MQTT TABANLI IOT SİSTEMİNE YAPILAN SALDIRILARIN MAKİNE ÖĞRENİMİ KULLANILARAK TESPİTİ

Ayça Nur KAHYA[1*], Esra Nergis YOLAÇAN[2]

[1] Eskişehir Osmangazi Üniversitesi, Bilgisayar Mühendisliği Bölümü, Eskişehir,
ORCID No : https://orcid.org/0000-0002-6950-4421
[2] Eskişehir Osmangazi Üniversitesi, Bilgisayar Mühendisliği Bölümü, Eskişehir,
ORCID No : https://orcid.org/0000-0002-0008-1037

Anahtar Kelimeler
ESP8266 Wifi Modülü
MQTT Protokol
MQTTset Veri Kümesi
Saldırı Tespiti
Makine Öğrenimi

Öz

Günümüzde cihazların neredeyse tümü internete bağlanma potansiyeli taşımaktadır. Bu nedenle IoT cihazların kullanımı gün geçtikçe artmaktadır. İnternete bağlı cihazların büyük bir kısmı siber saldırılara karşı savunmasız olmaktadır. Çok sayıda cihazın bağlı olduğu ağlarda saldırılar, ağ ve cihazların güvenliği için kritik bir konudur. Yüksek başarı oranları sayesinde makine öğrenmesi yaklaşımları, IoT güvenliğini etkileyen saldırıların tespit edilmesinde ön plana çıktığı görülmektedir. Bu çalışmada, IoT platformunda en sık kullanılan protokollerden biri olan MQTT protokolüne gerçekleşen saldırıların tespit edilmesi hedeflenmiştir. Bu amaçla, öncelikle literatürdeki çalışmalar incelenerek, saldırı tipleri belirlenmiş ve gerekli test ortamı oluşturulmuştur. Ardından MQTT protokolünü kullanan ağa, siber saldırılar uygulanarak makine öğrenimi algoritmaları ile saldırı tespitinde performansları test edilerek değerlendirilmiştir. Sunulan çalışmada, gerçekleştirilen veri kümesi iyileştirmesinin MQTT saldırılarının tespitinde başarı oranının artmasına katkı sağladığı görülmüştür.

ATTACKS ON THE MQTT-BASED IOT SYSTEM DETECTION USING MACHINE LEARNING

Keywords
ESP8266 Wifi Module
MQTT Protocol
MQTTset Dataset
Intrusion Detection
Machine Learning

Abstract

Almost all devices today have the potential to connect to the internet. Therefore, the use of IoT devices is increasing day by day. Most of the devices connected to the internet are vulnerable to cyber-attacks. In networks where a large number of devices are connected, attacks are critical to the security of the network and devices. Due to high success rates, machine learning approaches appear to be at the forefront of detecting attacks affecting IoT security. In this study, it was aimed to detect attacks on the MQTT protocol, which is one of the most commonly used protocols on the IoT platform. For this purpose, first of all, studies in the literature have been examined, attack types have been determined and the necessary test environment has been created. Then, cyber-attacks were applied to the network using the MQTT protocol and their performance in Intrusion Detection was tested and evaluated with machine learning algorithms. In the presented study, it was shown that realizable dataset improvement contributed to increased success rate in detection of MQTT attacks.

Araştırma Makalesi		Research Article	
Başvuru Tarihi	: 27.10.2021	Submission Date	: 27.10.2021
Kabul Tarihi	: 03.02.2022	Accepted Date	: 03.02.2022

* Sorumlu yazar; e-posta : aycanurkahya@gmail.com

1. Giriş

İçinde bulunduğumuz teknoloji çağında, internet kullanımı ve internete bağlı cihazların sayısı sürekli artmaktadır. Bu küresel ağ kullanılarak veriler yoğun bir şekilde paylaşılabilir, depolanabilir ve birbirleriyle etkileşim sağlamaktadır. İnternete olan ihtiyaç arttıkça çeşitli tehditler, güvenlik açıkları ve saldırılar ortaya çıkmakta ve ağ güvenliğinin sağlanması bir zorunluluk haline gelmektedir (Zhang, Li ve Wang, 2019).

Son yıllarda, ağ altyapılarında ve akıllı cihazlarda sağlanan gelişmeler ile bu cihazların genişleyen kullanılabilirliği, Nesnelerin İnterneti (IoT) kavramını oluşturmuştur. Birçok farklı cihazın birbiri ile iletişim sağladığı bu ağları, günümüzde endüstriyel alanda olduğu kadar ev sistemleri ile günlük hayatımızın içinde de yer almaktadır. Ancak mevcut IoT cihazlarının temel güvenlik mekanizmalarından yoksun olması ve IoT protokollerindeki eksiklikler güvenlik açıkları oluşturmaktadır (Vaccari, Chiola, Aiello, Mongelli ve Cambiaso, 2020). Bu sistemlerdeki zayıflıklardan faydalanan siber suçlular, sunuculardan bulut depolamaya kadar birçok noktada yer alan bilgilerden yararlanmak için çeşitli saldırılar gerçekleştirmektedirler. Güvenliği oldukça kritik olan IoT sistemlerin siber saldırılardan etkilenmemesi ve verilerin korunması amacıyla bu saldırıların tespit edilmesi için makine öğrenmesi ve yapay zekâ algoritmaları kullanılmaktadır. Siber güvenlik kapsamında saldırı tespitinde birçok veri kümesi bulunmaktadır. Fakat bu veri kümeleri IoT ortamlarına içermiş oldukları nitelikler sebebiyle nadiren uygundur. Açık kaynaklı olan MQTTSet veri kümesi IoT uygulamalarında kullanıldığı bu veri kümesine katkı sağlayacak nitelikler eklenerek literatüre katkı olması amaçlanmıştır.

Bu çalışmada, IoT sistemlerde yaygın olarak kullanılan MQTT (Message Queuing Telemetry Transport) protokolü ile akıllı ev ağındaki cihazların iletişimi sağlanmış ve DoS, MITM, SlowITe ve Kaba Kuvvet saldırıları gerçekleştirilerek MQTTset veri kümesi genişletilmiştir. Bu kapsamda, açık kaynak teknolojileri kullanılarak MQTT protokolüne yapılan saldırıları için veri örnekleri sayı ve çeşit olarak artırılarak literatüre katkı sağlanmıştır. IoT uygulamalarında kullanılan MQTTset veri kümesine doğru tespit performansının artmasına katkı sağlayacak nitelikler eklenmiştir. Böylece, gelecekte araştırmacılar için bir bakış açısı kazandırılması amaçlanmıştır. Bu çalışmanın 2. Bölümünde literatür incelemesi sunulmuştur. 3. Bölümde önerilen yöntemin metodolojisi ve uygulamada kullanılan araçlara yer verilmiştir. 4. Bölümde bulgular ve uygulama hakkında değerlendirmeler yapılmıştır. 5. Bölümde ise elde edilen sonuçlar ve gelecekteki çalışmalar için öneriler yer almaktadır.

2. Bilimsel Yazın Taraması

IoT sistemlerinde siber tehditlerden korunmak için alınacak önlemler ve saldırı tespit sistemleri konuları sürekli olarak saldırı tehditlerinin ortaya çıkması sebebiyle oldukça geniş bir alanı kapsamaktadır. Bu bölümde IoT sistemlere yapılan saldırıların tespiti, veri kümeleri, makine öğrenmesi ve yapay zekâ konularını içeren çalışmalar incelenmiştir.

