Bilgisayar Ağları 223 (2023) 109576

ScienceDirect'te bulunan içerik listeleri

Bilgisayar Ağları

dergi ana sayfası: www.elsevier.com/locate/comnet

IoT bağlamlarında ölçeklenebilir ve birlikte çalışabilir uç tabanlı federe öğrenme

Claudia Campolo [a,b,*], Giacomo Genovese [a,b], Gurtaj Singh [a,b], Antonella Molinaro [a,b,c]

[a] Reggio Calabria Mediterranea Üniversitesi, İtalya
[b] Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT), İtalya
[c] Laboratoire des Signaux et Systémes, CentraleSupélec, Université Paris-Saclay, Fransa

ARTICLEINF

Anahtar Kelimeler:
Federe öğrenme
MQTT
OMA LwM2M

OABSTRACT

Kitlesel olarak konuşlandırılan Nesnelerin İnterneti (IoT) cihazlarından gelen verilerin analizi, çeşitli dikey alanlarda sayısız akıllı uygulamanın önünü açmaktadır. Federated Learning (FL), Makine Öğrenimi (ML) modellerini eğitim prosedürlerinden sorumlu merkezi sunuculara taşımak yerine doğrudan veri üreten cihazların (FL istemcileri) üzerinde eğitmek için yakın zamanda öne çıkan bir çözüm olarak önerilmiştir. FL, özellikle gizliliğin korunması ve veri kümelerinin değişimi için ağ tıkanıklığının azaltılması açısından doğal faydalar sağlamaktadır. Son zamanlardaki büyük araştırma çabalarına rağmen, düşük iletişim ayak izi, sağlamlık ve birlikte çalışabilirliği etkin bir şekilde hedefleyen pratik bir uygulama için hala zorluklarla karşı karşıyadır. Bu boşluğu doldurmak için bu çalışmada, FL işlemlerini kolaylaştırmak ve FL istemcisi olarak hareket eden IoT cihazlarını idare etmeye daha uygun hale getirmek için Mesaj Kuyruğu Telemetri Aktarımı (MQTT) yayınlama/abone olma mesajlaşma protokolü ve Açık Mobil İttifak (OMA) Hafif Makineden Makineye (LwM2M) semantiği üzerine inşa edilmiş yeni ve kapsamlı bir çerçeve öneriyoruz. Önerinin uygulanabilirliği ve literatürdeki bir çözüme kıyasla iletişim verimliliği, farklı bağlantı ayarları altında ve farklı veri kümeleri için gerçekçi bir Kavram Kanıtı (PoC) aracılığıyla değerlendirilmektedir.

1. Giriş

Son yıllarda Nesnelerin İnterneti (IoT) cihazları dramatik bir hızla çoğaldı. Ericsson'a göre [1], 2050 yılına kadar birbirine bağlı 24 milyar cihaz olacak. Makine Öğrenimi (ML) tekniklerinin uygulanmasıyla, bu tür cihazlar tarafından üretilen büyük miktarda veri, ulaşımdan endüstriye, sağlık hizmetlerinden tarıma kadar farklı alanlarda zekanın kilidini açabilir.

Tipik olarak bir bulut sunucusuna yerleştirilen merkezi ML işlevleri, IoT veri patlaması göz önüne alındığında kritik ölçeklenebilirlik sınırlamalarına maruz kalabilir. Dahası, son cihaz verilerinin eğitim amacıyla üçüncü taraf bir sunucuya aktarılması gerekliliği nedeniyle kullanıcıların gizlilik sızıntısından muzdariptirler. Gerçekten de, toplanacak veriler izole adalar şeklinde bulunur ve örneğin sağlık uygulamaları söz konusu olduğunda gizli bilgiler içerebilir ve bu nedenle veri paylaşımı mümkün olmayabilir [2]. Endüstriyel IoT (IIoT) uygulamaları söz konusu olduğunda bu tür bir tehdit aynı işletme içinde bile ortaya çıkabilir [3].

Federated Learning (FL) kavramı yakın zamanda yukarıda bahsedilen sorunları ele almak için önerilmiştir [4,5]. Geleneksel makine öğrenimi tekniklerinden farklı olarak FL, FL *istemcisi olarak* hareket eden birden fazla cihazı, gerçek veri kümelerini paylaşmadan FL *toplayıcısı* olarak adlandırılan merkezi bir sunucu ile koordine ederek bir makine öğrenimi modelinin cihaz üzerinde eğitilmesini sağlar.

Eğitim prosedürlerine dahil olan FL istemci popülasyonu, farklı işlem kapasitelerine sahip ve FL toplayıcı ile model/parametre alışverişi yaparken farklı kanal koşullarını deneyimleyen heterojen cihazları kapsayabilir. FL istemcilerinin rastgele seçilmesi, eğitim yakınsama süresi ve/veya model doğruluğu açısından düşük performansa neden olabilir [6]. Hesaplama/pil kaynakları açısından kısıtlı ve/veya FL toplayıcı ile veri alışverişinde yavaş olan cihazlar genel eğitimi yavaşlatabilir. Bunun yerine, zayıf veri kümelerine sahip cihazlar modelin doğruluğunu tehlikeye atabilir. Bununla birlikte, akıllı istemci seçim şemaları öneren son zamanlardaki büyük literatüre rağmen, örneğin [6-11], birkaç çalışma FL istemcilerinin yeteneklerini keşfetmek ve bunları mantıklı bir şekilde seçmeyi sağlamak için gereken prosedürlere odaklanmıştır. Akıllı telefonlar, araçlar, dizüstü bilgisayarlar veya tabletler gibi IoT cihazları söz konusu olduğunda potansiyel FL istemcilerinin durumu hakkında bilgi almak daha da önemlidir [12,13]. Bir IoT istemcisi, hedef yakınsamaya beklenen süre içinde ulaşmak için her zaman cihaz üzerinde hesaplama yapamayabilir. Dahası, iletişim, hesaplama, depolama, güç ve enerji kullanımı açısından, örneğin başıboşluk sorunu gibi daha büyük zorluklarla karşılaşılması gerekir [13].

* Sorumlu yazar: Reggio Calabria Akdeniz Üniversitesi, İtalya.
E-posta adresleri: claudia.campolo@unirc.it (C. Campolo), giacomo.genovese@unirc.it (G. Genovese), gurtaj.singh@unirc.it (G. Singh), antonella.molinaro@unirc.it (A. Molinaro).

https://doi.org/10.1016/j.comnet.2023.109576
Alındı 31 Ekim 2022; Gözden geçirilmiş haliyle alındı 6 Ocak 2023; Kabul edildi 11 Ocak 2023
Çevrimiçi olarak 14 Ocak 2023'te kullanılabilir

7]'de potansiyel müşteriler FL toplayıcısına kaynakları (örneğin, işleme yetenekleri, veri kümesi, vb.) hakkında bildirimde bulunur. İkincisi, yerel olarak eğitilmiş modellerin güncellemelerinden başlayarak küresel modeli oluşturmak ve güncellemek için gereken süreyi tahmin etmek için alınan bilgilerden yararlanır. Ardından, istemciler gerçekleştirilen tahminlere göre seçilir. Benzer şekilde [11]'de, FL toplayıcısı tüm potansiyel müşterilere bir katılımcı istek mesajı iletir. Bunlar arasından eğitime katılmak isteyenler, veri dağılımlarını ve özel veri miktarlarını içeren bir yanıt mesajı gönderir. Ancak, yukarıda bahsedilen çalışmalarda bu tür bilgi alışverişi için kullanılan protokol tartışılmamaktadır.

14]'te aşağıdakilerden yararlanmayı önerdik: *(i)* potansiyel müşterilerin yeteneklerini almayı amaçlayan FL istemcileri-FL toplayıcı etkileşimleri için Message Queue Telemetry Transport (MQTT) protokolü [15] ve
(ii) Açık Mobil İttifak (OMA) Hafif Makineden Makineye (LwM2M) veri modelleri [16] bu tür yeteneklerin anlamsal tanımı için. Çözüm, uygulanabilirliğini teyit edecek şekilde ilk kez uygulanmıştır.

İstemci seçimini hedefleyen çalışmaların çoğunda ihmal edilen, istemcilerin yeteneklerini keşfetme amaçlı etkileşimlerin yanı sıra FL, seçilen FL istemcileri ile FL toplayıcı arasında kapsamlı bir iletişim gerektirir. Daha önce [17]'de tartışıldığı gibi, bu amaçla kullanılan iletişim protokolünün FL performansı (örneğin, modeli eğitmek için harcanan toplam süre açısından) ve ortaya çıkan iletişim ayak izi üzerindeki etkisi yeterince araştırılmamıştır.

18]'de FL toplayıcısı bir MQTT abonesi olarak hareket etmekte ve durumlarını bildirmek isteyen cihazlar (örneğin, bir modeli eğitmek için hazır olmak) MQTT yayıncıları olmaktadır. MQTT ayrıca [19]'da, bant genişliği verimliliği ve sock-et'lere [17] kıyasla daha düşük gecikme süresi açısından üstünlüğü ve küresel modelin dağıtımı sırasında ihtiyaç duyulduğu gibi bire-çok etkileşime uygunluğu göz önüne alındığında, model parametrelerinin değişimini düzenlemek için kullanılmaktadır.

Bu çalışma, yukarıda bahsedilen çözümlere değer vermekte ve aşağıdaki ana orijinal katkıları sağlayarak bunların ötesine geçmektedir:

- Yalnızca istemci keşif prosedürlerine odaklanan [14]'teki ön çalışmamızı, IoT cihazlarının varlığında FL eğitim prosedürünün tüm aşamalarını kapsayan daha kapsamlı bir hafif uç tabanlı FL çerçevesi ile genişletiyoruz.
- Birlikte çalışabilirliği hedeflemek ve IoT zorluklarıyla özel olarak başa çıkmak için OMA LwM2M semantiği ile güçlendirilmiş, önceden tanımlanmış konular ve mesaj yükleri ile MQTT'ye dayalı iletişim ilkelleri tasarlıyoruz.
- Farklı bağlantı ayarları altında ve farklı veri kümeleri için bir kıyaslama FL uygulaması ile karşılaştırıldığında, sanallaştırılmış ve fiziksel FL istemcileri ile bir Kavram Kanıtı (PoC) aracılığıyla önerinin performansını değerlendiriyoruz.

Bu çalışmanın geri kalanı aşağıdaki şekilde düzenlenmiştir. Bölüm 2, çalışmanın arka planını ve motivasyonlarını sunmaktadır. Öneri Bölüm 3'te tartışılmaktadır. Tasarlanan çerçeve tarafından öngörülen ana prosedürler Bölüm 4, Bölüm 5 ve Bölüm 6'da sunulmuştur. Bölüm 7'de tasarlanan çerçevenin temel faydaları tartışılmaktadır. Elde edilen sonuçlar Bölüm 8'de rapor edildikten sonra Bölüm 9'da gelecekteki çalışmalara ilişkin ipuçları verilmektedir.

2. Arka plan ve motivasyonlar

Bu bölümde, genel olarak federe öğrenmenin nasıl çalıştığını ve araştırmamızı motive eden ilgili açık konuları kısaca tanıtıyoruz.

2.1. Federe öğrenme: genel bakış

FL paradigması, dağıtık ve gizliliği artıran doğası sayesinde Yapay Zeka (AI) çözümlerini artırmak için 2016 yılında önerilmiştir [4].

FL'de, paylaşılan bir global model, birden fazla cihazdan gelen model güncellemelerinin birden fazla tur boyunca yinelemeli olarak toplanmasıyla eğitilir. Her turda FL istemcileri FL toplayıcısından en son global modeli indirir, modeli kendi yerel veri kümeleri üzerinde eğitir ve kendi model güncellemelerini FL toplayıcısına bildirir. Sonuncusu tüm yerel model güncellemelerini birleştirir ve yeni bir geliştirilmiş küresel model oluşturur. Model toplama işlemi uçta ya da bulutta gerçekleştirilebilir [12]. Süreç, küresel eğitim tamamlanana kadar yinelenir.

FL toplayıcı, dağıtılmış öğrenicilerin bilgi işlem kaynaklarından yararlanarak eğitim kalitesini artırabilir ve FL toplayıcıda eğitim için ham veri gerekmediğinden kullanıcı gizliliği sızıntısını en aza indirebilir. Bir başka sonuç olarak, veri boşaltma nedeniyle gecikmeler azalır ve veri alışverişi için ağ kaynaklarından tasarruf edilir. Buna ek olarak, farklı cihazlardan farklı veri kümelerini kullanarak, FL, merkezi bir yaklaşımla sağlananlarla karşılaştırılabilir doğruluk performansı elde edebilir.

2.2. FL sınırlamaları

Yukarıda bahsedilen avantajlar, FL'yi akıllı sağlık hizmetlerinden akıllı ulaşıma kadar çeşitli alanlarda sayısız akıllı ve gizliliği artırılmış IoT uygulaması oluşturmak için uygun bir aday haline getirmektedir [20].

Bununla birlikte FL, aralarında cihaz heterojenliği (örneğin, hesaplama kaynakları, mevcut güç açısından), kablosuz kanal belirsizlikleri ve dengesiz yerel veri kümeleri ve cihazlar arasında bağımsız olmayan ve aynı dağılım (IID olmayan) verileri nedeniyle veri eşitsizliği gibi çeşitli zorluklarla yüzleşmek zorundadır [12].

