde en başarılı sonuçları vermiştir. Gradient Boosting algoritması, duyarlılık ölçümünde en başarılı algoritma olmuştur. Naive Bayes ile yapılan tahminlerde başarı oranı diğer algoritmalara göre düşüktür. Normal trafiğin doğru sınıflandırılmasında en iyi sonuç Gradient Boosting algoritmasına aittir. DoS trafiğinin doğru olarak sınıflandırılmasında ise en iyi sonucu XGBoost algoritması vermektedir.

Model analizi bölümünde, özniteliklerin ve değerlerin tahmine etkisi araştırılacağı için, en başarılı algoritma ile oluşturulan modelin kullanılması doğru analiz için önem arz etmektedir. DoS verisinin toplam veri setindeki oranı az olmasına rağmen yüksek oranda doğru tahmin edilmiş olması ve metriklerdeki yüksek başarı oranından dolayı, model analizi, XGBoost ile elde edilen model üzerinde yapılmıştır.

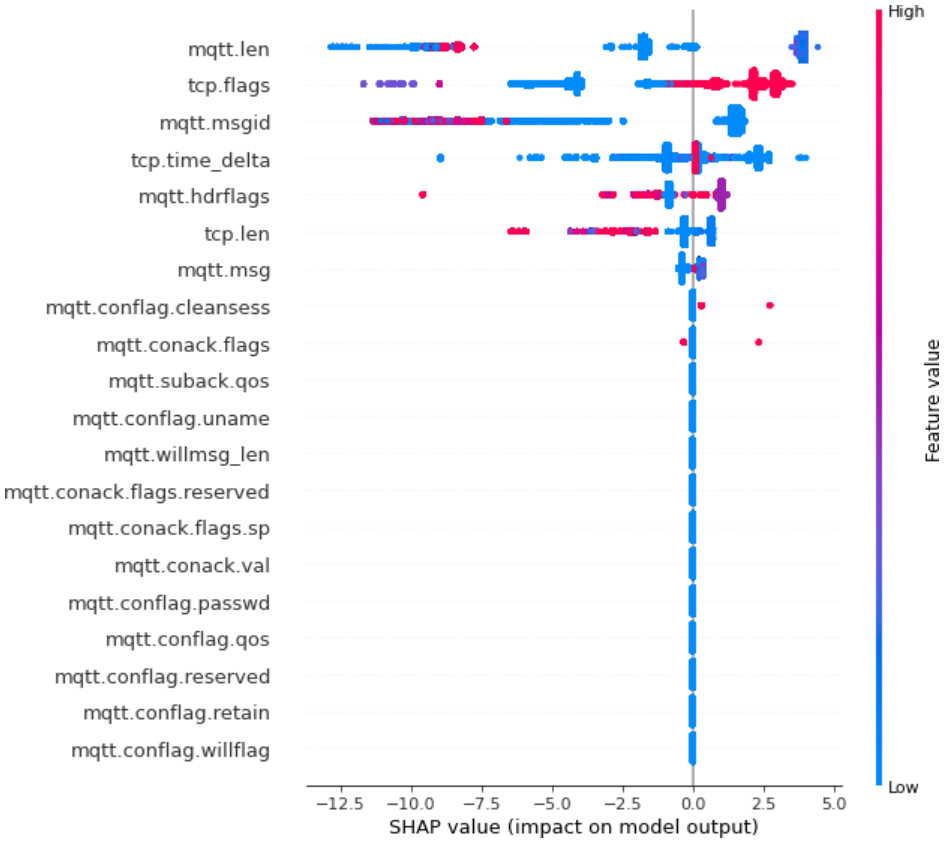
Model analizi adımından önce, başarı oranını artırmak amacıyla, XGBoost üzerinde hiper parametre optimizasyonu gerçekleştirilmiştir. XGBoost algoritmasında, 'max_depth' parametresi

için ‘3’, ‘6’, ‘9’ değerleri, ‘learning_rate’ parametresi için de ‘0.1’, ‘0.3’, ‘0.6’ değerleri denenmiştir. Bu parametrelerden en iyisi, ‘max_depth’ için ‘6’, ‘learning_rate’ içinde ‘0.1’ olarak hesaplanmıştır. Yeni parametreler ile oluşturulan yeni XGBoost modelinde 0.98 olarak hesaplanan doğruluk oranı değişmemiştir.

