

Makale

Nesnelerin İnterneti için Federe Öğrenmenin Hesaplama ve İletişim Verimli Uyarlanabilir Federe Optimizasyonu

Zunming Chen ¹, Hongyan Cui ^{2,*}, Ensen Wu ² ve Xi Yu ²¹ Ağ ve Anahtarlama Teknolojisi Devlet Anahtar Laboratuvarı, Pekin Posta ve Telekomünikasyon Üniversitesi, Pekin 100876, Çin; czm@bupt.edu.cn² Bilgi ve İletişim Mühendisliği Fakültesi, Pekin Posta ve Telekomünikasyon Üniversitesi, Pekin 100876, Çin; wuensen@bupt.edu.cn (E.W.); yusy@bupt.edu.cn (X.Y.)

* Yazışma adresi: cuihy@bupt.edu.cn

Özet: Nesnelerin İnternetinin (IoT) yaygınlaşması ve algılama, hesaplama ve iletişim yeteneklerine sahip cihazların yaygın kullanımı, yapay zeka ile güçlendirilmiş akıllı uygulamaları motive etmiştir. Klasik yapay zeka algoritmaları, artan veri gizliliği endişeleri ve dağıtılmış veri kümeleri nedeniyle gerçekçi akıllı IoT uygulamalarında zorlayıcı olan merkezi veri toplama ve işleme gerektirir. Federated Learning (FL), IoT cihazlarının model parametrelerini paylaşarak küresel modelleri eğitmesini sağlayan, gizliliği koruyan dağıtılmış bir öğrenme çerçevesi olarak ortaya çıkmıştır. Ancak, sık parametre aktarımından kaynaklanan verimsizlik FL performansını önemli ölçüde düşürmektedir. Mevcut hızlandırma algoritmaları, yerel güncelleme ve parametre sıkıştırma olmak üzere iki ana türden oluşur ve bunlar sırasıyla iletişim ve gönderim/hassasiyet arasındaki ödünleşimleri dikkate alır. Bu iki ödünleşimin birlikte dikkate alınması ve yakınsama üzerindeki etkilerinin uyarlanabilir bir şekilde dengelenmesi çözülmemiştir. Bu makale, sorunu çözmek için FL'nin verimliliğini artırmak üzere yerel güncelleme ve parametre sıkıştırma dahil olmak üzere iki değişkeni birlikte dikkate alarak öğrenme hatasını en aza indiren yeni bir verimli uyarlanabilir federe optimizasyon (FedEAFO) algoritması önermektedir. FedEAFO, FL'nin iki değişkeni uyarlamalı olarak ayarlamasını ve hesaplama, iletişim ve hassasiyet arasındaki dengeyi kurmasını sağlar. Deney sonuçları, son teknoloji algoritmalarla karşılaştırıldığında, FedEAFO'nun daha yüksek doğruluklara daha hızlı ulaşabildiğini göstermektedir.



Atıf: Chen, Z.; Cui, H.; Wu, E.; Yu, X. Nesnelerin İnterneti için Federe Öğrenmenin Hesaplama ve İletişim Verimli Uyarlanabilir Federe Optimizasyonu.

Elektronik **2023**, *12*, 3451. <https://doi.org/10.3390/electronics12163451>

Akademik Editör: Imran

Alındı: 21 Temmuz 2023

Revize: 11 Ağustos 2023

Kabul edildi: 13 Ağustos 2023

Yayınlanma tarihi: 15 Ağustos 2023



Telif hakkı: © 2023 yazarlar tarafından. Lisans sahibi MDPI, Basel, İsviçre. Bu makale Creative Commons Attribution (CC BY) lisansının hüküm ve koşulları altında dağıtılan açık erişimli bir makaledir (<https://creativecommons.org/licenses/by/4.0/>).

Anahtar Kelimeler: federe öğrenme; dağıtık makine öğrenimi; iletişim verimliliği; gizlilik koruması

1. Giriş

Cihazlardan gelen veri miktarındaki patlayıcı büyüme, nesneleri internete bağlamak için her yerde algılama, hesaplama ve iletişim yetenekleri sağlayan Nesnelerin İnterneti'nin (IoT) hızlı gelişimine tanık olmuştur [1]. IoT cihazlarından gelen veriler için derin analiz sağlamak amacıyla, akıllı ulaşım ve akıllı şehir gibi akıllı IoT uygulamalarını mümkün kılmak için yapay zeka (AI) algoritmaları benimsenmiştir [2]. Geleneksel olarak, verilerin merkezi olarak toplanmasına ve işlenmesine ihtiyaç duyan YZ algoritmaları, veri madenciliği için merkezi bir bulut/uç sunucu veya veri merkezinde konuşlandırılır. Ancak, büyük miktarda IoT verisinin uzak sunuculara aktarılması ve verilerin uzak sunucularda işlenmesi önemli gecikmelere neden olmaktadır. Ayrıca, üçüncü taraf sunucular veri gizliliği endişelerini de beraberinde getirmektedir [3]. Bu bağlamda, gizliliğin korunması ve dağıtık yapay zekanın IoT'ye entegre edilmesi önemli bir konu haline gelmektedir.