Orijinal FL uygulamasında [4] olduğu gibi yerel eğitim prosedürlerine katılan FL istemcilerinin rastgele seçilmesi, uzun yakınsama süresi sağlayabilir. Bu durum, FL toplayıcısının her turda seçilen FL istemcileri tarafından yüklenen güncellenmiş yerel ağırlık parametrelerini beklediği senkron beslemeli öğrenme için zararlı olabilir. Örneğin, düşük hesaplama kaynaklarına sahip veya kötü kablosuz kanal koşullarına sahip düğümlerin seçilmesi, genel FL eğitim süresinde bir artışa neden olur. IoT cihazları, pillerinin bitmesi veya uyku moduna geçmeleri durumunda eğitim görevini tamamlayamayabilir ve buna bağlı olarak model doğruluğunu etkileyebilir.

Literatürde birkaç tane daha sofistike istemci seçim şeması ortaya konmuştur, örneğin [6-11] bunlardan sadece birkaçıdır. FL müşterilerini seçmek için dikkate alınan kriter/ kriter kümesi açısından farklılık gösterirler. Bu tür şemaların etkinliği büyük ölçüde FL toplayıcının oluşturabileceği potansiyel müşterilerin kaynakları ve yetenekleri hakkındaki farkındalığa dayanmaktadır. Bununla birlikte, yukarıda bahsedilen çalışmalar genellikle bu tür bilgilerin nasıl alınabileceğini tartışmayı ihmal etmektedir. Bunun yerine, potansiyel FL istemcilerinin ve FL toplayıcısının bu tür verileri düşük bir iletişim ek yükü ile ve ilgili cihazların doğal heterojenliği ile başa çıkmak için tek tip bir şekilde değiş tokuş etmesini sağlamak için bir protokole ihtiyaç vardır.

İstemci bulma prosedürü sırasında ortaya çıkan iletişim ek yükü, dağıtılmış model eğitimi için gerekli olan genel veri alışverişinin yalnızca küçük bir miktarını temsil eder. Aslında, istemci seçiminden sonra, FL toplayıcı ve FL istemcilerinin önce orijinal modeli, ardından da güncellenmiş yerel ve global model parametrelerini birden fazla tur boyunca değiş tokuş etmesi gerekir [4].

Bu aşamalarda kullanılan iletişim protokolünün seçimi, özellikle

büyük model boyutları için FL performansını ve dolayısıyla ağ çalışmasını etkileyebilir [17], bu nedenle çalışmamızı motive eden uygun bir araştırmayı hak etmektedir.

3. Önerilen /L-MQTT çerçevesi: temel bilgiler

Bu çalışmada, uçta uygulanan bir *FL* aggre- gator ile FL istemcileri olarak hareket eden dağıtılmış IoT cihazları arasındaki etkileşimleri desteklemek için *FL-MQTT olarak* adlandırdığımız kapsamlı ve hafif bir çerçeve öneriyoruz.

Bu çerçeve, akıllı istemci seçim prosedürlerinin uygulanması amacıyla *FL istemci keşfinin* desteklenmesini özellikle hedefleyen [14]'teki ön çalışmamızın üzerine inşa edilmiştir.

Burada, FL istemci yeteneklerinin keşfedilmesini sağlamak için MQTT'den yararlanılmaktadır. Ayrıca, FL istemcisi olarak hareket etmek isteyen cihazların yeteneklerinin tanımlanması için OMA LwM2M semantiğinden yararlanılmaktadır.

Bu çalışmada hem MQTT hem de OMA LwM2M'den yararlanıyoruz, böylece MQTT'nin ölçeklenebilirliğine anlamsal açıdan zengin OMA LwM2M veri modelinin sunduğu birlikte çalışabilirliği ekliyoruz. Ayrıca, model yakınsamasına ulaşılana kadar model değişim adımları da dahil olmak üzere *tüm* FL *aşamaları boyunca FL* istemcileri ve *FL* toplayıcı arasında veri alışverişini desteklemek için önceki öneriyi önemli ölçüde genişletiyoruz.

3.1. Temel yapı taşları

Aşağıda, MQTT ve OMA LwM2M'nin temelleri kısaca bildirilmektedir.

3.1.1. MQTT

MQTT, IoT için yayınla/abone ol paradigmasını izleyen hafif mesajlaşma protokollerinden biridir [15]. MQTT'de bir istemci hem *yayıncı* hem de *abone* olarak hareket edebilir. İkincisi, bir "konuya" abone olur ve bir yayıncı tarafından bu konuda yeni bir mesaj oluşturulduğunda, MQTT sunucusu rolünü oynayan bir varlık, yani broker aracılığıyla bildirimler alır. MQTT konuları, bir dosya sistemindeki klasörlere ve dosyalara benzer şekilde bir hiyerarşi içinde yapılandırılır ve sınırlayıcı olarak ileri eğik çizgi (/) kullanılır. Her seviyenin geçerli olması için en az bir karakterden oluşması gerekir. Mesaj yayını her seferinde bir konu için yapılabilirken, protokol istemcinin birden fazla konuya aboneliğine izin verir. Özellikle, iki joker karakterden birini veya her ikisini içeren bir mesaj kullanarak birden fazla konuya abone olmak mümkündür: # (karma karakter) çok seviyeli joker karakter; + (artı karakter) tek seviyeli joker karakter. Değiştirilen mesaj yükü dize veya dosya içerebilir; standart tarafından belirli bir format tanımlanmamıştır ve tipik olarak uygulamaya özgüdür.

IoT alanında MQTT, İnternet alanında yaygın olarak kullanılan ve Hiper Metin Aktarım Protokolü (HTTP) üzerinden istemci-sunucu modelinde veri aktarımını sağlayan web odaklı Temsilci Durum Aktarımı (REST) mimarisine göre yaygın olarak tercih edilmektedir [21]. REST'in daha büyük başlıklarının yanı sıra istek-yanıt protokolleriyle ilişkili ek yük, pil kullanımını MQTT'den daha fazla etkilemektedir [22].

MQTT, istemcilerin aralıklı bağlantı ve veri kaybından muzdarip olabileceği bağlamlar için geliştirilmiştir. Yukarıda bahsedilen zorlukların üstesinden gelmek için protokole belirli özellikler dahil edilmiştir. İlk olarak, en az garantiliden (0) en çok garantiliye (2) kadar üç ayarlanabilir Hizmet Kalitesi (QoS) seçeneği mevcuttur. Aslında, istemci ve broker arasında farklı onay (ack) mesajları kullanırlar: QoS 0, ack olmadan bir ateşle ve unut modeli kullanır; QoS 1, mesajın en az bir kez alındığından emin olmak için bir PUBACK mesajı kullanır; QoS 3, yayınlanan bir mesajın tam olarak bir kez alındığından emin olmak için PUBREC, PUBREL ve PUBCOMP mesajlarını öngörür.

Bir istemci bir broker'a bağlandığında, kalıcı olmayan bir bağlantı (temiz oturum) veya kalıcı bir bağlantı kullanabilir. *Temiz oturum* seçeneği, istemci ve sunucu arasında oturum geri yüklendiğinde ayarları yönetir. Seçenek true olarak ayarlanırsa, broker istemci için herhangi bir abonelik bilgisi veya teslim edilmemiş mesaj saklamaz. *Yanlış* olarak ayarlanırsa, aracının konu aboneliklerini tutmasını ve istemci yeniden bağlanana kadar teslim edemediği mesajları saklamasını sağlar. *cleansession false* seçeneği, aralıklı olarak bağlanan istemcilerin yeniden bağlandıklarında abone olunan konularda iletilen mesajları okumalarına izin vererek bu durumla başa çıkar. Benzer şekilde, mesaj saklama bayrağı yayıncı tarafından ayarlanarak broker'a mesajı gelecekteki tüm abonelere göndermek üzere o konuda saklaması talimatı verilebilir. *Saklama bayrağı* kullanıldığında, her zaman ve sadece o konuda yayınlanan son mesaj saklanır.

3.2.2. OMA LwM2M

OMA LwM2M standardı, her bir cihazın anlamsal modellerle tanımlandığı, eXtensible Markup Language (XML) model dosyası kullanan, hafif bir istemci-sunucu protokolü tanımlar [23,24]. Temel olarak bir Nesne, cihazın belirli bir yazılım/donanım bileşenini (sensörler, antenler veya cihaz yazılımı gibi) ilişkili kaynaklarla (örneğin, değer, birim, maksimum değer, minimum değer) tanımlamak ve kontrol etmek için kullanılır. Cihazın karakteristiğine bağlı olarak, cihaz üzerinde aynı nesnenin bir veya daha fazla örneği olabilir, örneğin birden fazla sıcaklık sensörü gibi.

Standart, her kaynağın *objectID/instanceID/resourceID* şeklinde birleştirilen tanımlayıcılardan oluşan bir Tekdüzen Kaynak Tanımlayıcı (URI) yolu aracılığıyla benzersiz bir şekilde tanımlanmasını gerektirir. Örneğin, sıcaklık sensörü nesnesi 3303 tanımlayıcısıyla tanımlanırken, sıcaklık değeriyle ilgili kaynak 5700 ile tanımlanır. Ardından, standardı izleyerek, belirli bir cihazın ilk sıcaklık sensörünün kaynağı URI 3303/0/5700, ikinci sensörün kaynağı URI 3303/1/5700 aracılığıyla tanımlanır, böylece aynı nesnenin farklı *instanceID*'leri için iki farklı URI yolu oluşturulur.

İttifak, standart nesnelerin ve yeniden kaynakların halka açık bir kaydını sağlar. Her geliştirici yeni standart nesnelerin oluşturulmasına katkıda bulunabilir veya özelleştirilmiş nesneleri kendi ortamında kullanabilir.

OMA LwM2M, IoT cihazından/cihazına aktarılan verileri azaltmak için kompakt ve verimli bir veri modeli sağlamak üzere tasarlanmıştır. Daha spesifik olarak, bilgiler tanımlayıcılar kullanılarak etiketlenir ve kodlanır (örneğin, sıcaklık nesnesi için 3303). Bu tanımlayıcılar daha sonra çevrilir ve modellerde depolanan tüm bilgiler sayesinde bilgiler zenginleştirilir.

3.2. Temel varsayımlar ve tasarım seçimleri

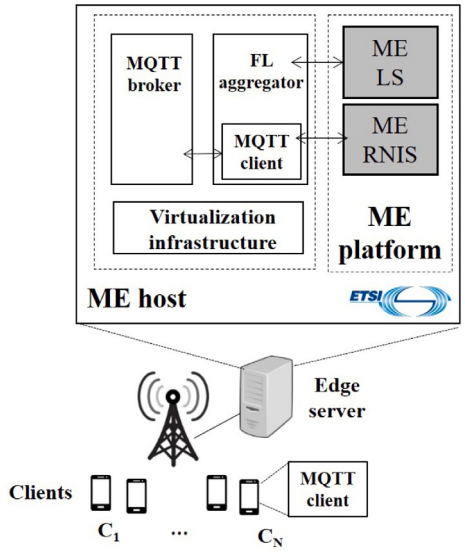
Şekil 1'de gösterildiği gibi, küresel model toplamanın Avrupa Telekomünikasyon Standartları Enstitüsü (ETSI) Çok Erişimli Uç Hesaplama (MEC) spesifikasyonlarına [25] uygun olarak konuşlandırılmış bir uç sunucuda, yani ME ana bilgisayarında gerçekleştirildiği uç tabanlı bir FL yaklaşımını ele alıyoruz. İkincisi, sanallaştırılmış bir uygulama (ME uygulaması) olarak FL prosedürlerini yöneten bir FL toplayıcı barındırır.

FL istemcisi olarak hareket etmeye istekli N IoT cihazlarından oluşan bir popülasyon bağlanır
FL toplayıcısına bir uç ağ alanı üzerinden bağlanır. Özellikle, uç sunucunun bağlı olduğu Baz İstasyonuna (BS)/Erişim Noktasına (AP) bağlı olduklarını varsayıyoruz. N cihazları arasında yalnızca bir $N^{\sim} \leq N$ cihazlarının alt kümesi FL istemcisi olarak hareket etmeye uygundur ve en fazla $M \leq N^{\sim}$ cihazlar FL toplayıcı tarafından model eğitim prosedürü için FL istemcileri olarak hareket etmek üzere seçilebilir.

Genelliği kaybetmeden, senkronize FL'nin uygulandığını varsayıyoruz, yani sunucu her turda seçilen tüm FL istemcilerinden alınan model güncellemelerinin senkronize bir şekilde toplanmasını gerçekleştirir. Hem N cihazları hem de FL toplayıcısı bir MQTT istemcisi çalıştırırken, MQTT aracısı da aynı uç sunucuya yerleştirilmiştir.
FL toplayıcı. Bu seçim, [14]'teki önceki çalışmamızda gösterildiği gibi, yayınlama ve abone olma mesajlarının iletilmesi için iletişim ayak izini azaltmayı amaçlamaktadır. Bununla birlikte, öneri uzak bir komisyoncunun varlığında da çalışabilir.