Öznitelik tabanlı şifreleme kullanarak mesaj aktarımlarının güvenliğinin sağlandığı çalışmada CloudMQTT adlı bulut tabanlı aracı kullanmışlardır (Gupta, Khera ve Turk, 2021). Sensörler ile kurulan ağda, gerekli bilgiler toplanır ve ona entegre edilmiş Wi-Fi modülü, verilere erişmek, ayrıştırmak ve uygun aboneye göndermek için CloudMQTT aracısına yüklemektedir. NodeMCU, tüm ağın yayın yaptığı tüm konulara abone olmuştur. KP-ABE şemaları erişim politikalarını belirler ve seçilen özelliklere göre şifreleri hesaplamaktadır. IoT projelerinde erişim politikaları sıklıkla tercih edilmektedir. Şifreleme yöntemleri ile saldırılar için önlem alınsa da saldırı tespit sistemlerinde makine öğrenimi uygulamalarının kapsamı her geçen gün artmaktadır. Saldırı tespit sistemlerinin test edilmesi için KDD CUP 99 veri kümesi geliştirilmiştir. Shalaginov, Semeniuta ve Alazab (2019) sundukları çalışmada yazarlar KDD CUP 99 veri kümesi kullanarak, Orange Pi Zero ve Arduino Uno'nun YSA (Yapay Sinir Ağları) eğitimi sırasında güç tüketimlerini karşılaştırmışlardır. KDD CUP 99 veri kümesinin kullanıldığı diğer çalışmada veri ön işleme ile ayrıştırılarak eğitim ve test veri kümelerine bölünmüştür. Agrawal, Agrawal ve Yadav (2020) tarafından sınıflandırma ve saldırı test işlemlerinden oluşan bir çalışma gerçekleştirmiştir. WEKA sınıflandırıcı Bayes Net, Naive Bayes, J48 ve Random Forest üzerinde testleri gerçekleştirdiler. IoT izleri ve MQTT niteliklerini içermeyen ve bu yönden KDD CUP 99 veri kümesi ile benzerlik gösteren diğer veri kümesi NSL-KDD'dir. Karande ve Joshi (2021) tarafından sunulan çalışmada trafiğe dayalı olarak önerilen yöntemin saldırı tespitinin doğruluğunu değerlendirdiler. Ayrıca sistem performansı ve veri kümesinin analizlerini içeren çalışmalar gerçekleştirdiler. Oral ve Çakır (2017) tarafından nesnelerin interneti hakkında bilgiler vermişlerdir. Elde ettikleri bilgiler ışığında örnek bir prototip uygulama gerçekleştirdiler. Gerçekleştirdikleri uygulamada NodeMCU modülü ve nem sensörü kullandılar. Sensörlerin MQTT protokolü ile haberleşmesini sağladılar. MQTT Dash uygulamasını mobil cihaza kurarak sıcaklık ve nem verilerini anlık olarak takip edilmesini sağladılar. Mısır ve Gökrem (2020) tarafından gerçekleştirdikleri uygulama ile MQTT protokolü, haberleşme hiyerarşisi, mesaj iletim metotları hakkında detaylı açıklamalarda bulundular. Yapmış oldukları uygulamada ESP8266 modülü, Raspberry Pi Zero , Wemos NodeMCU V3 Lolin kit, ve

CC3200 Launch Pad kitini kullandılar. MQTT arabulucusu olan Mosquito kullanarak yayımcılar ve aboneler arasında doğrudan veri alışverişini gerçekleştirdiler. İstemci olarak ise MQTT Paho kullanarak farklı seviyelerdeki konularla veri paylaşımını bir ağ kurarak gösterdiler.

IoT izleri barındıran veri kümeleri için saldırı tespit sistemleri, veri kümelerinin karşılaştırılması ve sistem performanslarının analiz edilmesi konularında yazarlar birçok çalışma yapmıştır. IoT saldırı tespit sistemleri, performans iyileştirme modelleri, veri kümelerinin detaylı açıklaması ve saldırı tespit sistemi optimizasyonu hakkında derinlemesine bilgiler verilmiştir (Khraisat ve Alazab, 2021). IoT ağlarında izinsiz giriş tespit sistemi için iki seviyeli bir anormal aktivite algılama modeli önerdikleri çalışmada Random Forest sınıflandırıcı ve IoT Botnet veri kümesini kullanmışlardır. Yapay Sinir Ağları, kaynak kullanımını iyileştirmek ve karmaşıklığını azaltmak için kullanılan yöntemdir. YSA yöntemleri ile azaltılmış veri kümesinde, saldırının tespitinde tahmin yüzdesi daha yüksek olduğu saptanmıştır (Manzoor ve Kumar, 2017).

YSA (Yapay Sinir Ağları) algoritmalarının eğitilmesi ile IoT sisteminde gönderilen verilerin geçerli ve geçersiz olup olmadığını tespit etmek için yaptıkları çalışmada başarılı sonuçlar elde etmiştir. Bununla birlikte, çeşitli veri kurcalama saldırıları içeren çeşitli ve zenginleştirilmiş veri kümeleri, pratik ayarlarda yüksek doğruluğu koruyup koruyamayacağını veya diğer gelişmiş öğrenme algoritmalarının gerekli olup olmadığını doğrulamak için YSA'yı eğitmek ve test etmek için kullanmışlardır (Canedo ve Skjellum, 2016).

MQTT niteliklerini barındıran veri kümeleri kullanılarak makine öğrenmesi algoritmalarından Sinir ağı (Neural Network), Random Forest, Naive Bayes, Karar Ağacı (Decision Tree), Gradient Boost ve Çok Katmanlı Algılayıcı (Multilayer Perceptron)'lar ile çalışmalar gerçekleştirilmiştir. Birden fazla makine öğrenimi tekniğini kullandıkları çalışmada anormallik algılama K-means kümeleme, SO-GALL (Single-Objective Generative Adversarial Active Learning), OCSVM (One-Class Support Vector Machines), Random Forest, Isolation Forest ve AutoEncoder modelleri ile gerçekleştirilmiş ve sonuçları karşılaştırılmıştır. Raspberry Pi'ye bütünleşmiş sensörler ve Node-RED5 IoT programlama aracı ile entegre bir ağ kurulmuştur. ARTEMIS saldırı tespit sisteminde OneClassSVM kullanarak 0.9998 doğrulukla, kötü amaçlı MQTT mesajlarını tespit ettiler (Ciklabakkal ve diğ., 2019). MQTT niteliklerini içeren diğer veri kümesi MQTT-IOT-IDS2020 oluşturularak eğitim ve değerlendirme süreçleri için altı farklı makine öğrenimi tekniği ile veri kümesi test edilmiştir (Hindy, ve diğ., 2020). Vaccari ve diğ., (2020) tarafından sunulan çalışmada ise MQTTset veri kümesi kullanılmıştır. Siber güvenlik alanında yaygın olarak kullanılan makine öğrenmesi algoritmalarından Sinir ağı (Neural Network), Random Forest, Naive Bayes, Karar Ağacı (Decision Tree), Gradient Boost ve Çok Katmanlı Algılayıcı (Multilayer Perceptron) kullanılarak veri kümesi doğrulanmış ve algoritmalar karşılaştırılmıştır. Aveleira-Mata, Alaiz-Moreton (2019) tarafından sunulan çalışmada veri kümesi, protokolü yöneten bir sunucu ile birlikte MQTT protokolünü kullanan çeşitli sensörlerin ve cihazların durumuna bakan diğer bilgisayarları kullanarak bir ortamdan bilgi toplamaktadır. Saldırı tespit sistemi ile WLAN ağının tüm çerçevelerini toplayıp ve bunları bir makine öğrenimi modeli aracılığıyla işlemektedir.