Son zamanlarda, Federated Learning (FL), dağıtılmış IoT cihazlarının makine öğrenimi modellerini işbirliği içinde eğitmesine izin vererek akıllı IoT uygulamalarına olanak tanıyan gizliliği koruyan dağıtılmış bir öğrenme çerçevesi olarak ortaya çıkmıştır [4]. FL, birden fazla cihazın Stokastik Gradyan İnişi (SGD) yoluyla ortak bir küresel modeli eğitmesini ve ham veriler yerine yerel model parametrelerini paylaşmasını sağlar [5]. FL, son zamanlarda çeşitli uygulamalarda başarılar elde etmiştir. Örneğin, [6]'da önerilen Genetik Kümelenmiş Federe Öğrenme (CFL) uygulanmıştır

COVID-19 hastalarını gizliliği koruyan bir şekilde tespit etmek için. 7]'deki FL destekli işbirlikçi algılama görevi, araçların sürüş ortamının genişletilmiş bir vizyonunu elde etmek için sensör verilerini birleştirdiği araç ağlarına uygulanmıştır.

Bununla birlikte, mevcut birleştirilmiş öğrenme yöntemleri, çeşitli nedenlerden dolayı kaynak kısıtlı IoT ortamlarında uygulandığında model eğitim performansında verimsizlikten muzdariptir. (i) Dengesiz Veri: IoT cihazlarındaki eğitim verileri, farklı algılama ortamları nedeniyle boyut ve dağılım açısından farklıdır [8]. Bu nedenle, FL'deki yukarı bağlantı ve aşağı bağlantıdaki iletişimler, bağımsız olmayan ve aynı şekilde dağıtılmış (IID) verilere tabi olarak oldukça hassastır. (ii) Sınırlı Bant Genişliği: IoT cihazları, kısıtlı ağ yetenekleri nedeniyle genellikle sınırlı iletişim bant genişliğine sahiptir. Geleneksel federe öğrenme kurulumlarında büyük model güncellemelerinin iletilmesi bu tür ortamlarda verimsiz ve yavaş olabilir. (iii) Yüksek Gecikme Süresi: IoT cihazları yüksek iletişim gecikmesine sahip olabilir ve bu da sık model güncellemelerini pratik olmaktan çıkarır. Bu, güncellemelerin toplanmasında gecikmelere yol açabilir ve eğitim sürecini engelleyebilir. (iv) Güvenilmez Bağlantı: IoT cihazları kesintili veya güvenilmez bağlantı yaşayabilir. Bu durum iletişim sürecini kesintiye uğratabilir ve eksik veya gecikmeli güncellemelere yol açabilir. Kesintili bağlantılar nedeniyle IoT için FL'nin yakınsaması her zaman garanti edilemez [9-11]. Mobil cihazlarda (örneğin sensörler ve İnsansız Hava Araçları (İHA'lar)) federe öğrenmenin durumu, kablosuz kanallar üzerinden iletişim kurdukları ve daha düşük bant genişliği, daha yüksek gecikme süresi ve kesintili bağlantılardan muzdarip oldukları için daha da kötüleşir [12]. Böylece, verimsiz eğitim performansı sorunu FL'nin ölçeklendirilmesi için önemli bir darboğaz haline gelmektedir.