3.2 SHAP ile Model Analizi

Analiz için, en yüksek doğruluk oranına sahip XGBoost modeli seçilmiştir. Analiz adımında XGBoost sınıflandırılmasında elde edilen 0.98’lik doğruluk oranına sahip model kullanılmıştır.

SHAP grafikleri, model sonucuna etki eden tüm SHAP değerleri üzerinden oluşturulmaktadır. Veri setinde yer alan her hücreye ait bir SHAP değeri hesaplanmaktadır. Yapılan sınıflandırmaya olumsuz etkisi olan SHAP değerleri negatif işaretli, olumlu etkisi olan SHAP değerleri pozitif işaretlidir. SHAP özet grafiklerinde mavi renkler, veri değerinin küçük olduğu durumları, kırmızı renkler ise veri değerinin büyük olduğu durumları göstermektedir.

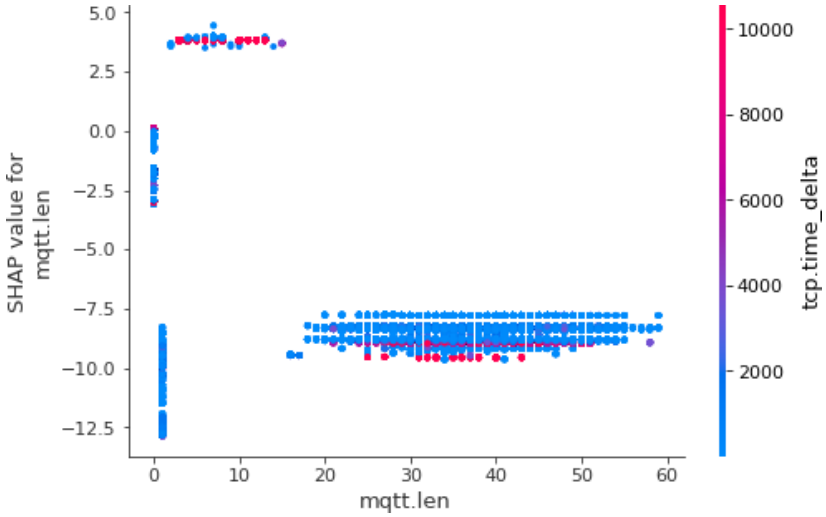


Şekil 5. Öznitelikler üzerinden, model için hesaplanan SHAP değerleri

SHAP grafiklerini doğru değerlendirmek modeli daha iyi analiz etmek açısından önemlidir. Şekil 5‘teki grafikte, y ekseninde veri setinde yer alan öznitelikler belirtilmiştir. Bu öznitelikler önem sırasına göre yukarıdan aşağıya doğru sıralanmıştır. Grafiğe baktığımızda sınıflandırma yaptığımız model için en önemli öznitelik ‘mqtt.len’ olarak hesaplanmıştır. X ekseninde yer alan SHAP değerleri ise sınıflandırmaya pozitif ve negatif etkiyi temsil etmektedir. Grafik incelendiğinde ‘mqtt.conack.flags’ özelliğinin altında yer alan özelliklerin sınıflandırmaya hiçbir katkısı olmadığı gözlemlenmektedir. Renklerden yola çıkarak, grafik üzerinde yorumlar yapılabilmektedir. ‘tcp.flags’ özelliğine bağlı renkleri incelendiğinde, büyük ‘tcp.flags’ değerlerinin (kırmızı noktalar), sınıflandırmaya pozitif yönde etki ettiği, küçük değerlerin (mavi) ise sınıflandırmaya negatif yönde etki ettiği görülmektedir. Sınıflandırma için en önemli özellik olarak hesaplanan ‘mqtt.len’ değerleri