Tablo 1'de 1998 yılından 2020 yılına kadar yayınlanmış veri kümeleri yer almaktadır. Bu veri kümelerinde MQTT niteliklerinin yer alan tek veri kümesi MQTTset'dir. Bu çalışmada MQTT protokolüne gerçekleştirilen saldırılar incelendiği için MQTTset veri kümesi seçilmiştir. MQTT niteliklerinin yer aldığı MQTT-IOT-IDS2020 veri kümesinde MQTT niteliklerinin sayısı az olduğu için tercih edilmemiştir. Tablo 1'deki diğer veri kümelerinde ise MQTT niteliklerini içermemektedir.

Tablo 1

Veri Kümelerinin Karşılaştırması

Veri Kümesi	Etiket Verileri	IoT İzleri	MQTT Nitelikleri	Yıl
DARPA 98	✓	✗	✗	1998
DARPA 99	✓	✗	✗	1999
KDDCUP 99	✓	✗	✗	1999
NSL-KDD	✓	✗	✗	2009
ISCX 2012	✓	✗	✗	2012
ADFA-WD	✓	✗	✗	2014
ADFA-LD	✓	✗	✗	2014
CICIDS2017	✓	✗	✗	2017
CICIDS2018	✓	✗	✗	2018
Bot-IoT	✓	✓	✗	2018
TON_IoT	✓	✓	✗	2019
MQTTset	✓	✓	✓	2020
MedBIoT	✓	✓	✗	2020
IoT Botnet	✓	✓	✗	2020
MQTT-IOT-IDS2020	✓	✓	✓	2020

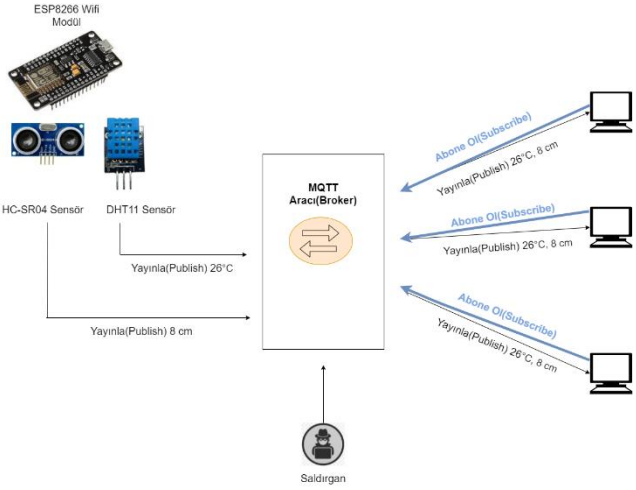
Literatürdeki çalışmalardan görüldüğü üzere, MQTT protokolü özelinde geliştirilmiş veri kümeleri oldukça kısıtlıdır. Veri kümelerinin detaylı incelemesi sonucunda, IoT güvenliği ve saldırı tespiti amacıyla oluşturulmuş birkaç veri kümesinde ise, MQTT ile ilgili niteliklerin sayı olarak oldukça az olduğu ve daha çok

network katmanında yapılan saldırıların tespitinin hedef alındığı görülmüştür. Bu çalışmada, belirlenen veri kümesi eksikliğinin giderilebilmesi amacı ile MQTT saldırılarını içeren yeni bir veri kümesi oluşturulmuş ve var olan MQTTset veri kümesi zenginleştirilerek literatüre katkı sağlanmıştır. Geliştirilen veri kümesi sayesinde algoritmaların başarı oranlarının iyileşmesi hedeflenmiş ve gerçekleştirilen testler ile performansları karşılaştırılmıştır.

3. Yöntem

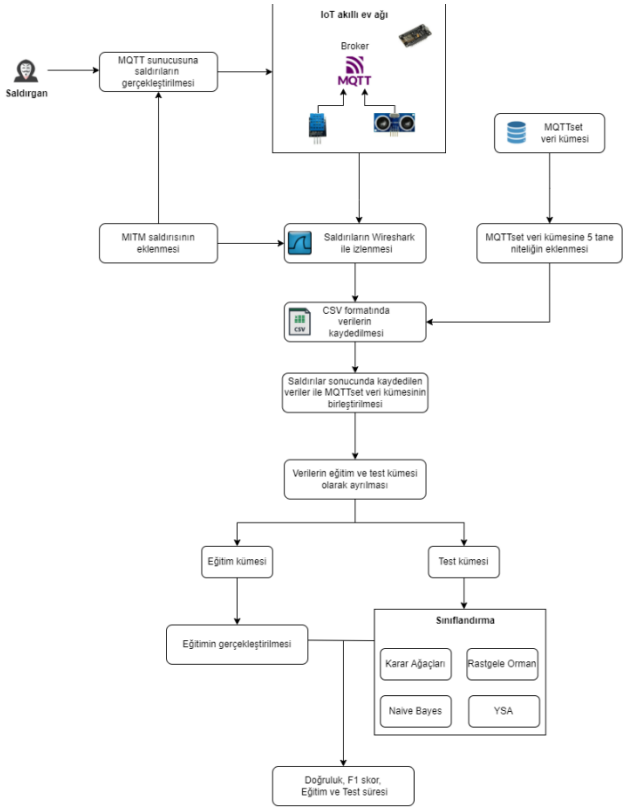
Kullanılan donanımlar, veri kümesi, saldırı araçları ve saldırı tespit yöntemleri için önerilen metot ve yaklaşımlar bu bölümde açıklanmıştır.

MQTT protokolü, IoT platformlarında kullanılan hafif bir mesajlaşma protokolüdür. Son zamanlarda en popüler mesajlaşma protokolleri arasında olması ve kullanım kolaylığı bu çalışmada tercih sebebi olmuştur. MQTT minimum kontrol mesajı 2 bayt, maksimum 256 MB boyutunda olabilir. MQTT protokolünün temel unsuru iletişimi sağlayan bir aracı (broker) olmakla birlikte, yayıncı veya abone olabilen istemci (client) ile sanal bir iletişim kanalı görevi yapan konu (topic) bileşenlerinden oluşur. Aracı, tüm mesajları almaktan, mesajları filtrelemekten, her mesaja kimin abone olduğunu belirlemekten ve ardından mesajı bu abone olan istemcilere yönlendirmekten sorumludur. İstemci, bir MQTT kitaplığı veya SDK çalıştırır ve ağ üzerinden bir MQTT aracısına bağlanır. Hem yayıncı hem de aboneye MQTT istemcileri denir. Konu ise bir MQTT istemcisinin yayınladığı ve aynı zamanda istemcilerin abone olduğu başlıklardır. Mesajların yayınlanması ve abone olunması için merkezi dağıtım merkezi görevi görür. Bir istemci, verileri belirli bir konu için yayınlar. Konuya ekli mesaj internet üzerinden MQTT aracısına gönderilir. Bu belirli konuya başka bir istemci tarafından abone olunması gerekir ve mesajı MQTT aracısından alır. Konular, UTF-8 ile kodlanmış, basit ve hiyerarşik dizelerdir]. MQTT protokolü, TCP/IP yığınına dayanır ve yapılan saldırılar uygulama katmanı saldırılarıdır (GSL, 2022). Bu çalışmada oluşturulan IoT ağına ait mimari Şekil 1'de verilmiştir. Bu çalışmada Wifi modülü (ESP8266), sıcaklık ve nem sensörü (DHT11) ve ultrasonik mesafe sensörü (HC-SR04) kullanılarak bir IoT ağı oluşturulmuştur. Bu bölümde IoT ağında yer alan sensörlerin detaylı bilgileri ve çalışmada nasıl kullanıldığına değinilmiştir.