Federe öğrenmenin verimsiz eğitim performansı sorununu çözmek gereklidir. Yerel güncelleme ve parametre sıkıştırma yoluyla federe öğrenme yakınsamasını hızlandırmak için çeşitli çalışmalar yapılmıştır [13,14]. Yerel güncelleme yaklaşımı, cihazların hesaplama kapasitesinden tam olarak yararlanarak model parametrelerinin sık aktarımını azaltmayı amaçlamaktadır. Yerel güncelleme algoritmaları, yerel güncellenmenin bant genişliğine oranını belirleyen önerilen bir yerel güncelleme katsayısı kavramı aracılığıyla hesaplama ve iletişim arasında bir ödünleşimi karakterize eder [15]. Parametre sıkıştırma yaklaşımı, niceleme ve seyrekleştirme gibi veri sıkıştırma teknikleri aracılığıyla iletilen veri miktarını azaltmayı amaçlamaktadır. Parametre sıkıştırma algoritmaları, bant genişliği ve doğruluk arasındaki oranı belirleyen parametre sıkıştırma katsayısı (sıkıştırma bütçesi) aracılığıyla iletişim ve model hassasiyeti arasındaki dengeyi karakterize eder [16]. Bu yöntemler, federe öğrenmenin verimliliğini artırmak için ayrı ayrı çalışılmıştır. Ancak, bu iki ödünleşimin birlikte ele alınması ve bunların hem matematiksel tahmin hem de teorik analiz perspektiflerinden federe öğrenmenin yakınsaması üzerindeki etkilerinin uyarlanabilir bir şekilde dengelenmesi henüz çözülmemiştir. Bu önemli sorun araştırmamızı motive etmiştir. Bu makale, IoT için federe öğrenmenin yakınsamasını hızlandırmak için yerel güncelleme ve parametre sıkıştırma dahil olmak üzere iki yöntemi birlikte değerlendirerek öğrenme hatasını en aza indiren yeni bir verimli uyarlanabilir federe optimizasyon (FedEAFO) algoritması önermektedir. FedEAFO, federe öğrenmenin iki değişkeni uyarlamalı olarak ayarlamasına ve hesaplama, iletişim ve doğruluk arasındaki dengeyi sağlamasına olanak tanır.

Önerilen verimli uyarlanabilir federasyon optimizasyon algoritması, geleneksel optimizasyon yaklaşımlarından birkaç önemli şekilde ayrılmaktadır. Bu farklılıklar, algoritmanın ilerlemelerini ve IoT ortamlarının karmaşıklığı için özel olarak tasarlanmış uygunluğunu ortaya koymaktadır. (i) Ortak İletişim ve Hesaplama Optimizasyonu: FedEAFO algoritmasının en önemli farklarından biri optimizasyona yönelik bütünsel yaklaşımıdır. Genellikle ya iletişim ya da hesaplama sorunlarına odaklanan geleneksel yöntemlerin aksine FedEAFO, hem model sıkıştırma tekniklerini hem de çoklu yerel eğitim stratejilerini sorunsuz bir şekilde entegre ederek öncü bir adım atmaktadır. Bu entegre optimizasyon, federe öğrenme çerçevesi içinde hesaplama ve iletişim verimliliği arasındaki karmaşık etkileşimi etkili bir şekilde ele alır. (ii) Uyarlanabilir Optimizasyon: FedEAFO, kendisini geleneksel tekniklerden ayıran uyarlanabilir bir boyut sunar. Bu uyarlanabilirlik, algoritmanın optimizasyon parametrelerini dinamik ve akıllı bir şekilde ayarlamasını sağlayarak çeşitli ödünleşimler arasında ince ayarlı bir denge kurulmasını kolaylaştırır. Bu uyarlanabilirlik çok açık bir şekilde

Genellikle sabit veya önceden belirlenmiş parametreler kullanan geleneksel yöntemlerin aksine, FedEAFO'yu IoT ortamlarının dinamik doğasına özellikle duyarlı hale getirir.

(iii) Geliştirilmiş Yakınsama Hızı: FedEAFO'nun bir diğer ayırt edici özelliği, federe öğrenmenin yakınsama hızını artırma hedefinde yatmaktadır. IoT ortamlarının getirdiği özel zorluklara uyum sağlamakta zorlanabilecek bazı geleneksel yöntemlerin aksine FedEAFO, model sıkıştırma ve yerel eğitimi stratejik olarak entegre eder. Bu sinerji, öğrenme güncellemelerini optimize ederek ve iletişim kaynaklarını aynı anda koruyarak daha hızlı yakınsamayı teşvik eder. (iv) IoT Kısıtlamalarına Uyarılma: Geleneksel optimizasyon yöntemleri, IoT ortamlarının benzersiz kısıtlamalarını ve karmaşıklıklarını tam olarak hesaba katmayabilir. Öte yandan FedEAFO, IoT cihazlarının kısıtlı bant genişliği, yüksek gecikme süresi ve enerji hususları gibi sınırlamalarına uyum sağlamak için titizlikle hazırlanmıştır. Bu faktörleri optimizasyon hedefleriyle uyumlu hale getirme becerisi, genel optimizasyon paradigmalarından önemli bir ayrımı işaret etmektedir.