Genel olarak, önerilen FL-MQTT çerçevesi boyunca aşağıdaki ana tasarım seçenekleri dikkate alınmaktadır:

- Tüm MQTT konuları, FL istemci keşfi ve bildirimi aşaması da dahil olmak üzere FL modeli eğitiminin her adımında kendi kendini açıklayacak şekilde tasarlanmıştır; farklı aşamaları karakterize eden belirli önekler ve bileşenler kullanılır;
- değiş tokuş edilen tüm MQTT yayınlama mesajları JavaScript Object Notation (JSON) formatlı bir yük taşır ve iletilen veriler OMA LwM2M semantik standardını takip eder, böylece birlikte çalışabilirliği artırarak herhangi bir FL istemcisi (ve FL toplayıcısı) tarafından anlaşılabilir;



/ig. 1. ETSI MEC mimarisi içinde tasarlanan FL-MQTT çerçevesi.

- İstemci keşfi ve model eğitimi aşamalarını desteklemek için gerektiğinde, bazı yeni OMA LwM2M nesneleri ve ilgili kaynaklar, OMA yönergelerine göre [24] 10241-26240 aralığında özel şirketlere veya bireylere ayrılmış kimlikler arasından kimlikleri seçilerek tanımlanır.

Aşağıdaki Bölümlerde, genel FL eğitiminin ana adımları, tasarlanan iş akışları açıklanarak tartışılmaktadır.

4. L istemci keşfi

Belirli bir türdeki FL görevine katkıda bulunmak isteyen IoT cihazları (örneğin, insan etkinliği tanıma, ses tanıma, görüntü tanıma), *konu#1,disc olarak* kısaltılan ve Tablo 1*'de* tam olarak bildirilen MQTT konusunda FL istemcileri arayan bir FL toplayıcısından gelen güncellemelere abone olur. Burada, keşif aşamasında yararlanılan iki konunun tanımı, ilgili MQTT yayınlama mesajlarının yükleriyle birlikte özetlenmiştir. Tablo 1 ve aşağıdaki Tablolar, tasarlanan konuların semantik açıdan zengin ancak kompakt yapısını anlamaya ve FL istemcileri ile FL toplayıcısı arasında gerçekte neyin değiş tokuş edildiği ve birlikte çalışabilirliğin somut olarak nasıl sağlandığı hakkında fikir vermeye olanak tanır. Özellikle, *konu#1,disc'*te, *disc* öneki bu kanalın istemci keşif aşamasına adanmış olduğunu gösterir; *fl* bileşeni bir FL görevine atıfta bulunduğunu gösterir; ve konu ağacının son seviyesi başlatılacak FL görevinin türünü belirtir, örneğin ses tanıma.

Bir FL eğitim prosedürünün başlatılması gerektiğinde, FL aggregator buna uygun olarak, yükü *DiscoveryFL* (Object ID 18333 ile) adlı yeni tanımlanmış bir nesne taşıyan bir MQTT mesajı yayınlar.

Daha ayrıntılı olarak, FL toplayıcı tarafından *konu#1,disc* üzerinde yayınlanan *DiscoveryFL* nesnesinin kaynakları Tablo 2*'de* gösterilmektedir. Nesne şu bilgileri içerir: bu durumda FL toplayıcı kimliği olan ilgili FL varlığının tanımlayıcısı (FL Varlık Kimliği); Görev Türü Kimliği (örn. ses tanıma) ve FL toplayıcının FL istemci seçimi için bilmek istediği bilgilere (örn. pil, CPU ve bellek durumu) giden özel yolu taşıyan OMA URI yolu.

Bu tür bilgiler, Tablo 3'te gösterildiği gibi *ClientFL* (Object ID 18332) adlı yeni bir OMA LwM2M nesnesi kullanılarak beyan edilebilir.

Ardından, FL toplayıcı aday FL istemcilerinden gelen güncellemelere abone olur.

MQTT aracısı, *konu#1,disc* üzerindeki yayını (FL toplayıcıdan) FL görev bilgisi alımına abone olan cihazlara iletir. Bu tür cihazlar FL olarak aday olup olamayacaklarını değerlendirir

Tablo 1
İstemci keşif prosedürü sırasında yararlanılan konuların ve yüklerin açıklaması.

#	Konu adı	Yükü yayınla
1,disk	*disc/fl/tasktype*	{‘‘bn’’:‘‘/18833/0/’’, e’’:[{‘‘n’’:‘‘26241’’ "v":"AB123"} {"n":"26249", "v":"voicerecog"}, {"n":"26250", "v":"/18832"}]}
2,disk	*info/fl/tasktype/serverid/taskid*	{"bn":"/18832/0/", e":[{"n":"26241", "v":"EF789"}, {‘‘n’’:‘‘26242’’,‘‘v’’:‘‘50"}, {‘‘n’’:‘‘26243’’,‘‘v’’:‘‘500"}, {‘‘n’’:‘‘26244’’,‘‘v’’:‘‘240"}, {‘‘n’’:‘‘26245,‘‘v’’:‘‘5000"}, {‘‘n’’:‘‘26246’’,‘‘v’’:‘‘60000"}, {‘‘n’’:‘‘26247’’,‘‘v’’:‘‘150"}, {‘‘n’’:‘‘26248’’,‘‘v’’:‘‘6500"}]}

Tablo 2
OMA LwM2M DiscoveryFL nesnesinin kaynakları.

URI	Kaynak
/18833/0/26241	FL Tüzel Kişi Kimliği
/18833/0/26249	Görev Türü Kimliği
/18833/0/26250	OMA LwM2M URI Yolu

Tablo 3
OMA LwM2M ClientFL nesnesinin kaynakları.

URI	Kaynak	Ölçü birimi
/18832/0/26241	FL Tüzel Kişi Kimliği	-
/18832/0/26242	Pil Seviyesi	[%]
/18832/0/26243	Akü Kapasitesi	mAh
/18832/0/26244	CPU	MHz
/18832/0/26245	Ücretsiz Bellek	kB
/18832/0/26246	Veri Kümesi Boyutu	kB
/18832/0/26247	Veri Kümesi Girişleri	-
/18832/0/26248	Veri Kümesi Yaşı	s

FL toplayıcı tarafından *DiscoveryFL* nesnesinde yayınlanan bilgilere göre istemciler.

N cihazlar arasında, $\mathcal{N}$ düğümlerinin bir alt kümesi, kendi kendine uygunluk testini geçer

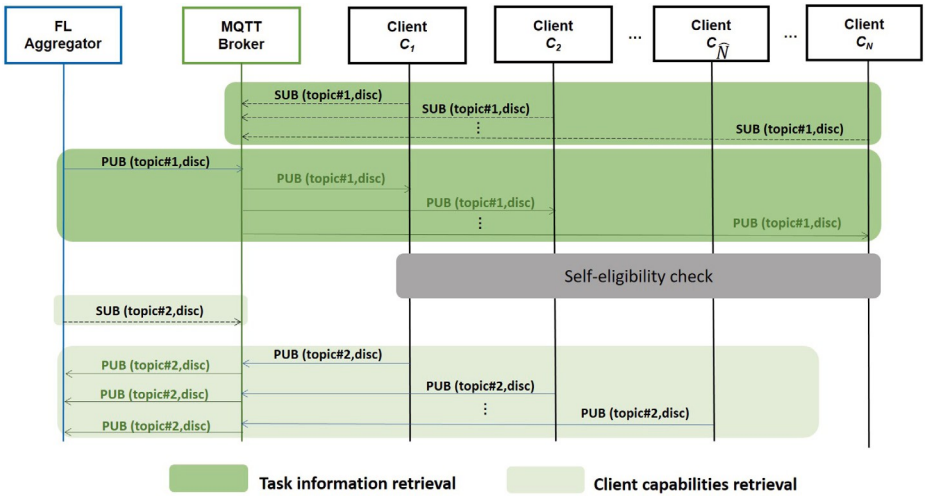
FL toplayıcının ilgilendiği bilgileri özel bir konu kullanarak kontrol edin ve yayınlayın: Tablo 1'de ayrıntıları verilen *konu#2,disc*. Böyle bir durumda: konudaki *info* ön eki mesajın bir bilgi içerdiğini gösterir; *fl/tasktype/* bileşenleri istenen göreve atıfta bulunur; *serverid/taskid*'in açık artırma sunucusunu ve özel görev tanımlayıcısını öğelendirmesi gerekir. Aynı toplayıcı üzerinde aynı görev türünün birden fazla örneği olabileceği için *taskid* bileşenine ihtiyaç vardır.

Bu MQTT yayınlama mesajı, *ClientFL* nesnesi kullanılarak OMA LwM2M semantiği aracılığıyla tanımlanan istemci tanımlayıcısını ve yeteneklerini taşıyacaktır, Tablo 3. Bu aşamada, genellik kaybı olmaksızın, cihazların pil, işlem ve bellek yeteneklerinin yanı sıra sahip olunan veri kümesinin temel özellikleri hakkındaki bilgilerin *ClientFL* nesnesine dahil edildiğini varsayıyoruz.

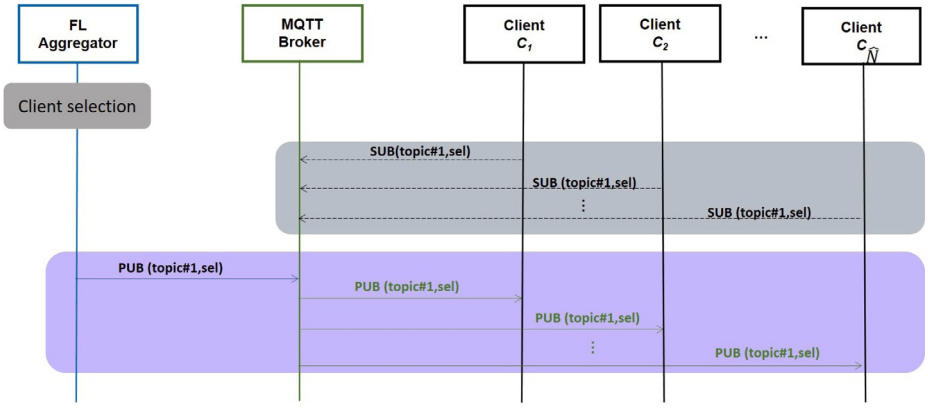
MQTT/OMA LwM2M destekli FL istemci keşif aşaması için genel iş akışı Şekil 2'de raporlanmıştır.

5. L istemci seçimi ve bildirimi

FL toplayıcısı, istemci seçimini gerçekleştirmek için alınan bilgileri kullanır. Özellikle bu amaçla FL toplayıcısı, alınan cihaza özgü yeteneklerin yanı sıra tasarımımızda ME platformu tarafından sağlanan ağla ilgili bazı bilgilerden de yararlanır. Aslında, ME platformunun *Radyo Ağı Bilgi Hizmeti (*RNIS) ve *Konum Hizmeti (*LS) bileşenleri yerel olarak



/ig. 2. FL müşteri keşif iş akışı.



/ig. 3. FL toplayıcı-FL istemcilerinin iş akışı, istemci bildirim prosedürü sırasında mesaj alışverişi.

sırasıyla telsizle ilgili ve artırılmış konumlandırma bilgileri sağlar. ME ana bilgisayarında konuşlandırılan FL toplayıcısı olarak, bu tür modüllerle etkileşim yoluyla yukarıda belirtilen bilgileri doğrudan alabilir. Bağlantı Kalitesi, FL istemcisi olarak seçilecek belirli bir IoT cihazı için model parametrelerinin/güncellemelerinin değiş tokuş edilme süresini tahmin etmek için kullanılabilir. Konum, FL toplayıcının, FL istemcisi olarak seçilen bir IoT cihazının genel eğitim görevi için kullanılabilir olup olmayacağını veya tamamlamadan önce bağlantısının kesilip kesilmeyeceğini çıkarmasına olanak tanır.

Bu makalenin kapsamı dışında kalan belirli FL müşteri seçim politikasına bakılmaksızın, seçilen M FL müşterilerinin eğitim prosedürüne katılımları konusunda bilgilendirilmeleri gerekmektedir.

Bu amaçla, FL toplayıcı seçilen istemcilerin listesini içeren bir mesaj yayınlar (bkz. Tablo 4) ve bu mesaj komisyoncunun tüm $\hat{N}$ kendilerini aday gösteren cihazlar *konu* hakkında yayın yaparak potansiyel müşteriler olarak keşif *aşaması#2,disc* ve ardından *#1,sel konusuna* abone olarak. İlgili iş akışı Şekil 3'te gösterilmektedir.

6. L eğitim prosedürü

FL istemcileri seçildikten ve bilgilendirildikten sonra, dağıtılmış eğitim başlayabilir ve birden fazla tur boyunca yinelenebilir. Şekil 4'te taslağı çizilen iş akışı, bu aşamada değiş tokuş edilen tüm mesajları ayrıntılı olarak göstermektedir.

Seçilen M FL istemcileri küresel aboneliği gerçekleştirir modeli, *konu#1* olarak kısaltılan belirli bir konunun kullanımı yoluyla, eğitim aşaması sırasında ilgili yayınlama mesajlarının yükleriyle birlikte özetlenir.

FL toplayıcısı da *topic#2,trai'*ye abone olarak her eğitim turundan sonra FL istemcilerinden model güncellemeleri almakla ilgilendiğini beyan eder. Buna ek olarak, yükü FL istemcileri tarafından eğitilecek orijinal modeli içeren bir mesaj yayınlar[1] veya modelin depolandığı ve FL istemcileri tarafından alınabileceği deponun Tekdüzen Kaynak Konum Belirleyicisi (URL).