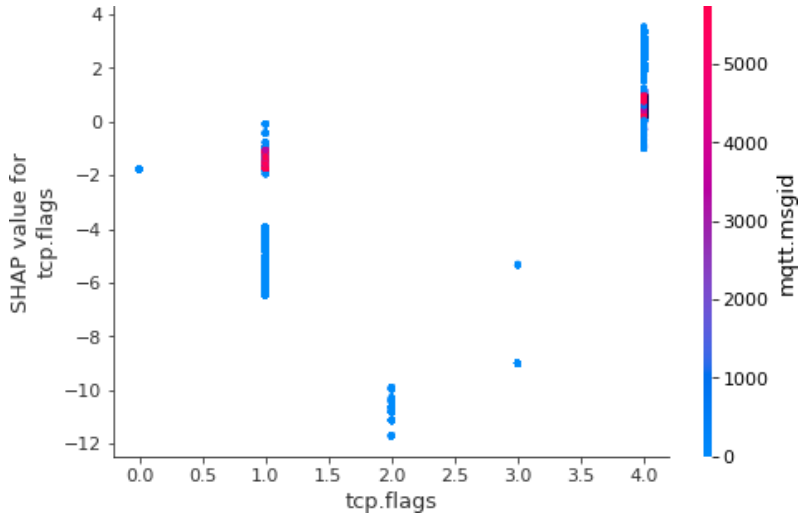
incelendiğinde sağ kısımda koyu mavi noktaların (düşük değerlerin), pozitif katkısından bahsetmek mümkündür.

SHAP özet grafikleri bize birçok noktada modeli yorumlama imkânı sunmaktadır, fakat hangi özellik değerlerinin sınıflandırmaya hangi etkide bulunduğu bilgisine SHAP bağlılık grafiklerinden ulaşılmaktadır. SHAP bağlılık grafiklerinde her bir nokta veri setindeki gerçek veriyi temsil etmektedir. Bağlılık grafiğinin amacı, bir özniteliğe ait değerlerin model çıktısına yaptığı olumlu ve olumsuz katkıları gözlemlemektir. SHAP özet grafiğinde renkler veri değerindeki büyüklük ve küçüklüğü temsil ederken, bağlılık grafiğinde renkler sağ tarafta y ekseninde bulunan öznitelik değerleri ile girilen etkileşimi ifade etmektedir. Sağ taraftaki y ekseninde bulunan öznitelik, x ekseninde belirttiğimiz öznitelik ile en çok etkileşime giren özniteliği ifade etmektedir. Sağ kısımdaki öznitelik SHAP tarafından otomatik olarak belirlenir. Özet grafiğinde sınıflandırmaya etkisi olan en önemli 4 öznitelik için grafikler oluşturulmuştur.



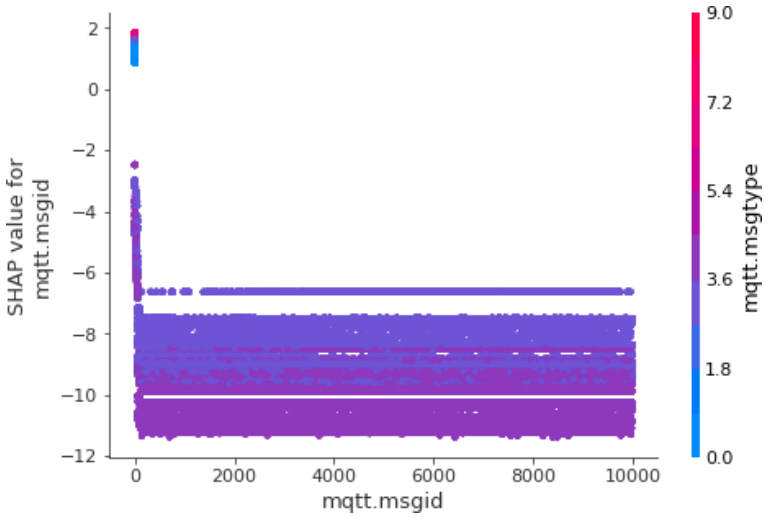
Şekil 6. 'mqtt.len' değerlerinin sınıflandırmaya etkisi

'mqtt.len' değerlerinin sınıflandırmaya etkisine baktığımızda, 1 ile 18 arasındaki değerlerin tamamının pozitif etkisinden bahsedilebilmektedir. 0 ve 1 değerleri ile 18-60 arası değerlerin tamamı negatif yönde etki etmişlerdir. Kırmızı noktalar sağ taraftaki değerler ile girilen yüksek etkileşimi ifade etmektedir. Mavi noktalar etkileşimin olmadığı verileri ifade etmektedir.