Şekil 1. IoT Akıllı Ev Ağı

Şekil 2'de verilen diyagramda uygulamada baştan sona bir akış verilmiştir. Bu diyagramda oluşturulan akıllı ev ağında kullanılan MQTT sunucusuna yapılan saldırılardan, eğitim ve test kümesinin oluşturulmasına kadar olan süreç ayrıntılı bir şekilde gösterilmiştir.



Şekil 2. Çalışmanın Akış Diyagramı

3.1. MQTTset Veri Kümesi

MQTTset veri kümesi, MQTT ve CoAP protokollerini temel alan ağları taklit edebilen bir ağ trafiği oluşturma aracı olan IoT-Flock kullanılarak oluşturulmuştur (Vaccari ve diğ., 2020). Oluşturulan MQTT trafiği, MQTTset verilerinin oluşturulması sırasında yakalanan bir paket yakalama (PCAP) dosyası olarak temsil edilmiştir. Veri kümesi 11.915.716 ağ paketi ve toplam 1.093.676.216 bayt boyutuyla temsil edilmiştir. 3 saldırı, 1 normal ve 1 bozulmuş olmak üzere 5 farklı sınıf içeren MQTTset veri kümesinin zenginleştirilmesi amacıyla bu çalışmada MQTTset verilerine ek olarak MITM saldırısı gerçekleştirilmiştir. Elde edilen veriler yeni bir sınıf olarak eğitim ve test verilerine eklenmiştir. Tablo 2'de MQTTset eğitim kümesi ve yeni oluşturulan MITM saldırı sınıfı için paket sayıları verilmiştir. Akıllı ev ağına uygulanan saldırılar ile toplanan veriler MQTTset veri kümesine dahil edilmiştir. Tablo 2'de paket sayıları ayrıştırılarak verilmiştir. Elde edilen eğitim kümesi hem MQTTset eğitim kümesini hem de ağdan toplanan verileri içermektedir. Test kümesi ise sadece akıllı ev ağından toplanan verilerden oluşmaktadır.

Tablo 2

MQTTset ve Ev Ağından Toplanan Eğitim-Test Kümelerinin Paket Sayıları

Saldırı Adı	MQTTset Eğitim Kümesi	Ev Ağından Toplanan Eğitim Kümesi	Ev Ağından Toplanan Test Kümesi
DoS Saldırısı	1,350,000	50,000	1,000
SlowITe Saldırısı	1,350,000	50,000	1,000
Kaba Kuvvet Saldırısı	1,350,000	50,000	1,000
Normal Veriler	7,000,000	5,000	1,000
Bozulmuş Veriler	1,350,000	50,000	1,000
MITM Saldırısı	-	5,000	1,000

DoS Saldırıları (Denial of Service): DoS Saldırıları, internete bağlı bir hostun hizmetlerini geçici veya süresiz olarak aksatarak, bir makinenin veya ağ kaynaklarının asıl kullanıcılar tarafından ulaşılamamasını hedefleyen bir siber saldırıdır. DoS genellikle hedef makine veya kaynağın, gereksiz talepler ile aşırı yüklenmesi ve bazı ya da bütün meşru taleplere doluluktan kaynaklı engel olunması şeklinde gerçekleştirilir. Bu saldırıyı uygulamak ve MQTT hizmetlerinin yükünü test etmek için MQTT-malaria tool kullanılarak eğitim kümesini yazarlar oluşturmuştur (Vaccari ve diğ., 2020).

Tablo 3

MQTTset Veri Kümesi Niteliklerin Listesi (Vaccari ve diğ., 2020)

No	Nitelik Adı	Açıklama	Protokol Katmanı
1	tcp.flags	TCP Flags	TCP
2	tcp.time_delta	Time TCP stream	
3	tcp.len	TCP Segment Length	
4	mqtt.conack.flags	Acknowledge Flags	MQTT
5	mqtt.conack.flags.reserved	Reserved	
6	mqtt.conack.flags.sp	Session Present	
7	mqtt.conack.val	Return Code	
8	mqtt.conflag.cleansess	Clean Session Flag	
9	mqtt.conflag.passwd	Password Flag	
10	mqtt.conflag.qos	QoS Level	
11	mqtt.conflag.reserved	Reserved	
12	mqtt.conflag.retain	Will Retain	
13	mqtt.conflag.uname	User Name Flag	
14	mqtt.conflag.willflag	Will Flag	
15	mqtt.conflags	Connect Flags	
16	mqtt.dupflags	DUP Flag	
17	mqtt.hdrflags	Header Flags	
18	mqtt.kalive	Keep Alive	
19	mqtt.len	Message Length	
20	mqtt.msg	Message	
21	mqtt.msgid	Message Identifier	
22	mqtt.msgtype	Message Type	
23	mqtt.proto_len	Protocol Name Length	
24	mqtt.protoname	Protocol Name	
25	mqtt.qos	QoS Level	
26	mqtt.retain	Retain	
27	mqtt.sub.qos	Requested QoS	
28	mqtt.suback.qos	Granted QoS	
29	mqtt.ver	Version	
30	mqtt.willmsg	Will Message	
31	mqtt.willmsg_len	Will Message Length	
32	mqtt.willtopic	Will Topic	
33	mqtt.willtopic_len	Will Topic Length	

MQTT Yayın Seli: IoT cihazı, periyodik olarak kötü amaçlı MQTT verileri göndererek, sunucunun tüm kaynaklarını ele geçirmektedir. Diğer saldırılardan farklı olarak, birden çok bağlantı yerine tek bir bağlantı kullanarak kaynakları doyurmaya çalışır. Bu saldırı durumda IoT kilidi aracı içindeki bir modül kullanılarak oluşturulur (Vaccari ve diğ., 2020).

SlowITe Saldırıları: MQTT protokolünü hedefleyen SlowITe (Slow DoS against Internet of Things Environments) saldırıları yavaş DoS saldırıları olarak da bilinmektedir. Bu tür tehdit, Yavaş DoS Saldırıları kategorisine girer ve minimum saldırı bant genişliğini ve hizmet reddi uygulayan bir ağ hizmetini hedeflemek için kaynakları kullanmaktadır. SlowITe'nin amacı, MQTT aracısının aynı anda yönetebildiği tüm kullanılabilir bağlantıları ele geçirmek için sunucuyla yüksek sayıda bağlantıyı sağlamaktır (Vaccari, Aiello ve Cambiaso, 2020).

Bozulmuş Veriler: Hatalı biçimlendirilmiş verileri oluşturmak için MQTTSA aracını kullanmışlardır. MQTT aracısında istisnaları artırmak için kurbana hatalı biçimlendirilmiş bağlantı veya yayın paketleri gönderilmiştir (Vaccari ve diğ., 2020).

Kaba Kuvvet Saldırıları: Kaba Kuvvet Saldırıları'nda (Brute Force) saldırganın amacı, kimlik doğrulama aşamasında kullanıcıların kimlik bilgilerini kırmaktır. MQTTSA aracı ile bu işlem yazarlar tarafından gerçekleştirilmiştir (Vaccari ve diğ., 2020).

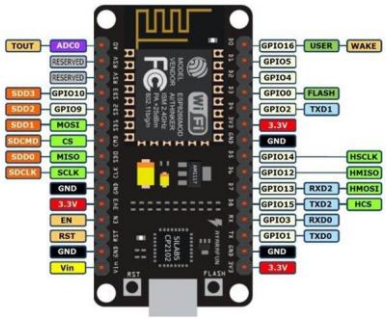
MITM Saldırıları: Saldırganın, birbiri ile doğrudan iletişim kuran iki taraf arasındaki iletişimi gizlice ilettiği veya değiştirdiği saldırı türüdür (MITM, 2022).

Tablo 3'te MQTTset veri kümesinde yer alan niteliklerin adı, açıklamaları ve protokol katmanı verilmiştir.