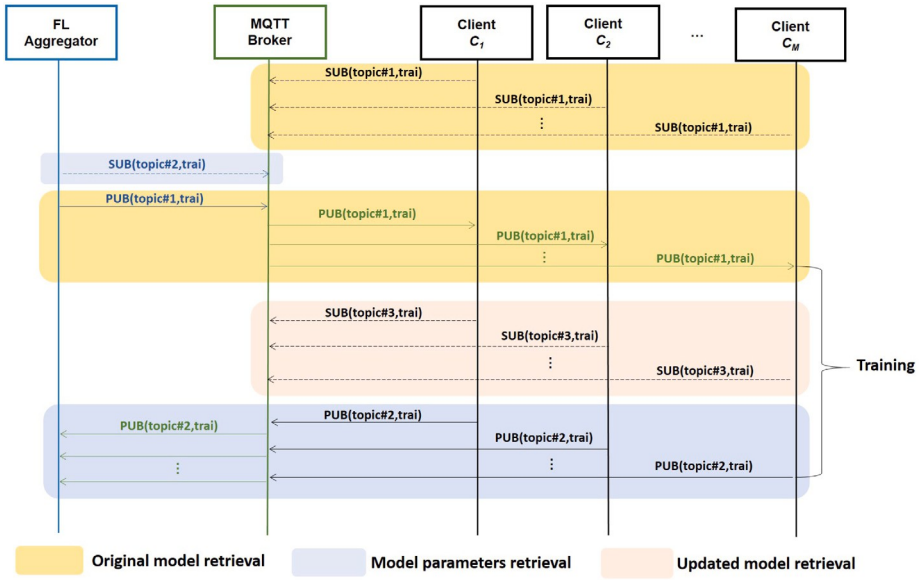
Sinir Ağı (NN) modeli (*NNModel*) olarak adlandırılan bir başka özel OMA LwM2M nesnesi, FL toplayıcı ve seçilen FL istemcileri arasında değiş tokuş edilecek model parametrelerini tanımlamak için yeni tanımlanmıştır (Nesne Kimliği 18334). Tablo 6'da bildirilen bu nesne şu bilgileri içerir: tur tanımlayıcısı (*Round ID*), istemci tanımlayıcısı (*FL Varlık Kimliği*), model (*Model Bilgisi)*, eğitim aşamasının gerçekleştiği zaman *eğitim* ve kullanılan tüm konuların tanımının yer aldığı Tablo 5'te tam olarak raporlanmıştır

başladı (T_{start}) ve eğitimin başlamasından bu yana geçen süre (T_{rel}), FL istemcilerinin senkronizasyonu sürdürmesi için yararlıdır.

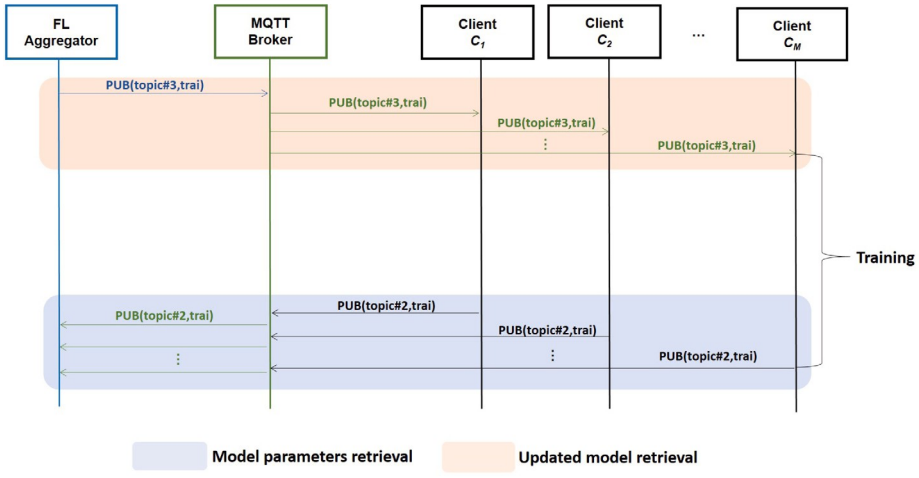
Özellikle, NN modelinin mimarisi ve ağırlıkları hakkındaki bilgileri kaydetmek için Hiyerarşik Veri Formatı'nın (HDF) [28] güncel sürümü olan HDF5 formatını kullandık [29].

Aynı OMA LwM2M nesnesi, FL toplayıcısı tarafından paylaşılan orijinal modeli ve FL istemcileri tarafından ağırlıkların sonraki güncellemelerini tanımlamak için kullanılır. Bunlar Tur Kimliği ile ayırt edilir

[1] MQTT yükünde 256 MB'a kadar bayt aktarılabilir [26], dolayısıyla son teknoloji modellerin yüzlerce MB boyutunda olduğu düşünüldüğünde orijinal modeli taşımak için yeterlidir [27].



(a) Training: first round with the exchange of the original model.



(b) Training: rounds after the exchange of the original model.

/ig. 4. FL eğitim prosedürünün iş akışı.

Tablo 4
İstemci bildirim prosedürü için yararlanılan konu ve yükün açıklaması. # Konu adı

	Yükü yayınla	
1,sel	*modl/fl/görevtipi/seçim*	[{"n":"clnts", "v":"*clientID*$_1$, . . . *clientIDM*"}]

kaynağı, orijinal model için sıfır ve model güncellemeleri için sıfırdan büyüktür.

FL toplayıcının FL istemcilerine tur tur gönderebileceği eğitim prosedürüyle ilgili diğer parametreler öğrenme oranı veya mini partinin boyutu olabilir [30]. Bu parametreler, yeni OMA LwM2M kaynakları tanımlanarak Tablo 6'da tanımlanan nesneye eklenebilir.

FL istemcileri, kendi veri kümeleri üzerinde yerel olarak eğitim gerçekleştirdikten sonra, model parametrelerini eğiterek *konu#2'de* yayınlar. Bu amaçla Tablo 6'dakiyle aynı nesne kullanılır. Bu durumda nesne, model güncellemesinin alındığı FL istemcisini izlemek için *FL Varlık Kimliği* kaynağını da taşır. Bu kaynak *konu#1,trai'*nin yayınlama mesajında gerekli değildir. Bilgi için Amerikan Standart Kodu *Model Bilgisi* kaynağı olarak yayınlanan Değişim (ASCII) dosyası b u n u n yerine yalnızca FL toplayıcısına gönderilecek NN'nin ağırlıklarına ilişkin bilgileri içerir.

Daha sonra FL istemcileri, sonraki turlarda FL toplayıcısından güncellenmiş modeli almak için *konu#3,trai'*ye abone olurlar.

Konu#1,trai sadece ilk turda FL toplayıcının eğitilecek modelin parametrelerini göndermesine izin vermek için kullanılır. Daha sonra, eğitim prosedürünün sonuna kadar, FL toplayıcı, eğitim altındaki modelin güncellenmiş parametrelerini değiştirmek *için konu#3,trai'yi* kullanacaktır.

7. Beklenen faydalar

Tasarlanan çerçeve aşağıdaki ana güçlü yönleri sergilemektedir. Bunlardan bazıları kaldıraçlı yapı taşlarının vanilya tasarımından miras kalırken, diğerleri tasarlanan bireysel özelleştirmelerden ve bunlar arasında öngörülen etkileşimden kaynaklanmaktadır.

Ölçeklenebilirlik. FL, merkezi öğrenme yaklaşımlarına kıyasla ö l ç e k l e n e b i l i r l i k [12,13] gibi çeşitli avantajlar sunmaktadır. Ölçeklenebilirlik, FL istemcileri olarak hareket eden ve eğitim sonuçlarını dağıtan daha fazla cihazı federe öğrenme sürecine dahil etme yeteneğini ifade eder. Bununla birlikte, bu özellik FL'de verili

olarak kabul edilemez ve

Tablo 5
Eğitim prosedürü sırasında yararlanılan konuların ve yüklerin açıklaması.

#	Konu adı	Yükleri yayınlayın
1, eğitim	*modl/fl/tasktype/serverid/taskid*	{{"bn":"/18834/0/",} e": [{‘‘n":‘‘26251",‘‘v":‘‘0"} {"n":"26252", "v":"fileAsciiFormat"} {"n":"26253", ‘‘v":‘‘9/03/22-12:02:23"} {"n":"26254", "v":"200"}]}
2, eğitim	*modl/fl/tasktype/serverid/taskid/trained*	{"bn":"/18834/0/",} e": [{"n":"26241", "v":"AB123"}, {‘‘n":‘‘26251",‘‘v":‘‘0"} {"n":"26252", "v":"fileAsciiFormat"} {"n":"26253", ‘‘v":‘‘9/03/22-12:02:23"} {"n":"26254", "v":"400"}]}
3, eğitim	*modl/fl/tasktype/serverid/taskid/update*	{ "bn":"/18834/0/", e": [{‘‘n":‘‘26251",‘‘v":‘‘1"} {"n":"26252", "v":"fileAsciiFormat"} {"n":"26253", ‘‘v":‘‘9/03/22-12:02:23"} {"n":"26254", "v":"500"}]}

Tablo 6
OMA LwM2M NNModel nesnesinin

kaynakları. URI	Kaynak
/18834/0/26251	Yuvarlak Kimlik
/18834/0/26241	FL Tüzel Kişi Kimliği
/18834/0/26252	Model Bilgisi
/18834/0/26253	Tstart
/18834/0/26254	Trel

hem genel eğitim performansı hem de iletişim yükü açısından farklı perspektifler altında analiz edilmelidir. FL istemcisi olarak hareket eden heterojen ve potansiyel olarak kaynak kısıtlı cihazların varlığında ve bu çalışmada hedeflendiği gibi senkronize FL işlemleri için ölçeklenebilirlik engellenebilir [31]. Bununla birlikte, önerimiz ölçeklenebilirliği sağlayabilecek mantıklı bir FL istemci seçimini zorlamaya izin vermektedir [12,31]. Ölçeklenebilirliği geliştirmeye yönelik diğer teknikler, özellikle yüzlerce veya binlerce cihaz düşünüldüğünde, ağ kaynaklarının optimizasyonu ile ilgilidir [12]. MQTT gibi özellikle kaynak kısıtlı ortamlar için tasarlanmış ve milyonlarca IoT cihazı içerebilen akıllı şehirler, güç dağıtım şebekeleri veya bağlı araç filolarını içeren büyük ölçekli IoT senaryolarında çalışacağı düşünülen bir iletişim protokolünden yararlanarak, yaklaşımımız ölçeklenebilirlik hedefini takip etmektedir. FL istemcileri ve FL toplayıcısının birbirinden ayrılması, bilgi alışverişinde daha fazla ölçeklenebilirlik sağlar. FL toplayıcısının ilgili FL istemcileriyle olan bağlantıları yönetmesine gerek yoktur çünkü bu iş MQTT aracısına devredilmiştir ve model toplama prosedürlerine odaklanabilir. Bu seçim, FL istemcilerinin sayısı arttıkça özellikle uygun görünmektedir. Bununla birlikte, yükün aracıya aktarılması, çok sayıda bağlı cihaz ve mesaj iş yükü ile başa çıkmak için uygun kaynaklarla donatılması amacıyla düzgün bir şekilde konuşlandırılması ihtiyacını beraberinde getirebilir. MQTT broker örneklerinin uygun dikey/yatay ölçeklendirme prosedürleri, uç platformu yöneten orkestratör tarafından uygulanabilir, ancak bu, bu çalışmanın kapsamı dışındaki bir konudur.

/esneklik. Gerekli parametrelerin yukarıda belirtilen mesajlarda belirtilmesi koşuluyla, herhangi bir istemci seçim politikası tasarlanan çerçevede uygulanabilir. FL istemcilerini tanımlamak için farklı OMA LwM2M kaynak türleri ve sayılarından yararlanılabilir

yeteneklerini geliştirirken, OMA LwM2M *ClientFL* nesnesinin modelini kolayca genişletir.

Hata eğilimli ve kesintili bağlantıya karşı sağlamlık. MQTT, aktarım katmanında İletim Kontrol Protokolünü (TCP) kullanarak güvenilir iletişim sağlar. MQTT'deki mesajların güvenilirliği, daha önce bahsedilen QoS seviyeleri, *temizleme* ve bayrakları *tutma* yoluyla daha da artırılabilir. Bu özellikler, mesajların tamponlanmasını (cleansession *false, retain*) ve konu aboneliği (cleansession *false*) gibi istemci ayarlarının korunmasını sağlayarak hataya açık kablosuz bağlantılar üzerinden bağlanan IoT FL istemcileriyle başa çıkmak için son derece uygun görünmektedir. Bu tür mekanizmalar, FL istemcilerinin zorlu yayılma koşulları veya IoT bağlamlarında yaygın olan uyku modu işlemleri nedeniyle sık sık bağlantısının kesilebileceği senaryolarda çok önemlidir.