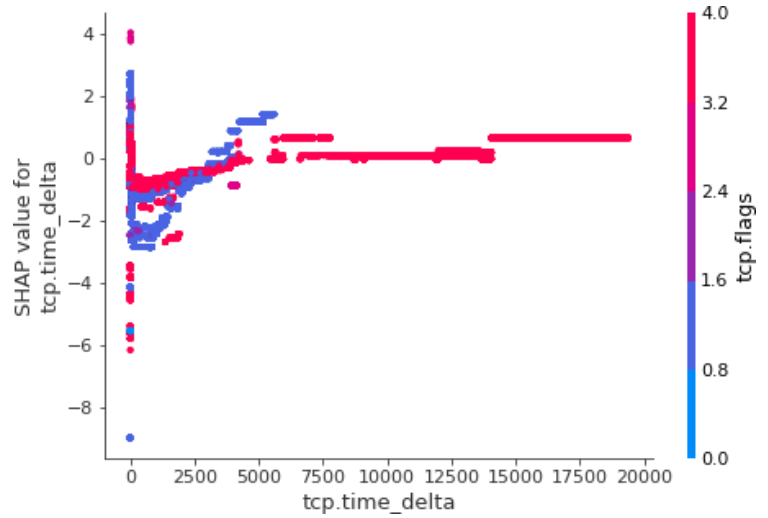
Şekil 7. 'tcp.flags' değerlerinin sınıflandırmaya etkisi

Özet grafiğinde ‘tcp.flags’ özelliğine ait değerler büyüdükçe modele pozitif katkı sağlandığı gözlemlenmişti. Şekil 6’daki grafikte bu değerin 4 olduğu bilgisi elde edilmektedir.



Şekil 8. ‘mqtt.msgid’ değerlerinin sınıflandırmaya etkisi

Sıfırdan farklı tüm ‘mqtt.msgid’ değerleri modele negatif yönde etki etmiştir. 0 değerlerinin ise pozitif katkısı görülmektedir. Grafikler bize sadece veri tipleri için SHAP değerlerinin dağılımını göstermektedir. Örneğin ‘mqtt.msgid’ değerinin 0 olduğu durumda pozitif etki vardır fakat bu durum kaç veri için geçerlidir bilinememektedir. Veri setinde 0 olan verilerin frekansına ait bilgi çıkarımı yapılamamaktadır.



Şekil 9. ‘tcp.time_delta’ değerlerinin sınıflandırmaya etkisi

Grafikte x eksenindeki değerler, XGBoost tarafından kategorik dönüşümler yapılan zaman değerleridir. Zaman bilgisi sayılarla ifade edilmiştir. Grafik incelendiğinde ‘tcp.time_delta’ verilerinin ‘tcp.flags’ ile yüksek etkileşimde olduğu görülmektedir. 0-5000 arası değerlerin negatif etkilerinden bahsedilebilmektedir. 0 için hem pozitif hem negatif etki görülmektedir. 5000’den büyük değerlerin ‘tcp.flags’ ile etkileşimi yüksek olmasına rağmen etkileri sıfıra yakındır.

4. SONUÇ

MQTTset veri seti kullanılarak, 3 farklı makine öğrenmesi algoritması ile normal ve DoS trafiği sınıflandırılmıştır. Oluşturulan modeller üzerinde, 4 farklı değerlendirme metriği ve karışıklık matrisi ile başarı oranları gözlemlenmiştir. DoS tespitinin sınıflandırılması, doğruluk, kesinlik ve F1 skoru metriklerinde en başarılı algoritma XGBoost algoritması olmuştur. Doğruluk, kesinlik ve F1 skorunda elde edilen başarı oranları 0.98 seviyesinde gözlemlenmiştir. SHAP analizi için gerekli model seçiminde bu yüzden XGBoost algoritması ile oluşturulan model kullanılmıştır.

SHAP ile model analizi yapıldığında elde edilen sonuçlara göre, makine öğrenmesi sınıflandırma yaparken 33 özelliğin sadece 7 tanesini kullanmıştır. Bu 7 özelliğin 5 tanesi önemli ölçüde sonuca etki etmiştir. Bu veriler, veri setinin küçültülmesi ve eğitim süresinin kısalması için faydalı olacaktır.