3.2. Kullanılan Donanımlar

Veri kümesinin oluşturulması amacıyla gerçekleştirilen IoT akıllı ev ağı mimarisinin (Şekil 1) oluşturulmasında kullanılan sensör ve modüller bu bölümde detaylandırılmıştır. Uygulamada ESP8266 Wifi modülü, DHT 11 sıcaklık ve nem sensörü, HC-SR04 mesafe sensörü ile ev ağı oluşturulmuştur.

ESP8266 Wifi Modülü: ESP8266 ve 1 MB lık dahili flaş bellekten oluşan tek çip ile Wi-Fi bağlantısı oluşturmaya olanak sağlamaktadır (ESP8266, 2022). Bu modül Arduino gibi geliştirici kartlarla kullanılabildiği gibi üzerinde bulunan işlemciyle programlanarak herhangi bir geliştirici kart olmadan da kullanılabilir. Bu çalışmada Arduino IDE kullanılarak ESP8266 modülü programlanmıştır. Şekil 3'te ESP8266 modülüne ait pin şeması verilmiştir.



Şekil 3. ESP8266 Wifi Modül Pin Şeması (Dönmez, 2022)

Arduino IDE: Arduino için Entegre Geliştirme Ortamı (IDE), C ve C ++ dilleri ile yazılmış bir platformlar arası uygulamadır (Linux, macOS, Windows için). Arduino IDE (Integrated Development Environment) temelde, çalıştırılabilir kodu hexadecimal format ile metin dosyasına işler. Ardından kullandığımız Arduino IDE'si bu metin dosyasını bağlı olan Arduino kartına aygıt yazılımına yükleyici program ile aktarmayı gerçekleştirir (Arduino, 2022).

DHT11 Sıcaklık ve Nem Sensör: Bu çalışmada sıcaklık ve nem sensörü (DHT11) ve ultrasonik mesafe sensörü (HC-SR04) tercih edilmiştir. DHT11, ortamdaki sıcaklığı ve nemi ölçmek amacıyla kullanılan bir sensördür. Ortam sıcaklığını çeşitli ısı birimlerinde (celcius, farhenayt, kelvin) elde etmek ve ortamdaki nemi % cinsinden ölçmek için kullanılmaktadır (DHT11, 2022). Akıllı ev projelerinde genellikle DHT11 sensörü kullanıldığı için tercih edilmiştir. Bu sensörden verileri okumak için dht kütüphanesi kullanılmıştır. Şekil 4'te DHT11 sensörünün pin şeması verilmiştir.

Şekil 4. DHT11 Sensör Pin Şeması

HC-SR04 Mesafe Sensörü: Arduino gibi geliştirme kartlarında kullanılmak üzere tasarlanmış olan bu sensör, kullanım kolaylığından dolayı çoğu robotik projelerinde yer alan bir mesafe ölçüm sensörüdür. Bu sensör 2 cm- 400 cm arasında %1 hata payıyla ölçüm yapabilmektedir. En verimli ölçüm yaptığı mesafe 2 – 200 cm arasındadır (İzgöl ve Kerem (2022)). Akıllı ev projesinde kullanılacağı için bu sensör yeterli mesafe aralığına sahiptir. Ultrasonik sensörü tetiklemek ve alınan sinyal darbe genişliğini otomatik ölçmek için NewPing kütüphanesi kullanılmıştır. Şekil 5'te mesafe sensörünün pin şeması verilmiştir.

Şekil 5. HC-SR04 Sensör Pin Şeması

3.3. Saldırı ve İzleme Araçları

Bu çalışmada HOIC, DirSearch, Slowloris, Cain ve Abel saldırı araçları tercih edilmiştir. Saldırılar Windows ortamında kurulan IoT ev ağına gerçekleştirilmiştir.

HOIC Saldırı Aracı: High Orbit Ion Cannon (HOIC), aynı anda 256 URL'ye kadar saldırmak için tasarlanmış açık kaynaklı bir ağ stres testi ve hizmet reddi saldırı uygulamasıdır (HOIC, 2022). Hedef URL'ye yapılan hizmet reddi (DoS) saldırısı, siteyi aşırı yükleme ve onu indirme girişiminde aşırı trafik göndererek gerçekleştirilir.

DirSearch Kaba Kuvvet Saldırı Aracı: Bu çalışmada, kaba kuvvet saldırısı gerçekleştirmek için DirSearch aracı kullanılmıştır (Dirsearch, 2022). Açık kaynak olan bu saldırı aracı maurosoria ve shelld3v tarafından geliştirilmiştir. Saldırı aracı ile MQTT sunucusuna saldırı yapılmış ve Wireshark ile bu saldırı izlenmiştir. Saldırı sonucunda Tablo 2'deki niteliklere ait değerler CSV formatında kaydedilmiştir.

Slowloris Saldırı Aracı: Slowloris, temelde iş parçacıklı sunucuları etkileyen bir HTTP Hizmet Reddi saldırısıdır (Slowloris, 2022). Bu saldırıda sunucu iş parçacığı havuzunu tüketir ve sunucu diğer kişilere yanıt veremez hale gelmektedir. Bu çalışmada Python ile geliştirilmiş Slowloris saldırı aracı kullanılmıştır. Saldırı Wireshark ile izlenmiş ve değerler CSV formatında kaydedilmiştir.

MQTTSA Saldırı Aracı: MQTT tabanlı ortamlardaki yanlış yapılandırmaları otomatik olarak değerlendirerek ve olası güvenlik açıklarını doğal dil açıklamalarından farklı düzeylerde bir rapor sunarak IoT geliştiricilerinin güvenlik farkındalığını artırmak için tasarlanmış bir araçtır (MQTTSA, 2022). Bu saldırı aracı ile Koklama (Sniffing) Saldırıları, Kaba Kuvvet Saldırıları, Bilgilerin Açıklanması (Information Disclosure), Hatalı Biçimlendirilmiş Veriler (Malformed) ve Hizmet Reddi (Denial of Service) Saldırıları gerçekleştirilmektedir. MQTTSA aracı ile test verilerinde yer alan hatalı biçimlendirilmiş veriler elde edilmiştir.

Cain & Abel Saldırı Aracı: Cain ve Abel; Kaba Kuvvet, ARP zehirlenmesi ve MITM Saldırıları gibi birçok saldırıyı gerçekleştirmek için kullanılan Windows tabanlı bir araçtır. MITM, iki bağlantı arasındaki iletişimin dinlenmesi ile çeşitli verilerin ele geçirilmesi veya iletişimi dinlemekle kalmayıp her türlü değişikliğin yapılmasını da kapsayan bir saldırı yöntemidir (Cain ve Abel, 2022). Bu çalışmada MQTTset veri kümesinde yer alan eğitim ve test verilerine MITM saldırıları sonucunda kaydedilen veriler de eklenmiştir.

Wireshark İzleme Aracı: Bu çalışmada, izleme aracı olarak Wireshark kullanılmıştır. Wireshark, kullanıcının ağ arabirimi denetleyicilerini rastgele moda koymasına olanak tanır (Wireshark, 2022). MQTTset veri kümesinde yer alan nitelikler, saldırı anında ve normal zamanda Wireshark ile veriler toplanmıştır. Yapılan saldırılar Wireshark ile izlenerek saldırı anı tespit edilmiş ve CSV formatında bütün saldırılar sonucunda değerler kaydedilmiştir.