Düşük iletişim ayak izi. HTTP gibi diğer güvenilir protokollerle karşılaştırıldığında, MQTT daha hafif başlığı sayesinde daha düşük bir iletişim ayak izine sahiptir [32]. Belirli özellikleri onu IoT uygulamaları için değerli bir mesajlaşma protokolü haline getirmektedir [33]. Ayrıca konu yapısı hafif olacak şekilde tasarlanmıştır: OMA LwM2M, potansiyel olarak kısıtlı FL istemcilerinden/istemcilerine aktarılan verileri azaltmak için kompakt ve verimli veri modeli sağlar. Ek olarak, FL toplayıcının bir ME uygulaması olarak konuşlandırılması, FL istemcisi olarak hareket etmek isteyen cihazın ve Radyo Erişim Ağı (RAN) tesislerinde halihazırda mevcut olan ağla ilgili bilgilerin alınması için bağlı olduğu radyo arayüzünün yükünü önler.

Düşük protokol ek yükü. Bilgi alışverişi için gerekli işin bir kısmını aracıya yükleyen MQTT'den yararlanarak, teklif kısıtlı kaynaklara sahip cihazlar için uygundur. Her FL istemcisi, almak istediği bilgiler ne olursa olsun, yalnızca bir bağlantıyı, yani aracı ile olan bağlantıyı aktif tutma yüküne sahiptir.

Heterojen cihazlar için destek. Tasarlanan çerçeve, işlem ve pil ömrü açısından heterojen yetenekler sergileyen ve farklı bağlantı arayüzleri üzerinden farklı ağ koşullarını deneyimleyen potansiyel istemcilerin tanımlanmasını doğal olarak destekleyebilmektedir.

Birlikte çalışabilirlik. Bu zorlu hedefe, FL prosedürlerinin farklı aşamalarını ayırt edebilen ve farklı FL görevlerini (örneğin, nesne algılama, konuşma tanıma, sınıflandırma) tanımlayabilen örnek MQTT konuları ve heterojen istemcilerin yeteneklerini *tek* tip *semantik* açıdan *zengin* bir şekilde tanımlamak için yayınlama mesajlarının yükünde gerçekleştirilen OMA LwM2M nesneleri aracılığıyla ulaşılmaktadır. FL istemcilerinin FL toplayıcısından ayrılması, öğrenme çıktılarında üçüncü ilgili taraflarla birlikte çalışabilirliğin de önünü açmaktadır. Herhangi bir yeni aktör, uygulanan güvenlik mekanizmalarına uygun olarak, FL eylemleriyle ilgilenen tarafların mevcut iletişim/mekanizmada herhangi bir değişiklik yapmasına gerek kalmadan bilgi alışverişini devralabilir. Örneğin, halihazırda (kısmen) eğitilmiş bir modeli elde etmek isteyen yeni bir FL toplayıcısı, FL toplayıcısının eğitime katılan FL istemcilerine gönderdiği güncelleme konusuna (yani, *konu#3,trai*) abone olarak bu modeli elde edebilir. Bu şekilde, mesaj *tutma* bayrağıyla yayınlanmışsa, güncelleme konusunda yayınlanan son mesajı alacaktır. Az önce açıklanan işlevsellik türü, Federated Transfer Learning (FTL) [2,3] kavramını uygulamak için kullanılabilir. FL'nin bir evrimini temsil eden bu paradigma, diğer etki alanları hakkında oluşturulan bilgiden yararlanarak bir hedef etki alanı için etkili modeller oluşturmayı içerir. Veri kümeleriyle ilgili özelliklerin uzayında bir örtüşme ve/veya FL istemcileri arasında bir benzerlik b u l u n a n iki A ve B etki alanı göz önüne alındığında, A üzerinde öğrenilen bir model, bu küçük örtüşmelerden yararlanılarak B'ye aktarılır. Tanımlanan MQTT tabanlı prosedürler sayesinde, B alanına ait ikinci bir FL toplayıcısı, A alanına ait bir FL toplayıcısı tarafından eğitilen model hakkında kolayca bilgi edinebilir.

ML model-agnostisizm. Yerel veri kümelerini kullanarak yerel öğrenme modellerini hesaplamak için son cihazlarda çeşitli şemalar kullanılabilir. Onlar

örneğin ileri beslemeli sinir ağları (FNN), konvolüsyonel sinir ağları (CNN), destek vektör makineleri (SVM) ve uzun-kısa süreli bellek (LSTM) olabilir. Kullanılan yerel öğrenme modelinin türü kesinlikle dikkate alınan IoT uygulamasına bağlıdır [12]. Teklif, eğitilecek belirli ML modelinden bağımsızdır. Bu nedenle, çeşitli modeller ve ilgili çerçeveler çözümümüz tarafından desteklenebilir.

8. Performans değerlendirmesi

8.1. PoC açıklaması

Önerimizin fizibilitesini, verimliliğini ve etkinliğini değerlendirmek için aşağıdaki gibi detaylandırılmış gerçekçi bir küçük ölçekli PoC oluşturduk.

8.1.1. FL-MQTT uygulaması

FL istemcilerinden oluşan hibrit bir popülasyon dikkate alınmıştır. Bunlardan bazıları Docker konteynerleri [34] aracılığıyla konuşlandırılırken, bir tanesi *RaspberryPi4* üzerinde uygulanmıştır. Ayrıca bu ikinci durumda, FL istemcisini uygulayan uygulama konteynerleştirilmiştir. Başka bir konteyner ise FL toplayıcısını barındırmaktadır.

FL istemcilerini ve FL toplayıcısını barındıran Docker konteynerleri Docker köprü ağı kullanılarak birbirine bağlanmıştır. Bir *MikroTik* anahtarı, FL istemcilerinden birini barındıran *RaspberryPi4*'ü sanallaştırılmış FL toplayıcıyı barındıran makineye bağlar.

Hem FL istemcileri hem de FL toplayıcısı python için *tensorflow* [35] kütüphanelerini içerir.

Hem FL toplayıcı hem de FL istemcilerinde MQTT istemci uygulaması için *Mosquitto* yazılımından yararlanıyoruz [26]. *Mosquitto* aracılığıyla da dağıtılan MQTT broker, FL toplayıcı ile birlikte konumlandırılmıştır. Mosquitto, kurulumdaki basitliği, işletim sistemi taşınabilirliği ve ayrıca birkaç satır kodla yapılması nedeniyle en popüler MQTT uygulamalarından biri olarak kabul edilir [36]. Bu nedenle, Mosquitto'nun seçimi, [37]'deki çalışmada da doğrulandığı gibi, ölçeklenebilirliği daha da sağlamak içindir.

8.1.2. Veri Seti

İki farklı veri seti kullandık: Kanada İleri Düzey Enstitüsü Research, 10 classes (CIFAR-10) [38] ve modifiye edilmiş National In- stitute of Standards and Technology veritabanı (MNIST) el yazısı rakamları [39]. Her ikisi de FL araştırma alanında yaygın olarak kullanılmaktadır [4,40, 41]. Ayrıca, veri kümelerinin toplam boyutu (CIFAR-10 için 163 MB ve MNIST için 57 MB) ve bireysel örneklerin boyutu (CIFAR-10 için 2,5 kB ve MNIST için 0,6 kB) açısından tamamlayıcı özellikler sergiledikleri için bunlara atıfta bulunuyoruz.

İlk veri kümesi 32×32 piksel boyutunda 60.000 renkli görüntüden oluşmaktadır. Görüntüler, birbirini dışlayan mevcut 10 sınıftan biriyle etiketlenmiştir. Bu on sınıf şunlardır: uçak, araba, kuş, kedi, geyik, köpek, kurbağa, at, gemi ve kamyon.

Bunun yerine ikinci veri kümesi, sıfırdan dokuza kadar elle çizilmiş şekillerden oluşan 70.000 gri tonlamalı görüntü içermektedir. Her görüntü 28×28 piksel boyutundadır.

Bunlardan başlayarak, çeşitli istemcilere yerel veri kümeleri atanmıştır. Özellikle, tüm FL istemcileri aynı boyutta bir veri kümesine ve eşit olarak dağıtılmış sınıflara sahiptir. Başka bir deyişle, her sınıf için tüm FL istemcileri aynı sayıda örneğe sahiptir. Bu nedenle verilerin IID olduğu bir senaryoya düşüyoruz.

8.1.3. Eğitim modeli

Her iki veri kümesi için de görüntülerin tanınmasını gerçekleştirmek üzere CNN'ler kullanılmıştır. Özellikle, CIFAR-10 için [42]'de önerilen CNN'i kullandık,

Tablo 7
Eşitlik (1) ve (2)'deki parametrelerin değerleri.

Parametre	CIFAR-10 [kB]	MNIST [kB]
a	0.096	0.096
b	0.174	0.174
c	0.456	0.396
d	0.101	0.101
e	0.201	0.201

8.1.4. Benchmark

Öneri, yerel fonksiyonlara benzer parametreler geçirerek uzak metodları çağırmak için yüksek performanslı bir çerçeve olan Google Remote Procedure Call (gRPC) [44] ile karşılaştırılmıştır. Bir işlemi yürütmek için uzak varlıklar tarafından çağrılabilecek arayüzleri uygulayacak bir hizmet tanımlamaya izin verir. FL bağlamında, örneğin [45-47]'de, F L toplayıcı ve tüm FL istemcileri arasındaki iletişim kanalı olarak yaygın bir şekilde kullanılmaktadır. Böyle bir çözümün temsili bir uygulaması olarak açık kaynaklı bir FL çerçevesi olan Flower'dan [45] yararlanıyoruz.

8.2. İletişim ayak izi sonuçları

İlk sonuç kümesi, tasarlanan çerçevenin ağ üzerindeki etkisini değiş tokuş edilen veri miktarı açısından değerlendirmeyi amaçlamaktadır. Bu amaçla, Wireshark aracılığıyla FL toplayıcısı ile FL istemcileri arasında aracı aracılığıyla değiş tokuş edilen ve FL prosedürünün her aşaması sırasında MQTT mesajlarında taşınan toplam veri miktarını (bayt cinsinden) ölçüyoruz.

8.2.1. Müşteri keşfi ve bildirimi

İstemci keşif aşaması, potansiyel FL istemcilerinin yetenekleri hakkında bilgi edinilmesini sağlamak ve akıllı bir seçim prosedürü yürütmek için çok önemlidir. Önerimizde tasarlanan iş akışı, dağıtılmış eğitim başlamadan önce hem değiş tokuş edilen mesajlar hem de gecikme süresi açısından ihmal edilebilir bir ek yüke neden olur.

Keşif aşamasında değiş tokuş edilen veri miktarı N, $\mathcal{N}$ parametrelerinin bir fonksiyonu olarak aşağıdaki gibi hesaplanabilir:

$$ExchangedDatadisc \quad (N, \mathcal{N}) = (a+b) \cdot N + c \cdot \mathcal{N}, \tag{1}$$

Burada a, b, c parametreleri iki farklı veri seti için Tablo 7'de tanımlanmıştır.

Bunlar sırasıyla üretilen trafiği (kB cinsinden) ifade eder: *topic*#1, *disc*'ye abone olmak için her aday FL istemcisi tarafından; FL toplayıcı tarafından (*topic*#1, *disc* konusunda) *DiscoveryFL* nesnesini yayınladığında (MQTT aracısı tarafından her aday FL istemcisine gönderilir); ve *ClientFL* nesnesini yayınlamak için kendi uygunluk kontrolünü geçen bir FL istemcisi tarafından (*topic*#2, *disc* konusunda). FL toplayıcısının *topic*#2, *disc* üzerindeki aboneliğinin, FL toplayıcısı MQTT aracısı ile aynı yerde bulunduğundan, katlanılan iletişim ek yükü olarak kabul edilmediğini hatırlatırız.

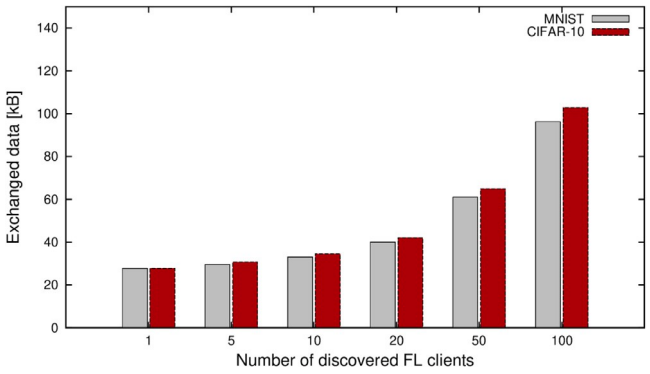
Rastgele strateji yerine akıllı bir istemci seçimi uygulamak için gereken potansiyel FL istemcilerinin yetenekleri hakkında bilgi iletmek için yayınlama mesajında (*topic*#2, *disc* konusunda) birkaç ek bayt taşınması gerekir. Gerçekten de, rastgele seçim uygulandığında bile, potansiyel FL istemcilerinin varlığı ve kimliğinin potansiyel FL istemcilerinin keşfedilmesi gerekmektedir. Yeteneklerini bildirmeden, her aday istemci CIFAR-10 ve MNIST için sırasıyla 456 bayt ve 396 bayta (c parametresi) karşı 221 bayt uzunluğunda bir mesaj yayınlayacaktır.

Bu durumda, FL istemcilerinin tüm popülasyonunu göz önünde bulunduruyoruz ve seçilen cihazların sayısı değiştikçe bildirim aşamasında değiş tokuş edilen trafiği ölçüyoruz.[2] Sonuçlar şu şekilde genelleştirilebilir

bunun yerine MNIST için [43]'te önerilen CNN'i kullandık.