Bu araştırma, IoT ortamlarına uygulandığında, cihazların sınırlı bant genişliği, güvenilir olmayan ağ bağlantısı, cihaz heterojenliği ve dengesiz veri gibi sınırlamalar nedeniyle ortaya çıkan federe öğrenmenin karşılaştığı benzersiz verimsizlik eğitim performansı sorununu ele alan yeni bir yaklaşım sunmaktadır. Bu araştırma, IoT cihazları ile merkezi sunucu arasında iletilmesi gereken model güncellemelerinin boyutunu azaltmayı amaçlayan IoT cihazlarına özel yenilikçi model sıkıştırma teknikleri önermektedir. Modeli etkili bir şekilde sıkıştırarak, bu araştırma IoT cihazlarının sınırlı iletişim bant genişliğinin verimli bir şekilde kullanılmasını sağlar. Buna ek olarak, IoT cihazlarının yüksek gecikme süresi ve sınırlı enerji kaynaklarının farkında olan bu araştırma, IoT cihazlarının güncellemeleri merkezi sunucuya iletmenden önce kendi yerel verileri üzerinde birden fazla eğitim turu gerçekleştirmesini sağlayan çoklu yerel eğitim yöntemlerinin dahil edilmesini önermektedir. Bu, sık iletişim ihtiyacını azaltarak yüksek gecikme süresi ve güvenilir olmayan bağlantının yarattığı zorlukları ortadan kaldırabilir. Ayrıca bu araştırma, donanım yetenekleri, ağ koşulları ve veri dağılımları açısından IoT cihazlarının doğal heterojenliğini dikkate almaktadır. Önerilen algoritma, model sıkıştırma ve çoklu yerel eğitimi birlikte değerlendirerek çeşitli IoT sistemine uyum sağlar ve farklı yeteneklere sahip cihazlarla uyumluluk sağlar. Son olarak, IoT cihazları genellikle çok çeşitli hesaplama yetenekleri ve veri dağılımı sergiler. Önerilen verimli uyarlanabilir optimizasyon yöntemi, uyarlanabilir çoklu yerel eğitimi dahil ederek bu heterojenlikleri ele alır. Önerilen algoritma, cihaz heterojenliğinin ve veri dengesizliğinin etkisini azaltmak için cihazların yerel güncelleme sıklıklarını ayarlayabilir. Bu yöntem, her IoT cihazının mevcut hesaplama kaynaklarını kullanarak birkaç tur eğitim gerçekleştirmesine olanak tanır. Daha yüksek hesaplama kapasitesine sahip cihazlar daha fazla yerel eğitim turu gerçekleştirebilir ve federe öğrenme sürecine daha etkili bir şekilde katkıda bulunabilir. Buna ek olarak, daha küçük veri kümelerine sahip cihazların merkezi sunucu ile iletişim kurmadan önce daha fazla yerel güncelleme yapmasına izin verilebilir, bu da seyrek veriye sahip cihazların daha fazla veriye sahip olanları yakalamasını sağlayarak veri dengesizliğinden kaynaklanan eşitsizlikleri azaltır. FedEAFO'nun temel katkıları aşağıdaki gibi özetlenebilir:

- Bu makale, sırasıyla iletişim ve hesaplama/model hassasiyeti arasındaki değiş tokuşları karakterize eden yerel güncelleme katsayısı ve sıkıştırma bütçesi açısından küresel modelin hatasını en aza indiren pratik bir formülasyonla federe öğrenme problemini araştırmaktadır.
- Bu makale, yerel güncelleme ve sıkıştırma katsayısı dahil olmak üzere iki değişkeni dikkate alan türetilmiş bir hata üst sınırı kullanarak, federe öğrenmenin verimliliğini artırmak için bu iki değişkeni uyarlamalı olarak ayarlayan yeni bir verimli uyarlamalı federe optimizasyon algoritması önermektedir.
- Önerilen algoritmanın teorik analizinin yanı sıra, FedEAFO'nun iki veri kümesi üzerinde, daha yüksek doğruluklara daha hızlı ulaşan diğer son teknoloji yöntemlerle karşılaştırıldığında güçlü deneysel performans gösteriyoruz.