Tablo 4

Sensörler ve İzleme Yöntemleri

Sensör	IP Adresi	Zaman	Konu	İzleme Yöntemi
DHT 11 Sensör	192.168.0.32	Periyodik Davranış	Sıcaklık, Nem	Serial Monitör, Wireshark
HC-SR04 Sensör	192.168.0.32	Rastgele Davranış	Uzaklık	Serial Monitör, Wireshark

Tablo 4'te sunulan sensörler ve izleme yöntemleri ile DoS, MITM, SlowITe ve Kaba Kuvvet saldırıları akıllı ev ağına gerçekleştirilerek veriler toplanmıştır. MITM saldırısı ile MQTTset veri kümesine Tablo 5'te yer alan nitelikler hem eğitim hem de test kümesine eklenmiştir. Ayrıca DoS, SlowITe ve Kaba Kuvvet saldırıları ile Tablo 2'de yer alan eğitim ve test kümeleri MQTTset veri kümesine dahil edilmiştir.

Tablo 5

MQTTset Veri Kümesi Eklenen Nitelikler

İsim	Açıklama	Protokol Katmanı
ipv6.plen	IPv6 Payload Length	IPV6
ipv6.nxt	IPv6 Next Header	IPV6
tcp.srcport	Source Port	TCP
tcp.dstport	Destination Port	TCP
eth.src	Source	TCP/IP

3.4. Python Programlama Dili

Python, nesne yönelimli, yorumlamalı, modüler ve etkileşimli yüksek seviyeli bir programlama dilidir. Python ile sistem programlama, kullanıcı arabirimi programlama, ağ programlama, web programlama, uygulama ve veri tabanı yazılımı programlama gibi birçok alanda yazılım geliştirilmektedir (Python,2022). Bu çalışmada kullanılan kütüphaneler Tablo 6'da verilmiştir.

Tablo 6

Kütüphaneler ve Kullanım Amaçları

Kütüphane	Kullanım Amacı
Scikit-Learn (sklearn)	Makine öğrenimi, Naive Bayes ve Karar Ağaçları sınıflandırma/eğitim ve test işlemleri
Pandas	Veri analizi
Numpy	Matematiksel hesaplama
Keras	Model tanımlama ve eğitim
Tensorflow	Sinir ağları eğitim ve test işlemleri
Matplotlib	Veri görselleştirme
Warnings	Uyarı mesajlarını görüntüleme

3.5. Algoritmalar

Bu çalışmada Naive Bayes, Rastgele Karar Ormanları, Karar Ağaçları ve Yapay Sinir Ağları kullanılarak sınıflandırma, eğitim ve test işlemleri gerçekleştirilmiştir.

Naive Bayes Algoritması: Naive Bayes Sınıflandırması, makine öğreniminde öğreticili öğrenme alt sınıfındadır. Daha açık bir ifadeyle sınıflandırılması gereken sınıflar ve örnek verilerin hangi sınıflara ait olduğu bellidir. Sınıflandırma işleminde genel olarak elde bir örüntü vardır. Buradaki işlem de bu örüntüyü daha önceden tanımlanmış sınıflara sınıflandırmaktadır. Her örüntü nicelik (Feature ya da parametre) kümesi tarafından temsil edilir (Naive Bayes, 2022). Az eğitim veri kümesiyle başarılı sonuçlar verebildiği için bu sınıflandırıcı seçilmiştir. Bu sınıflandırıcı ile Sklearn kütüphanesi (GaussianNB) kullanılarak eğitim ve test işlemleri gerçekleştirilmiştir.

Naive Bayes teoreminin formül ile gösterimi (1) verilmiştir.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{1}$$

P(A|B) ; B olayı gerçekleştiği durumda A olayının meydana gelme olasılığıdır. P(B|A) ; A olayı gerçekleştiği durumda B olayının meydana gelme olasılığıdır. P(A) ve P(B) ; A ve B olaylarının önsel olasılıklarıdır. Burada önsel olasılık Bayes teoreminine öznellik katar. Diğer bir ifadeyle örneğin P(A); henüz veri toplanmadan A olayı hakkında sahip olunan bilgidir (Naive Bayes, 2022).

Matematiksel bir ifadeyle Naive Bayes sınıflandırması

$$P(x|S_i)P(S_i) > P(x|S_j)P(S_j), \forall j \neq, i \tag{2}$$

ifadesindeki $P(x|S_i)$ terimi aşağıdaki gibi yeniden yazılır.

$$P(S_i)\prod_{k=1}^{L} P(x_k|S_i) > P(S_j)\prod_{k=1}^{L} P(x_k|S_j) \tag{3}$$

$P(S_i)$ ve $P(S_j)$ i ve j sınıflarının öncel olasılıklarıdır.

Rastgele Orman Algoritması: Rastgele Orman Algoritması (Random Forest); sınıflandırma, regresyon ve diğer görevler için, eğitim aşamasında çok sayıda karar ağacı oluşturarak problemin tipine göre sınıf veya sayı (Regresyon) tahmini yapan bir toplu öğrenme yöntemidir. Rastgele Ormanlar, karar ağaçlarının eğitim setlerine aşırı uyma problemlerini gidermektedir (Rastgele Orman, 2022). Bu sınıflandırıcı ile Sklearn kütüphanesi (Random Forest Classifier) kullanılarak eğitim ve test işlemleri gerçekleştirilmiştir. Sınıflandırma verilerine dayalı Rastgele Orman gerçekleştirirken, genellikle Gini indeksini veya bir karar ağacı dalında düğümlerin nasıl olacağına karar vermek için kullanılan formülü (4) verilmiştir.

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2 \tag{4}$$

Bu formül, bir düğümdeki her dalın Gini'sini belirlemek için sınıfı ve olasılığı kullanır ve dallardan hangisinin daha olası olduğunu belirler. Burada p_i, veri setinde gözlemlediğiniz sınıfın göreli frekansını, c ise sınıf sayısını temsil eder (Schott, 2022).

$$Entropy = \sum_{k} - p_i * \log_2(p_i) \tag{5}$$

Entropy, Gini indeksinden farklı olarak, hesaplanmasında logaritmik fonksiyon kullanılmaktadır (Schott, 2022).

Karar Ağaçları: Karar Ağaçları, en çok kullanılan gözetimli öğrenme algoritmalarındandır. Genel itibariyle ele alınan bütün problemlerin (Sınıflandırma ve regresyon) çözümüne uyarlanabilirler. Karar Ağaçları (Decision Tree), Rastgele Orman, Gradyen Güçlendirme (Gradient Boosting) gibi yöntemler, her türlü veri bilimi problemlerinde yaygın bir şekilde kullanılmaktadırlar (Ulgen, 2022). Karar Ağacı Algoritması, veri setini küçük parçalara bölerek geliştirilmiştir. Bu sınıflandırıcı ile Sklearn kütüphanesi

(Decision Tree Classifier) kullanılarak eğitim ve test işlemleri gerçekleştirilmiştir.

Karar Ağaçları Entropy formulü (6) aşağıda verilmiştir.

$$H(X) = E\big(I(X)\big) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i) \quad (6)$$

$I(X)$; rastgele bir değişken olan X'in bilgi içeriğidir. $p(x_i)$; X'in olasılık kütle fonksiyonudur.

Yapay Sinir Ağları: Gözetimli öğrenme algoritmalarından olan Yapay Sinir Ağları, eş zamanlı çalışarak karmaşık işleri gerçekleştirebilmektedir. Örüntü tanıma, sınıflandırma yaparak. eksik örüntüleri tamamlayabilirler. Yapay Sinir Ağları başlıca teşhis, sınıflandırma, tahmin, kontrol, veri ilişkilendirme, veri filtreleme, yorumlama gibi alanlarda kullanılmaktadır (Yıldırım, 2022). Çok katmanlı Perceptron Sınıflandırıcısı olan Sklearn kütüphanesi (MLPClassifier) ile parametreleri güncelleme ve parametrelere göre kayıp fonksiyonunun hesaplanması sağlanmıştır.