FL toplayıcı, FL istemcilerinden aldığı yerel olarak eğitilmiş modelleri *FedAvg* politikasına göre birleştirir [4].

[2] Daha iyi işlem kabiliyeti sergileyen FL istemcilerini seçen basit bir politika varsayılmıştır.



/ig. 5. Keşfedilen FL istemci sayısı ($\tilde{N}$) değişirken keşif ve bildirim aşamalarında değiş tokuş edilen veriler, $N = 100$.

N, $\tilde{N}$ ve M'nin bir fonksiyonudur ve sırasıyla şunlara eşittir:

$$ExchangedDatasel(\tilde{N}) = d \cdot \tilde{N} + e \cdot \tilde{N}. \quad (2)$$

Sırasıyla d ve e parametreleri, her bir FL istemcisinin *#1,sel konusuna* abone olmak ve FL toplayıcısından aynı konuyla ilgili broker aracılığıyla bildirim almak için oluşturduğu trafiği temsil etmektedir. Değerleri Tablo 7'de raporlanmıştır. Bu metrik, dikkate alınan veri kümesinden bağımsızdır.

Şekil 5, parametreleri deneysel sonuçlardan türetilen Eşitlik (1) ve (2)'ye göre istemci keşfi ve bildirim aşamaları sırasında değiş tokuş edilen toplam veri miktarını bildirmektedir. $N =$ için $\tilde{N} =$100 istemci keşfi sırasında değiş tokuş edilen veri miktarı ve bildirim aşamaları, dikkate alınan ayarlar altında en kötü durumda 102,8 kB'ye eşittir (yani, CIFAR-10 veri seti). Bu, prosedürün neden olduğu küçük ek yükü doğrulamaktadır.

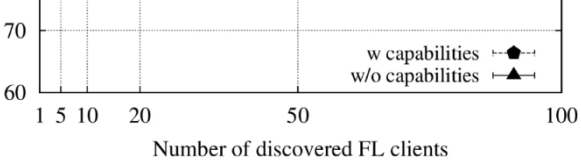
İstemci yeteneklerinin keşfini desteklemek için değiş tokuş edilen sınırlı miktarda ek bayt göz önüne alındığında, ihmal edilebilir bir ek zaman ek yükü yaşanır.

Genel olarak, istemci keşfi için harcanan zaman, keşfedilecek FL istemcilerinin sayısına, potansiyel FL istemcileri ile FL toplayıcı arasındaki bağlantı koşullarına ve MQTT aracısının yayınlama/abonelik mesajlarını iletme kapasitesine bağlıdır.

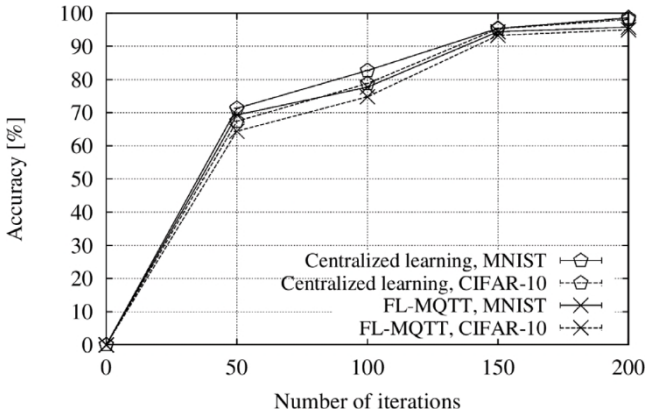
Şekil 6, FL istemcisi olarak hareket etmek isteyen cihazların aboneliğinden seçilen FL istemcilerinin bildirimine kadar geçen istemci keşif süresini bildirmektedir. Sınırlı sayıda istemciye sahip bir PoC'ye atıfta bulunarak ya da analitik olarak elde edilen önceki sonuçların aksine, bu durumda, istemci sayısıyla ölçeklendirmek için Mininet ağ emülatörü [48] ile bir ölçüm kampanyası yürütüyoruz. Elde edilen sonuçlar, Mininet ana bilgisayarları olarak konuşlandırılan tüm potansiyel FL istemcilerinin, özel bağlantılar aracılığıyla aynı ağ düğümüne bağlandığı ve ağ düğümünün, MQTT aracısının ve FL toplayıcısının birlikte konumlandırıldığı ME ana bilgisayarı olarak hareket eden bir Mininet ana bilgisayarına bağlandığı durumu ifade eder. 100 Mbps ve 10 ms sırasıyla trafik kontrolü aracılığıyla ayarlanan kapasite ve gecikme süresidir.
(*tc*) mesajların iletildiği bağlantılar üzerindeki yardımcı program FL istemcilerine MQTT aracısı. Bu tür ayarlar, sıkışık bir uç ortamına benzeyecek şekilde seçilmiştir.

En kötü durumda bile (100 keşfedilmiş istemci), değiş tokuş edilen mesajların sınırlı boyutu göz önüne alındığında, gecikme 100 ms'nin altındadır. Ayrıca, potansiyel müşterilerin yeteneklerinin değiş tokuşunu gerektiren öngörülen akıllı müşteri seçim prosedürü (*w yetenekleri* olarak etiketlenen eğri), rastgele müşteri seçimine kıyasla (yetenekler *olmadan olarak etiketlenen eğri*) keşif süresinde ihmal edilebilir bir artışa neden olur.

Şaşırtıcı olmayan bir şekilde sonuçlar, istemci keşif süresinin küçük olduğunu ve zamanın büyük bir kısmı yerel model eğitimi ve küresel model birleştirme için harcandığından toplam FL eğitim gecikmesine çok sınırlı bir katkı sağladığını doğrulamaktadır.



/ig. 6. Keşfedilen FL istemci sayısı değiştiğinde istemci keşif süresi (^{N}), N = 100, CIFAR-10 veri seti.



/ig. 7. Merkezi ve Federe öğrenme: yineleme sayısı değiştiğinde doğruluk.

8.2.2. Model eğitimi

İlk olarak FL yaklaşımını, bir dizi istemcinin veri kümelerini eğitim prosedürlerinden sorumlu bir sunucu ile paylaştığı geleneksel merkezi öğrenme ile karşılaştırıyoruz. Özellikle, sonuçlar hem MNIST hem de CIFAR-10 veri kümeleri için doğruluk açısından raporlanmış (bkz. Şekil 7) ve istemci sayısı 5 olarak sabitlendiğinde ve yineleme sayısı değiştirildiğinde FL yaklaşımıyla elde edilen doğrulukla karşılaştırılmıştır ions.[3]

Öncelikle, merkezi öğrenme ve FL yaklaşımı arasında kayda değer bir fark olmadığını gözlemleyebiliriz. Bu da FL'nin etkinliğini doğrulamaktadır. Özellikle, FL global modeli tarafından 200 tur [4] sonrasında elde edilen maksimum eğitim doğruluğu CIFAR-10 için %95 ve MNIST için %95,8'dir. Her iki veri kümesi için de 0,001'lik bir eğitim kaybına ulaşılmıştır. Daha sonra, FL istemcileri keşfedilip seçildikten sonra eğitim aşamasında önerimiz tarafından değiştirilen veri miktarını karşılaştırıyoruz. gRPC protokolüne karşı bildirilmiştir.[4]

Adil bir karşılaştırma elde etmek için, eğitim prosedürü her iki sistem için de doğruluk ve kayıp açısından aynı sonuçları verecek şekilde ayarlanmıştır.

[3] Lütfen yinelemelerin FL'de toplayıcıda gerçekleştirilen global turlara ve merkezi öğrenmede öğrenicide eğitim epoklarına karşılık geldiğine dikkat edin.

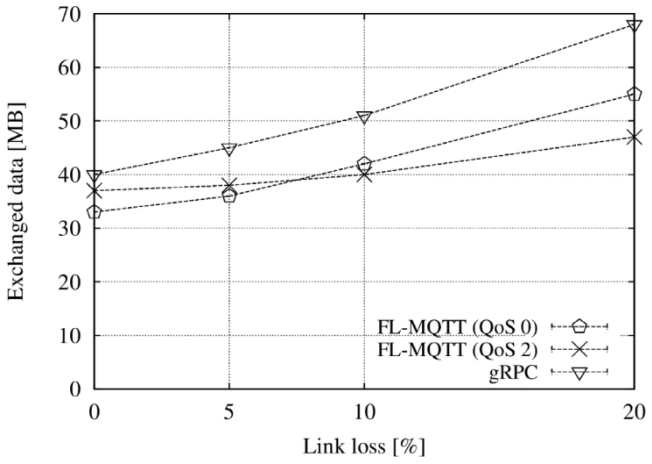
[4] Karşılaştırma önceki durumlar için öngörülmemiştir, çünkü Çiçek uygulaması FL keşif ve bildirim aşamaları için yerel olarak ilkel araçlar sağlamaz.

Tablo 8
Eğitim ayarları.

Parametre	Değer
Öğrenme oranı (η)	0.25
Epoch	5
Mini parti	50

Tablo 9
CPU ve RAM kullanımı.

	gRPC	FL-MQTT
CPU [%]	19.46	14.87
RAM [%]	21.84	22.60



/ig. 8. FL-MQTT Vs. gRPC: bağlantı kaybı değişirken eğitim prosedürü sırasında veri alışverişi (veri seti CIFAR-10).

çözümleri, Tablo 8'de bildirilen ayarlar altında [4]. Ölçümler her iki veri kümesi için örnek sayısının %50'si ile gerçekleştirilmiştir.[5] Şekil 8, FL toplayıcı ve 5 FL istemcisini birbirine bağlayan bağlantılar üzerindeki kayıp olasılığını değiştirirken, eğitim prosedürü sırasında değiş tokuş edilen veri miktarını göstermektedir. Yukarıda bahsedilen bağlantılar üzerindeki paket kaybı olasılığını gerçekçi ağ koşullarına benzeyecek şekilde ayarlamak için *tc* yardımcı programından yararlanılmıştır. FL'nin performansı MQTT hem QoS seviyesi 0 olarak ayarlandığında hem de 2 olarak ayarlandığında raporlanır.

Sonuçlar, grafiğin dağınıklığını önlemek için yalnızca CIFAR-10 veri kümesi için raporlanmıştır. Ancak, veri kümesi değiştirildiğinde önemli bir fark görülmemektedir. İki veri kümesi için orijinal modellerin boyutu karşılaştırılabilir olduğundan bu beklenen bir durumdur: *(i)* MNIST veri kümesinde kullanılan model için 137 kB ve *(ii)* CIFAR-10 veri kümesinde kullanılan model için 175 kB. Daha ayrıntılı *olarak, konu#1,trai* üzerine gönderilen NN yapısını taşıyan yayın mesajlarının boyutu: *(i)* MNIST için 143 kB ve *(ii)* CIFAR-10 için 182 kB. Güncellenmiş ağırlık değerlerini içeren ve *topic#2,trai* üzerinden gönderilen yayınlama mesajının boyutu: *(i)* MNIST için 102 kB ve *(ii)* CIFAR-10 için 122 kB'dir.

QoS seviyesi ne olursa olsun, öneri, her iki veri kümesi için de dikkate alınan kıyaslamaya kıyasla değiş tokuş edilen veri miktarını azaltmaya izin verir. Bunun temel nedeni, MQTT'nin HTTP protokolüne dayanan gRPC tabanlı çözüme kıyasla daha küçük başlıklar kullanmasıdır [49]. Şaşırtıcı olmayan bir şekilde, kayıpların olmadığı durumlarda en verimli çözüm QoS seviyesi 0 olan FL-MQTT, daha düşük ek yüke neden olduğu için.

İlginç bir şekilde, bağlantı kaybı arttıkça, veri alışverişi için tasarruf edilen ağ kaynakları açısından FL-MQTT'nin gRPC çözümüne göre iyileştirmeleri her iki veri kümesi için de daha önemli hale gelmektedir. Bu durum FL-MQTT için QoS seviyesi 2 etkinleştirildiğinde de geçerlidir. İkinci durumda, %20'ye eşit bağlantı kaybı için, en düşük miktarda veri alışverişi yapmak mümkündür. Bu eğilim, uygulama katmanında zorunlu kılınan el sıkışma sayesinde, uzun verilerin yeniden iletilmesine gerek olmadığı gerçeğine atfedilmelidir. kayıp durumunda mesajları yayınlar. Yeniden iletimler bunun yerine FL-MQTT (QoS 0) ve gRPC tarafından aktarım katmanında zorlanır.

Ayrıca, bütünlük adına, merkezi yaklaşıma kıyasla FL'nin faydalarını anlamak için, değiş tokuş edilen veriler açısından farklılıklar da dikkate alınmalıdır. Kayıpların olmadığı durumlarda, CIFAR-10 veri kümesi üzerinden gerçekleştirilen eğitim için FL-MQTT ile 40 MB'tan daha az veri alışverişi yapılmaktadır. Bu değer, merkezi öğrenme durumunda değiş tokuş edilmesi gereken 163 MB düzeyindeki toplam veri kümesi boyutunun oldukça altındadır.