XGBoost modelinde elde edilen karışıklık matrisi sonuçlarına göre, MQTT trafiğinde, makine öğrenmesi ile normal trafik %99, DoS trafiği ise %90 oranında doğru sınıflandırılmıştır. Veri setinde Normal trafiğin sayısı, Dos trafiğinin 5 katıdır. Bu durum makine öğrenmesi modelinin normal trafiği tespit etmek yönünde eğilmesine sebep olmuştur. DoS saldırısının sınıflandırılmasında en önemli özellikler ‘mqtt.len’, ‘tcp.flags’ ve ‘mqtt.msgid’ olmuştur. Sınıflandırmada zaman bilgisine 4.sırada önem verilmiştir. Veri setinde, belirtilen önemli 3 özniteliğe ait değerlerin normal ve DoS için belirgin ayrımlar gösterdiği anlaşılmaktadır.

Analizler sonucunda, kompleks ağ verileri ile oluşturulan makine öğrenmesi modeli yorumlanabilir hale getirilmiştir. Uygulanılan yöntemler, modelin yorumlanması alışkanlığının kazanılması açısından önemlidir. Makine öğrenmesi modelini analiz etme ve açıklama üzerinde uygulanılan bu yenilikçi yöntemler, başka veri setleri ve uygulamalar üzerinde de denenebilir ve bu konuda veri bilimcilere yol göstererek doğru sonuçlar elde edilmesine yardımcı olabilir.

5. ÇIKAR ÇATIŞMASI

Yazarlar, bilinen herhangi bir çıkar çatışması veya herhangi bir kurum/kuruluş ya da kişi ile ortak çıkar bulunmadığını onaylamaktadırlar.

6. YAZAR KATKISI

Çalışmanın kavramsal ve tasarım süreçlerinin belirlenmesi, çalışmanın kavramsal ve tasarım süreçlerinin yönetimi, veri analizi ve yorumlama, makale taslağının oluşturulması ve fikirsel içeriğin eleştirel incelemesine her iki yazarda katkı sağlamıştır. Ayrıca Ali Cihat KELLE, veri toplama kısmına da katkı sağlamıştır.

7. KAYNAKLAR

Chen T., Guestrin C., Xgboost: A scalable tree boosting system, In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, San Francisco California / USA, August 13-17, 2016, pp: 785-794.

Feng Z., Xu C., Tao D., Historical Gradient Boosting Machine. GCAI-2018: 4th Global Conference on Artificial Intelligence, Luxembourg City / Luxembourg, September 18-21, 2018.

Gündoğan C., Kietzmann P., Lenders M., Petersen H., Schmidt T., Wählisch M., NDN, CoAP, and MQTT: a comparative measurement study in the IoT, Proceedings of the 5th ACM Conference on Information-Centric Networking, Boston Massachusetts, September 21-23, 2018, pp: 159-171.

Hausken K., Mohr M., The Value of a Player in n-Person Games. Social Choice and Welfare 18(3), 465-483, 2001.

Lundberg S., Lee S., A Unified Approach to Interpreting Model Predictions, In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), Long Beach California / USA, December 4-9, 2017, pp: 4768-4777.

Naik N., Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP, 2017 IEEE International Systems Engineering Symposium (ISSE), Vienna / Austria, October 11-13, 2017, pp: 1-7.

Nebbione G., Calzarossa M. C., Security of IoT Application Layer Protocols: Challenges and Findings. Future Internet 12(3), 55, 2020.

Özdoğan E., Erdem A., Nesnelerin İnterneti İçin Hibrit Uygulama Katmanı Protokol Tasarımı. Mühendislik Bilimleri ve Tasarım Dergisi 8(1), 285-304, 2020.

Saritha S., Sarasvathi V., A study on application layer protocols used in IoT, 2017 International Conference on Circuits, Controls, and Communications (CCUBE), Bangalore / India, December 15-16, 2017, pp: 155-159.

Tantitharanukul N., Osathanunkul K., Hantrakul K., Pramokchon P., Khoenkaw P., MQTT-Topics Management System for sharing of Open Data. 2017 International Conference on Digital Arts, Media and Technology (ICDAMT), Chiang Mai / Thailand, March 1-4, 2017, pp: 62-65.

Vaccari I., Chiola G., Aiello M., Mongelli M., Cambiaso E., MQTTset, A New Dataset for Machine Learning Techniques on MQTT. Sensors, 20(22), 6578, 2020.