3.6. Performans Değerlendirmesi

Veri kümesi sınıflandırma sonucunda performansını karşılaştırmak için bu çalışmada Doğruluk (ACC) ve F1 skor değerleri kullanılmıştır (Accuracy and Precision, 2022).

Doğruluk:

Doğruluk değeri, doğru tahmin edilen alanların, toplam veri kümesine oranı ile hesaplanır.

Doğru Pozitif (DP): Gerçekte pozitif (saldırı) olan ve tahmin edildiğinde de pozitif (saldırı) olarak sınıflandırılan örnekleri ifade etmektedir.

Yanlış Negatif (YN): Gerçekte pozitif (saldırı) olan ve tahmin edildiğinde de negatif (normal) olarak sınıflandırılan örnekleri ifade etmektedir.

Yanlış Pozitif (YP): Gerçekte negatif (normal) olan ve tahmin edildiğinde de pozitif (saldırı) olarak sınıflandırılan örnekleri ifade etmektedir.

Doğru Negatif (DN): Gerçekte negatif (normal) olan ve tahmin edildiğinde de negatif (normal) olarak sınıflandırılan örnekleri ifade etmektedir. Denklem (7)'de bu çalışmada kullanılan başarım ölçütü verilmiştir (Burukanlı, Çıbuk ve Budak (2021)).

$$Doğruluk = \frac{DP + DN}{DP + YN + YP + DN} \quad (7)$$

F1 Skor:

F1 Skor, kesinlik ve duyarlılık değerlerinin harmonik ortalamasını vermektedir. F1 skor'un doğruluk değerinden farkı; eşit dağılmayan veri kümelerinde hatalı bir model seçimi yapmamaktır (F-score, 2022).

$$PRC = \frac{DP}{DP + YP} \quad (8)$$

$$REC = \frac{DP}{DP + YN} \quad (9)$$

PRC ve REC değerleri (8) ve (9)'da verilmiştir. Bu değerler (10)'da yerine yazılarak F1 Skor değeri hesaplanmaktadır.

$$F1\ Skor = 2 * \frac{PRC * REC}{PRC + REC} \quad (10)$$

Bu çalışmada araştırma ve yayın etiğine uyulmuştur.

4. Bulgular

Kaggle, analiz, metriklerin kullanımı ve makine öğrenmesi algoritmalarının test edilmesi için bir platformdur. MQTTset veri kümesi Vaccari ve diğ., (2020) tarafından eğitim ve test kümesi olarak ikiye ayrılmıştır. Bu çalışmada veri kümesi ve algoritmalar ile eğitim ve test sürelerinin hesaplanması sağlanmıştır. Naive Bayes, Karar Ağaçları, Rastgele Ormanlar ile sınıflandırma işlemi yapılmış ve sonucunda test örneklerinin ne kadarlık bir zaman süresinde sınıflandırıldığı tespit edilmiştir. Sınıflandırıcıları eğitim ve test sürelerini hesaplamak için Python kütüphaneleri kullanılmıştır.

DHT11 sensörü ile odadaki sıcaklık ve nem bilgilerinin anlık kontrolü sağlanmıştır. Bu sensör dışında birçok sensör bu akıllı ev ağında kullanılabilir. DHT11 sensörüne ek olarak akıllı ev ağında HC-SR04 (Ultrasonik sensör) da kullanılmıştır. Veriler aracı (broker)'dan okunarak Wireshark aracı ile izlenmesi sağlanmıştır. Modül, aracı (broker)'daki sıcaklık, nem ve mesafe konu (Topic)'larına abone (Subscribe) olacaktır. Bu modülün MQTT Broke'a bağlanması için PubSubClient kütüphanesi tercih edilmiştir.

Tablo 5'teki niteliklerin veri kümesine eklenmesi sonucunda Veri kümesinden elde edilen sonuçlara göre Karar Ağaçları Algoritması'nın Naive Bayes'ten daha yüksek doğruluk oranına sahip olduğu görülmüştür.

Karar Ağaçlarının eğitim süresi daha uzun iken test süresi diğer algoritmaya göre daha kısadır. Tablo 7'de MQTTset veri kümesi ile yapılan testlerin sonuçları verilmiştir.

Tablo 7

MQTTset Veri Kümesi ile Yapılan Test Sonuçları

Algoritma	Doğruluk	F1 Skor	Eğitim Süresi (s)	Test Süresi (s)
Karar Ağaçları	0.903	0.900	316	78
Naive Bayes	0.643	0.758	251	76
Yapay Sinir Ağları	0.849	0.861	778	130
Rastgele Orman	0.902	0.900	1500	85

Tablo 8'de MQTTset veri kümesine yeni eklenen verilerle birlikte yapılan testlerin sonuçları verilmiştir. Tablo 7 ve 8 incelendiğinde MQTTset veri kümesine eklenen veriler ile doğruluk ve F1 skor değerleri artarken, eğitim ve test sürelerinde artış gözlenmektedir.

Tablo 8

MQTTset Veri Kümesine Eklenen Verilerle Yapılan Test Sonuçları

Algoritma	Doğruluk	F1 Skor	Eğitim Süresi (s)	Test Süresi (s)
Karar Ağaçları	0.956	0.904	346	72
Naive Bayes	0.922	0.825	271	75
Yapay Sinir Ağları	0.815	0.851	780	75
Rastgele Orman	0.915	0.858	2500	95

5. Sonuçlar

Sunulan çalışma akıllı ev güvenlik ağının bir parçası olduğundan ileride farklı sensörler ve modüller ile genişletilebilir yapıdadır. MQTT ile haberleşme sağlayan bu ağda mesajların güvenilirliği test edilmiştir. MQTT protokolü kablosuz sensörlerin ağ ortamında çalışma kolaylığı sebebiyle tercih edilmiş olup asenkron haberleşmektedir. Geliştirilen uygulamada ve testler sonucunda MQTT protokolünün saldırılara karşı açık olduğu ortaya konulmuştur. Saldırılar herhangi bir ağa gerçekleştirilebileceği için veri kayıplarına ve sızma işlemlerine karşı savunmasızdır. MQTTset veri kümesindeki niteliklerin eklenmesi ve saldırı çeşidinin artırılması ile doğruluk ve skor değerlerinde artış gözlenmiştir. Ayrıca akıllı ev ağına gerçekleştirilen saldırıların sonucunda veriler ve ağda izlenen bozulmuş ve normal verilerin eklenmesi ile MQTTset veri kümesi genişletilmiştir. Bu veri kümesi ve saldırıların Windows ortamında gerçekleştirilmesi ile literatüre katkı sağlandığı düşünülmüştür. İlerleyen çalışmalarda bu saldırılara karşı önlemler alınması ve daha fazla saldırı aracı ile testlerin gerçekleştirilmesi amaçlanmaktadır. Ayrıca MQTTset veri kümesi dışında yer alan veri kümeleri ile de çalışmalar yapılmalıdır. Bu çalışma ile ileride bu konu üzerine çalışma yapacak araştırmacılar için geniş ve pratik bilgiler verilmiştir.

Araştırmacıların Katkısı

Bu araştırmada; Ayça Nur KAHYA, test ortamının kurulması, bilimsel yayın araştırması, makalenin oluşturulması, sonuçların değerlendirilmesi ve yorumlanması; Esra Nergis YOLAÇAN, bilimsel yayın araştırması, makalenin oluşturulması konularında katkı sağlamışlardır.

Çıkar Çatışması

Yazarlar tarafından herhangi bir çıkar çatışması beyan edilmemiştir.