8.3. İşlemci ve RAM ayak izi

Karşılaştırılan iki mesajlaşma şeması arasındaki farkları daha iyi anlamak için, sonuçlar veri alışverişi sırasında ortalama CPU ve bellek kullanımı açısından da raporlanmıştır. Sonuçlar yalnızca FL istemci tarafıyla ilgilidir, çünkü daha fazla kaynak kısıtlaması olabilir. Ölçümler, Intel Core i7-9750H işlemci, 16 GB DDR4 RAM ve NVIDIA GeForce GTX1050 Ti grafik kartı ile donatılmış fiziksel bir cihazda barındırılan sanal bir makine üzerinden yapılmıştır. Bunun yerine, fiziksel cihazın 4 GB RAM'i ve bir işlemci iş parçacığı sanal makineye ayrılmıştır.

Tablo 9, FL-MQTT yaklaşımının gRPC'ye kıyasla CPU kullanımı üzerinde önemli ölçüde daha düşük bir etki sergilediğini, ancak biraz daha yüksek bir RAM tüketimine neden olduğunu göstermektedir.

Elde edilen eğilimler, MQTT protokolünün kısıtlı cihazlar için gRPC'ye kıyasla daha uygun olduğunu doğrulamaktadır.

8.4. Birlikte çalışabilirlik testi

Bölüm 7'de tartışıldığı üzere, öneri farklı FL alanları arasında öğrenme transferini kolaylaştırabilir. Son sonuç kümesi, FTL'yi etkinleştirirken eğitim süresi açısından faydaları ölçmeyi amaçlamaktadır.

Daha önce açıklandığı gibi, $A1$ ⊠⊠⊠⊠⊠ gösterilen bir FL toplayıcısına ve bir dizi FL istemcisine sahip birincil bir etki alanı, $d1$ ve $A2$ olarak gösterilen başka bir FL toplayıcısına sahip ikincil bir hedef etki alanı, $d2$'yi ele alıyoruz. Bu

İkincisi, birincil senaryoya kıyasla benzer ancak farklı yerel veri kümelerine sahip bir dizi FL istemcisi ile çalışır. Özellikle, önceki ölçüm kampanyasında kullanılmayan örneklerin %50'sini kullandık.[6] $d1$⊠deki eğitim prosedürü sırasında, $A2$ aşağıdakilerden yararlanmaya çalışır

birincil etki alanının eğitimli modelini oluşturur ve bunu istemci kümesiyle paylaşır

(kısmen eğitilmiş) orijinal küresel model olarak.

Tasarlanan yayınla/abone ol çerçevesi ve kendi kendini açıklayan konular, $A2$⊠⊠⊠⊠ NN modelinin yapısı ve $d1$⊠deki global ağırlıklar hakkında turdan tura bilgi edinmek için aracıdaki *konu#1,trai* ve *konu#2,trai'*ye abone olmasını sağlar.

Şekil 9'da $A2$'nin küresel modeli eğitmek için tasarruf ettiği süre gösterilmektedir.

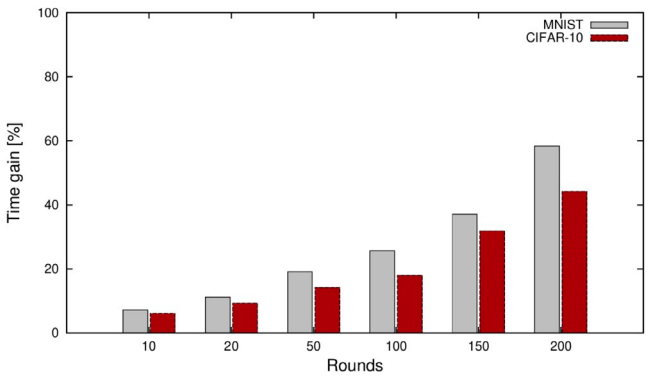
etki alanı. Sonuçlar, $A2$⊠⊠⊠⊠ modeli $d1$⊠den aldığı süre (tur olarak) değiştiğinde, orijinal eğitilmemiş küresel modelle başladığı duruma kıyasla rapor edilmiştir.

Tur sayısı ne kadar yüksek olursa, $d1$'deki model o kadar iyi eğitilir, dolayısıyla dikkate alınan her iki veri kümesi için de $A2$ için faydalar o kadar büyük olur. Aslında, $A2$, modeli kendi istemci kümesiyle eğitmek için daha fazla zaman kazandırır, çünkü ikincisi,

kısmen eğitilmiş bir modelle başlayarak, yerel eğitimi daha hızlı gerçekleştirir.

[5] Kalan %50'lik kısım bir sonraki alt bölümde tanımlanan birlikte çalışabilirlik ölçümlerini gerçekleştirmek için kullanılır.

[6] Bunun amacı dengeli veri setleri ve karşılaştırılabilir doğruluk sonuçları sağlamaktır.



/ig. 9. Kısmen eğitilmiş modelin hedef etki alanı için kaynak etki alanından alındığı farklı turlar için FTL konseptini uygularken kazanılan zaman.

Bununla birlikte, $A1$ tarafından eğitilen modelin tamamı $A2$ tarafından alındığında bile, yani 200 turdan sonra, $A2$'nin veri kümesindeki yeni örnekler nedeniyle $d2$'de eğitimi gerçekleştirmek için hala biraz zamana ihtiyacı vardır. Sonuç olarak, bu durumda her iki veri kümesi için eğitim süresinin yaklaşık yarısı tasarruf edilebilir.

9. Sonuçlar ve gelecekteki çalışmalar

Bu çalışmada, FL işlemlerini desteklemek için kapsamlı bir uç tabanlı çerçeve çalışması önerdik. Öneri, IoT bağlamları için referans çözümler olan MQTT protokolü ve OMA LwM2M standardı üzerine inşa edilmiştir. Özellikle, MQTT konu formatının yanı sıra hafif iş akışları, farklı FL aşamalarının taleplerini karşılamak için tanımlanmıştır: istemci keşfi, istemci seçimi ve model eğitimi. MQTT mesajlarının yükü OMA LwM2M semantiği ile elden geçirilmiştir.

9.1. Ana bulgular

Gerçekçi bir PoC'de toplanan sonuçlar, farklı veri kümeleri altında daha düşük iletişim ve hesaplama ayak izi açısından popüler bir FL uygulama çözümü olan gRPC'ye kıyasla önerinin üstünlüğünü göstermektedir. Bu da çözümü özellikle IoT senaryoları için uygun hale getirmektedir.

Birlikte çalışabilirlik ayrıca, farklı bir FL toplayıcı ve FL istemci kümesinin birincil bir etki alanında kısmen eğitilmiş modelleri yeniden kullanmasına izin vermenin faydalarını değerlendirerek kanıtlanmıştır.

Tipik olarak kısıtlı IoT cihazları söz konusu olduğundan, genel eğitim prosedürlerinde küçük bir yüzdeyle bile olsa zamandan tasarruf etmek, FL müşterilerinin yerel eğitimi gerçekleştirmek için daha az kaynak harcaması anlamına gelir.

9.2. Açık konular ve gelecekteki araştırma yönleri

Çalışmada daha fazla iyileştirmeye yer vardır.

Güvenlik konularının ele alınması gerekmektedir. Araştırmanın bu aşamasında, karşılaştırılan çözümler güvenlik mekanizmaları olmadan oldukça analiz edilmiştir. MQTT'de uygun güvenlik özelliklerini sağlama sorumluluğunun uygulayıcıya bırakıldığı iyi bilinmektedir. Bir yandan, bu özellik vanilla MQTT uygulamasını güvenlik ve gizlilik saldırılarına açık hale getirmektedir.

Aslında, MQTT şifrelenmemiş iletişim kullanır, bu da mesajların potansiyel olarak üçüncü şahıslar tarafından ele geçirilebileceği ve okunabileceği anlamına gelir. Varsayılan olarak, MQTT zayıf kimlik doğrulama mekanizmaları içerir, bu nedenle herkes potansiyel olarak broker'a bağlanabilir ve mesaj yayınlayabilir. Ayrıca, MQTT mesajların bütünlüğünü doğrulamak için herhangi bir mekanizma sağlamaz, bu nedenle mesajlar potansiyel olarak aktarım sırasında değiştirilebilir.

Öte yandan, MQTT uygulaması, ele alınan ortama göre uyarlanmış güvenlik mekanizmalarını barındıracak kadar esnektir.

MQTT'nin güvenli bir sürümü olan MQTTS, mesajları şifrelemek için Taşıma Katmanı Güvenliği (TLS) protokolünü kullanır. Aynı protokol gRPC tarafından da kullanılabilir. Ancak TLS yüksek hesaplama ve iletişim maliyetlerine sahiptir ve çalışmamızda hedeflendiği gibi IoT cihazları için uygun değildir.

IoT'de MQTT'nin belirli güvenlik sorunlarıyla başa çıkmak için çeşitli çalışmalar önerilmiştir [50-52].

Son zamanlarda, İnternet Mühendisliği Görev Gücü (IETF) IoT alanında sırasıyla kimlik doğrulama ve yetkilendirme ile uçtan uca şifreleme ve bütünlük sunan Kısıtlı Ortamlar için Kimlik Doğrulama ve Yetkilendirme (ACE) [53] ve Kısıtlı RESTful Ortamlar için Nesne Güvenliği (OSCORE) [54] yaklaşımlarını önermiştir. MQTT ile bağlantıları henüz tam olarak araştırılmamış olsa da umut verici görünmektedirler [55,56], dolayısıyla gelecekteki araştırma çalışmalarının önünü açmaktadırlar.

Ayrıca, öneri, tasarımı gelecekteki çalışmaların konusu olacak farklı çok kriterli istemci seçim şemalarını barındırabilir. Ayrıca, ileriye dönük bir adım olarak, daha dinamik ve heterojen bağlamlara daha iyi uyan eşzamansız [12] ve merkezi olmayan [57,58] FL uygulamaları için önerinin etkinliğini ve verimliliğini değerlendirmeyi planlıyoruz.

CRediT yazarlık katkı beyanı

Claudia Campolo: Kavramsallaştırma, Metodoloji, Araştırma, Yazma, Gözden Geçirme, Denetleme, Düzenleme. **Giacomo Genovese:** Kavramsallaştırma, Metodoloji. **Gurtaj Singh:** Kavramsallaştırma, Metodoloji, Yazılım, Doğrulama, Araştırma, Yazma, Düzenleme. **Antonella Molinaro: Kavramsallaştırma,** Metodoloji, İnceleme, Süpervizyon.

Rekabetçi çıkar beyanı

Yazarlar, bu makalede rapor edilen çalışmayı etkileyebilecek bilinen herhangi bir rakip finansal çıkarları veya kişisel ilişkileri olmadığını beyan ederler.

Veri kullanılabilirliği

Veriler talep üzerine kullanıma sunulacaktır.

Referanslar

[1] Nesnelerin interneti. Ericsson, 2022, [Çevrimiçi]. Şu adresten erişilebilir: https://www.ericsson. com/tr/internet-of-things (Erişim: 07 Ekim 2022).

[2] Y. Chen, X. Qin, J. Wang, C. Yu, W. Gao, Fedhealth: Giyilebilir sağlık hizmetleri için birleştirilmiş bir transfer öğrenme çerçevesi, IEEE Intell. Syst. 35 (4) (2020) 83-93.

[3] I. Kevin, K. Wang, X. Zhou, W. Liang, Z. Yan, J. She, Federated transfer learning based cross-domain prediction for smart manufacturing, IEEE Trans. Ind. Inform. 18 (6) (2021) 4088-4096.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Merkezi olmayan verilerden derin ağların iletişim verimli öğrenimi, in: Yapay Zeka ve İstatistik, PMLR, 2017, s. 1273-1282.

[5] O.A. Wahab, A. Mourad, H. Otrok, T. Taleb, Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems, IEEE Commun. Surv. Eğitmen. 23 (2) (2021) 1342-1397.

[6] H. Wang, Z. Kaplan, D. Niu, B. Li, IID olmayan veriler üzerinde federe öğrenmeyi optimize etme pekiştirmeli öğrenme ile, in: IEEE INFOCOM, 2020.

[7] T. Nishio, R. Yonetani, Mobil uçta heterojen kaynaklarla federe öğrenme için istemci seçimi, in: IEEE Uluslararası İletişim Konferansı, ICC, 2019, s. 1-7.

[8] J. Yao, N. Ansari, CPU frekansı ve kablosuz güç kontrolü ile sis destekli IoT'de federe öğrenmeyi geliştirmek, IEEE Internet Things J. 8 (5) (2020) 3438-3445.

[9] S. AbdulRahman, et al., FedMCCS: optimal IoT federe öğrenme için çok kriterli müşteri seçim modeli, IEEE Internet Things J. 8 (6) (2020) 4723-4735.