Kaynaklar

Accuracy and Precision (2022). Vikipedi içinde. Erişim Adresi: https://en.wikipedia.org/wiki/Accuracy_and_precision

Agrawal, D., Agrawal, C., ve Yadav, H. (2020) A Machine Learning Based Intrusion Detection Framework Using KDDCUP 99 Dataset. International Journal of Innovative Research in Technology and Management, 4(6), 179-189. Erişim adresi: http://www.ijirtm.com/UploadContaint/finalPaper/IJIRTM-0406202025.pdf

Arduino (2022). Vikipedi içinde. Erişim Adresi: https://tr.wikipedia.org/wiki/Arduino_IDE

Aveleira-Mata, J., ve Alaiz-Moreton, H. (2019). Functional Prototype for Intrusion Detection System Oriented to Intelligent IoT Models. In International Symposium on Ambient Intelligence 179-186. doi:https://dx.doi.org/10.1007/978-3-030-24097-4_22

Burukanlı, M., Çıbuk, M., & Budak, Ü. (2021) Saldırı Tespiti için Makine Öğrenme Yöntemlerinin Karşılaştırmalı Analizi. Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, 10(2), 613-624.

Cain&Abel (2022). Github içinde. Erişim Adresi: https://github.com/xchwarze/Cain

Canedo, J., ve Skjellum, A. (2016). Using machine learning to secure IoT systems. In 2016 14th annual conference on privacy, security and trust. 219-222. IEEE.doi:https://dx.doi.org/10.1109/PST.2016.7906930

Ciklabakkal, E., Donmez, A., Erdemir, M., Suren, E., Yilmaz, M. K., ve Angin, P. (2019). ARTEMIS: An intrusion detection system for MQTT attacks in internet of things. In 2019 38th Symposium on Reliable Distributed Systems (SRDS). 369-3692. IEEE.doi:https://dx.doi.org/10.1109/SRDS47363.2019.00053

DHT11 (2022). Erişim Adresi: https://www.arduinomedia.com/arduino-ile-dht11-sicaklik-ve-nem-sensoru-kullanimi/

Dirsearch. (2022). Github içinde. Erişim Adresi: https://github.com/maurosoria/dirsearch

Dönmez, M. (2022). ESP8266 üzerindeki hangi GPIO pinleri kullanılabilir. Erişim Adresi: https://www.muratdonmez.com.tr/esp8266-uzerindeki-hangi-gpio-pinleri-kullanilabilir/

ESP8266 (2022). Vikipedi içinde. Erişim Adresi: https://tr.wikipedia.org/wiki/ESP8266

F-Score (2022). Vikipedi içinde. Erişim Adresi: https://en.wikipedia.org/wiki/F-score

GSL (2022). IoT Uygulamalarında Verimli ve Esnek Bir Haberleşme Yapısı. Erişim Adresi: https://www.gsl.com.tr/mqtt-endustriyel-nesnelerin-interneti-uygulamalarinda-verimli-ve-esnek-bir-haberlesme-yapisi.html

Gupta, V., Khera, S., ve Turk, N. (2021). MQTT protocol employing IOT based home safety system with ABE encryption. Multimedia Tools and Applications, 80(2), 2931-2949. doi: https://dx.doi.org/10.1007/s11042-020-09750-4

Hindy, H., Bayne, E., Bures, M., Atkinson, R., Tachtatzis, C., ve Bellekens, X. (2020). Machine learning based IoT Intrusion Detection System: an MQTT case study (MQTT-IoT-IDS2020 Dataset). In International Networking Conference. 73-84. doi: https://dx.doi.org/10.1007/978-3-030-64758-2_6

HOIC (2022). Vikipedi içinde. Erişim Adresi: https://en.wikipedia.org/wiki/High_Orbit_Ion_Can

İzgöl, Kerem. (2022). Maker Robotistan. Erişim Adresi: https://maker.robotistan.com/arduino-dersleri-19-hc-sr04-ultrasonik-mesafe-sensoru-kullanimi/

Karande, J., & Joshi, S. (2021). EADA: An Algorithm for Early Detection of Attacks on IoT Resources. International Journal 10.1. doi: https://dx.doi.org/10.30534/ijatcse/2021/161012021

Khraisat, A., ve Alazab, A. (2021). A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. Cybersecurity, 4(1), 1-27. doi: https://dx.doi.org/10.1186/s42400-021-00077-7

Manzoor, I., ve Kumar, N. (2017). A feature reduced intrusion detection system using ANN classifier. Expert Systems with Applications, 249-257.doi:https://dx.doi.org/10.1016/j.eswa.2017.07.005

Mısır, O., & Gökrem, L. (2020). Nesnelerin İnterneti için MQTT ile Hiyerarşik Haberleşme. Journal of New Results in Engineering and Natural Sciences, (12), 1-11. Erişim Adresi: https://dergipark.org.tr/tr/download/article-file/1176460

MITM (2022). Vikipedi içinde. Erişim Adresi: https://tr.wikipedia.org/wiki/Man-in-the-middle_saldırısı

MQTTSA (2022). Sites Google. Erişim Adresi: https://sites.google.com/fbk.eu/mqttsa/home

Naïve Bayes (2022). Vikipedi içinde. Erişim: Adresi: https://en.wikipedia.org/wiki/Naive_Bayes_classifier

Oral, O., ve ÇAKIR, M. (2017). Nesnelerin İnterneti Kavramı ve Örnek Bir Prototipin Oluşturulması. Mehmet Akif Ersoy Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 172-177. Erişim Adresi: https://dergipark.org.tr/tr/download/article-file/328807

Python (2022). Vikipedi içinde. Erişim Adresi: https://tr.wikipedia.org/wiki/Python_(programlama_dili)

Schott, Madison. (2022). Random Forest Algorithm for Machine Learning. Erişim Adresi: https://medium.com/capital-one-tech/random-forest-algorithm-for-machine-learning-c4b2c8cc9feb

Rastgele Orman (2022). Vikipedi içinde. Erişim Adresi: https://tr.wikipedia.org/wiki/Rastgele_orman

Shalaginov, A., Semeniuta, O., ve Alazab, M. (2019). MEML: resource-aware MQTT-based machine learning for network attacks detection on IoT edge devices. In Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing Companion, 123-128. doi: https://dx.doi.org/10.1145/3368235.3368876

Slowloris (2022). Github. Erişim Adresi: https://github.com/gkbrk/slowloris

Ulgen, Kaan. (2022). Karar Ağaçları. Erişim Adresi: https://medium.com/@k.ulgen90/makine-öğrenimi-bölüm-5-karar-ağaçları-c90bd7593010

Vaccari, I., Aiello, M., ve Cambiaso, E. (2020). SlowITe, a novel denial of service attack affecting MQTT. Sensors, 20(10), 2932. doi: https://dx.doi.org/10.3390/s20102932

Vaccari, I., Chiola, G., Aiello, M., Mongelli, M., ve Cambiaso, E. (2020). MQTTset, a new dataset for machine learning techniques on MQTT. Sensors, 20(22), 6578. doi: https://dx.doi.org/10.3390/s20226578

Wireshark (2022). Vikipedi içinde. Erişim Adresi: https://tr.wikipedia.org/wiki/Wireshark

Yıldırım, Elif. (2022). Veri Bilimi Okulu. Erişim Adresi: https://www.veribilimiokulu.com/yapay-sinir-agiartificial-neural-network-nedir/

Zhang, Y., Li, P., ve Wang, X. (2019). Intrusion detection for IoT based on improved genetic algorithm and deep belief network. IEEE Access,7, 31711-31722. doi: https://dx.doi.org/10.1109/ACCESS.2019.2903723