[10] S. Wang, M. Lee, S. Hosseinalipour, R. Morabito, M. Chiang, C.G. Brinton, Heterojen federe öğrenme için cihaz örneklemesi: Teori, algoritmalar ve uygulama, in: IEEE INFOCOM 2021-IEEE Conference on Computer Communications, IEEE, 2021, pp. 1-10.
[11] J. Lee, H. Ko, S. Seo, S. Pack, Heterojen ağlarda federe öğrenme için veri dağıtımına duyarlı çevrimiçi istemci seçim algoritması, IEEE Trans. Veh. Technol. (2022).
[12] L.U. Khan, W. Saad, Z. Han, E. Hossain, C.S. Hong, Nesnelerin interneti için federe öğrenme: Son gelişmeler, taksonomi ve açık zorluklar, IEEE Commun. Hayatta Kalma. Eğitmen. 23 (3) (2021) 1759-1799.
[13] A. Imteaj, U. Thakker, S. Wang, J. Li, M.H. Amini, A survey on federated learning for resource-constrained IoT devices, IEEE Internet Things J. 9 (1) (2021) 1-24.
[14] G. Genovese, G. Singh, C. Campolo, A. Molinaro, MQTT ve OMA lightweight-M2M aracılığıyla uç tabanlı federe öğrenmeyi etkinleştirme, in: IEEE VTC Spring, 2022.
[15] A. Banks, R. Gupta, MQTT Sürüm 3.1.1, OASIS Standardı, 2014.
[16] Açık mobil ittifak, hafif makineden makineye teknik şartname core; V1_1-20180612-C, 2018.
[17] G. Cleland, D. Wu, R. Ullah, B. Varghese, FedComm: Uç tabanlı federe öğrenme için iletişim protokollerini anlama, 2022, arXiv ön baskı arXiv: 2208.08764.
[18] A. Feraudo, ve diğerleri, Colearn: MUD uyumlu IoT uç ağlarında federe öğrenmeyi etkinleştirme, in: EdgeSys, 2020, s. 25-30.
[19] B.C. Tedeschini, S. Savazzi, R. Stoklasa, L. Barbieri, I. Stathopoulos, M. Nicoli, L. Serio, Sağlık ağları için merkezi olmayan federe öğrenme: Tümör segmentasyonu üzerine bir vaka çalışması , IEEE Access 10 (2022) 8693-8708.
[20] D.C. Nguyen, M. Ding, P.N. Pathirana, A. Seneviratne, J. Li, H.V. Poor, Nesnelerin interneti için federe öğrenme: Kapsamlı bir araştırma, IEEE Commun. Surv. Eğitmen. 23 (3) (2021) 1622-1658.
[21] D. Glaroudis, A. Iossifides, P. Chatzimisios, Survey, comparison and research challenges of IoT application protocols for smart farming, Comput. Netw. 168 (2020) 107037.
[22] V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, J. Alonso-Zarate, A survey on application layer protocols for the Internet of things, Trans. IoT Cloud Comput. 3 (1) (2015) 11-17.
[23] OMA-LwM2M kayıt defteri, 2022, [Çevrimiçi]. Şu adresten erişilebilir: https://technical. openmobilealliance.org/OMNA/LwM2M/LwM2MRegistry.html (Erişim: 27 Ekim 2022).
[24] OMA-LwM2M nesne kaynak düzenleyicisi, 2022, [Çevrimiçi]. Şu adresten erişilebilir: https:// github.com/OpenMobileAlliance/ (Erişim: 27 Ekim 2022).
[25] ETSI GS MEC 003 v1.1.1. Mobile Edge Computing (MEC); Framework and Reference Architecture, ETSI, 2016.
[26] Mosquitto, MQTT uygulaması, https://mosquitto.org/.
[27] N. Tonellotto, A. Gotta, F.M. Nardini, D. Gadler, F. Silvestri, Neural network quantization in federated learning at the edge, Inform. Sci. 575 (2021) 417-436.
[28] B. Fortner, HDF: Hiyerarşik veri formatı, Dr Dobb's J. Softw. Tools Prof. Program 23 (5) (1998) 42.
[29] Kaydetme ve yükleme modelleri, 2022, [Çevrimiçi]. Şu adresten erişilebilir: https://www.tensorflow.org/ tutorials/keras/save_and_load?hl=en (Erişim: 07 Ekim 2022).
[30] B. Camajori Tedeschini, S. Savazzi, R. Stoklasa, L. Barbieri, I. Stathopoulos, M. Nicoli, L. Serio, Sağlık ağları için merkezi olmayan federe öğrenme: Tümör segmentasyonu üzerine bir vaka çalışması, IEEE Access 10 (2022) 8693-8708, http://dx.doi.org/10.1109/ACCESS.2022.3141913.
[31] W.Y.B. Lim, N.C. Luong, D.T. Hoang, Y . Jiao, Y . -C. Liang, Q. Yang, D. Niyato, C. Miao, Mobil uç ağlarda federe öğrenme: Kapsamlı bir araştırma, IEEE Commun. Sürv. Eğitmen. 22 (3) (2020) 2031-2063.
[32] J. Dizdarević, F. Carpio, A. Jukan, X. Masip-Bruin, A survey of communication protocols for Internet of things and related challenges of fog and cloud computing integration, ACM Comput. Surv. 51 (6) (2019) 1-29.
[33] B. Mishra, A. Kertesz, M2M ve IoT sistemlerinde MQTT kullanımı: Bir anket, IEEE Access 8 (2020) 201071-201086.
[34] D. Merkel, Docker: Tutarlı geliştirme ve dağıtımı için hafif linux konteynerleri, Linux J. 2014 (239) (2014) 2.
[35] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, ve diğerleri, TensorFlow: Büyük ölçekli makine öğrenimi için bir sistem, in: 12. USENIX İşletim Sistemleri Tasarımı Sempozyumu ve Uygulaması, OSDI 16, 2016, s. 265-283.
[36] E. Bertrand-Martínez, P.D. Feio, V. de Brito Nascimento, B. Pinheiro, A. Abelém, IoT brokerlerinin sınıflandırılması ve değerlendirilmesi için bir metodoloji, in: LANOMS, 2019.
[37] M. Bender, E. Kirdan, M.-O. Pahl, G. Carle, Açık kaynaklı MQTT değerlendirmesi, in: 2021 IEEE 18th Annual Consumer Communications & Networking Conference, CCNC, IEEE, 2021, pp. 1-4.
[38] A. Krizhevsky, G. Hinton ve diğerleri, Learning multiple layers of features from tiny images, Citeseer, 2009.
[39] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradyan tabanlı öğrenmenin belge tanımaya uygulanması, Proc. IEEE 86 (11) (1998) 2278-2324.

[40] Z. Zhang, L. Wu, D. He, Q. Wang, D. Wu, X. Shi, C. Ma, G-VCFL: Gruplandırılmış doğrulanabilir zincirleme gizliliği koruyan federe öğrenme, IEEE Trans. Netw. Hizmet Yönetimi. Manag. (2022).
[41] J. Li, Y. Shao, K. Wei, M. Ding, C. Ma, L. Shi, Z. Han, H.V. Poor, Blockchain destekli merkezi olmayan federe öğrenme (BLADE-FL): Performans analizi ve kaynak tahsisi, IEEE Trans. Paralel Dağıtım. Syst. 33 (10) (2021) 2401-2415.
[42] R.C. Çalik, M.F. Demirci, Cifar-10 görüntüsünün konvolüsyonel sinir sistemi ile sınıflandırılması
gömülü sistemler için ağlar, içinde: IEEE/ACS 15th International Conference on Computer Systems and Applications, AICCSA, 2018, pp. 1-2.
[43] L. Hertel, E. Barth, T. Käster, T. Martinetz, Derin evrişimli sinir ağları as generic feature extractors, 2017, arXiv.
[44] gRPC, google uzaktan yordam çağrısı, 2022, [Çevrimiçi]. Şu adresten erişilebilir: https://grpc. io./ (Erişim: 07 Ekim 2022).
[45] D.J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, P.P. de Gusmão, N.D. Lane, Çiçek: A friendly federated learning research framework, 2020, arXiv preprint a r X i v :2007.14390.
[46] P. Pinyoanuntapong, P. Janakaraj, R. Balakrishnan, M. Lee, C. Chen, P. Wang, EdgeML: Towards network-accelerated federated learning over wireless edge, Comput. Netw. (2022) 109396.
[47] I. Kholod, E. Yanaki, D. Fomichev, E. Shalugin, E. Novikova, E. Filippov, M. Nordlund, IoT için açık kaynaklı federe öğrenme çerçeveleri: A comparative review and analysis, Sensors 21 (1) (2020) 167.
[48] Mininet, http://mininet.org/.
[49] T. Yokotani, Y. Sasaki, IoT için gerekli ağ kaynakları üzerinde HTTP ve MQTT ile karşılaştırma, in: 2016 Uluslararası Kontrol, Elektronik, Yenilenebilir Enerji ve İletişim Konferansı, ICCEREC, IEEE, 2016, s. 1-6.
[50] H. Mrabet, S. Belguith, A. Alhomoud, A. Jemai, IoT güvenliği tabanlı bir araştırma katmanlı bir algılama ve veri analizi mimarisi üzerine, Sensors 20 (13) (2020) 3625.
[51] E. Lee, Y.-D. Seo, S.-R. Oh, Y.-G. Kim, Birlikte çalışabilirlik için standartlar üzerine bir araştırma
and security in the Internet of Things, IEEE Commun. Surv. Eğitmen. 23 (2) (2021) 1020-1047.
[52] C.-S. Park, H.-M. Nam, Güvenli MQTT-SN için güvenlik mimarisi ve protokolleri, IEEE Access 8 (2020) 226422-226436.
[53] F. Palombini, C. Sengul, Kısıtlı ortamlar için kimlik doğrulama ve yetkilendirme için Pub-sub profili (ACE), in: Internet Engineering Task Force, IETF, 2022.
[54] F.P.G. Selander, J. Mattsson, L. Seitz, Kısıtlı RESTful için nesne güvenliği ortamları (OSCORE), içinde: İnternet Mühendisliği Görev Gücü, no. RFC 8613, IETF, 2019.
[55] Z. Laaroussi, O. Novo, Güvenlik iletişiminin performans analizi CoAP ve MQTT'de, içinde: 2021 IEEE 18th Annual Consumer Communications & Networking Conference, CCNC, IEEE, 2021, pp. 1-6.
[56] V. Seoane, C. Garcia-Rubio, F. Almenares, C. Campo, Performance evaluation of IoT ortamları için güvenlik destekli CoAP ve MQTT, Comput. Netw. 197 (2021) 108338.
[57] H.T. Nguyen, R. Morabito, K.T. Kim, M. Chiang, Anında kaynak farkında model heterojen kenarda federe öğrenme için toplama, içinde: 2021 IEEE Global Communications Conference, GLOBECOM, IEEE, 2021, pp. 1-6.
[58] S. Savazzi, M. Nicoli, V. Rampa, İşbirliği yapan cihazlarla federe öğrenme: Büyük IoT ağları için bir fikir birliği yaklaşımı, IEEE Internet Things J. 7 (5) (2020) 4641-4654.

Claudia Campolo, İtalya'daki Reggio Calabria Mediterranea Üniversitesi'nde Telekomünikasyon alanında doçent olarak görev yapmaktadır. İtalya'daki Reggio Calabria Mediterranea Üniversitesi'nden Telekomünikasyon Mühendisliği alanında Laurea derecesi (2007) ve doktora derecesi (2011) almıştır. Mevcut görevinden önce, Reggio Calabria Mediterranea Üniversitesi'nde Yardımcı Doçent olarak çalışmıştır (2012-2020). Başlıca araştırma alanları araç ağları ve 5G sistemleridir.

Giacomo Genovese, 2010 ve 2014 yıllarında İtalya'daki Reggio Calabria Akdeniz Üniversitesi'nden Telekomünikasyon Mühendisliği alanında lisans ve yüksek lisans derecelerini almıştır. 2015 yılında CTTC of Castelldefels, Barselona, İspanya'da genç bir araştırma mühendisi olmuştur. H2020 INPUT projesi gibi çeşitli ulusal ve Avrupa projelerinde Araştırma Mühendisi olarak görev yapmıştır. Şu anda, İtalya'daki Reggio Calabria Akdeniz Üniversitesi ARTS Laboratuvarı'nda genç araştırmacı olarak çalışmaktadır. Araştırma ilgi alanları IoT heterojen ağ geçitleri tasarımı, IoT cihaz sanallaştırma, uç bilişim teknolojileri alanındadır.

Gurtaj Singh, Reggio Calabria Akdeniz Üniversitesi'nden Formasyon Mühendisliği alanında Lisans Derecesi (2018) ve Telekomünikasyon Mühendisliği alanında Yüksek Lisans Derecesi (2020) almıştır. 2021 yılında Alten'de telekomünikasyon mühendisi olarak çalıştı ve İtalyan topraklarında optik yolların tasarımı için Huawei ve Vodafone'u takip etti. Ocak 2022'den beri İtalya'daki Reggio Calabria Akdeniz Üniversitesi'nde doktora öğrencisi olarak çalışmaktadır. Araştırma alanları dağıtık yapay zeka, IoT için ağ protokolleri, uç bilişim teknolojileri ile ilgilidir.

Antonella Molinaro, Reggio Calabria Mediterranea Üniversitesi'nde telekomünikasyon alanında profesör olarak görev yapmaktadır. Daha önce Messina Üniversitesi (1998-2001) ve Calabria Üniversitesi'nde (2001-2004) yardımcı doçent ve Politecnico di Milano'da (1997-1998) araştırma görevlisi olarak çalışmıştır. Telesoft, Roma (1992-1993) ve Siemens, Münih'te (1994-1995) çalışmıştır. Şu anki araştırmaları 5G, araç ağları ve geleceğin İnternet mimarileri üzerine odaklanmaktadır.