Önerilen verimli uyarlanabilir federasyon optimizasyon algoritması, eğitim verimliliği, heterojenlik, IID olmayan veriler ve diğer zorlukları ele alan umut verici bir yaklaşımdır.

dağıtılmış bir makine öğrenimi ortamında gizlilik endişeleri. Çeşitli sektörlerde çeşitli pratik uygulamalar sunmaktadır: (i) Otonom Araçlar: Otonom araçlar çalışma sırasında çok sayıda veri üretir. Verimli uyarlanabilir federe öğrenme ile araçlar, kullanıcı gizliliğine saygı gösterirken güvenliği ve navigasyonu iyileştirmek için birbirlerinin deneyimlerinden toplu olarak öğrenebilir. (ii) Finansal Hizmetler: Bankalar ve finans kurumları genellikle veri gizliliği düzenlemeleriyle karşı karşıya kalır ve bu da veri paylaşımını zorlaştırır. Verimli uyarlanabilir federe öğrenme ile bankalar, hassas finansal bilgileri doğrudan paylaşmadan müşteri davranışlarını toplu olarak analiz edebilir ve hileli faaliyetleri tespit edebilir. Müşteri gizliliğini korurken gelişmiş risk değerlendirmesi, kişiselleştirilmiş finansal öneriler ve dolandırıcılık tespiti sağlar. (iii) Akıllı Şebekeler: Enerji sektöründe, akıllı şebekeler çok sayıda enerji tüketen ve enerji üreten cihazdan oluşur. Federe öğrenme, veri gizliliğini ve merkezi olmayan karar alma sürecini korurken enerji tüketimini optimize etmeye, enerji talebini tahmin etmeye ve şebeke istikrarını yönetmeye yardımcı olabilir. (iv) Üretim: Akıllı fabrikalarda, farklı makineler ve sensörler farklı özelliklere sahip veriler üretir. Verimli uyarlanabilir federatif öğrenme, merkezi veri toplamaya gerek kalmadan kestirimci bakım, süreç optimizasyonu ve kalite kontrolü sağlayarak üretim verimliliğini artırır ve arıza süresini azaltır.

2. İlgili Çalışma

2.1. Federe Öğrenmede Yerel Güncelleme

Federe öğrenmenin verimliliğini artırmaya ve iletişim darboğazlarının üstesinden gelmeye yönelik yaklaşımlar yerel güncelleme ve parametre sıkıştırma olarak kategorize edilebilir. Yerel güncelleme yaklaşımı, model parametrelerinin sık iletimini azaltmak için cihazların hesaplama kapasitesinden tam olarak yararlanmayı amaçlamaktadır. Örneğin, Nenghai ve arkadaşları, dağıtılmış gradyan inişinin yakınsama sınırını türeten ve küresel modelin toplanmasından önce yerel güncellenmenin yalnızca bir adımına izin veren Asenkron Stokastik Gradyan İnişi (ASGD) algoritmasını önermiştir [17]. Virginia ve arkadaşları, hesaplamanın iletişime oranını belirleyen yerel model güncelleme katsayısı kavramını ortaya koyan ve sabit bir yerel güncelleme katsayısı ile çoklu yerel güncellemeler gerçekleştiren Fedprox'u önermiştir [18]. Joshi ve arkadaşları, cihazların heterojen hesaplama ve iletişim yetenekleri sorununu çözmek için yerel güncelleme katsayısını ayarlamak ve hesaplama ile iletişim arasındaki dengeyi dinamik olarak ayarlamak için uyarlanabilir bir iletişim stratejisinin benimsendiği ADACOMM'u önermiştir [19]. Kevin ve arkadaşları, IID olmayan ortamlarda federe öğrenmeyi hızlandırmak için belirli bir kaynak bütçesi altında en iyi yerel güncelleme katsayısını belirlemek için uyarlanabilir bir strateji benimseyen bir AFD önermiştir [20].

2.2. Federe Öğrenmede Parametre Sıkıştırma

Parametre sıkıştırma yaklaşımı, iletilen veri miktarını önemli ölçüde azaltmak için seyrekleştirme ve niceleme gibi veri sıkıştırma algoritmalarını benimser. Parametre sıkıştırma algoritması, iletişim ve model hassasiyeti arasındaki dengeyi, iletişimin hassasiyete oranını belirleyen bir sıkıştırma katsayısı ile karakterize eder. Parametre sıkıştırma algoritmaları, model parametrelerinin kuantize edilmiş versiyonunu veya seyrek gösterimini yükleyerek iletişim ek yüklerini azaltır. Örneğin, Wen ve arkadaşları her bir parametreyi 2 bite kuantize eden Terngrad'ı önermiştir [21]. Alistarh ve arkadaşları, model ağlarının çeşitli katmanlarını nicelleştirmek için 2 bit ve 4 bit kullanan Niceleme Stokastik Gradyan İnişi (QSGD) algoritmasını önermiştir [22]. Parametreyi 1 bite kuantize eden 1 bitlik bir SGD [23]'te çalışılmıştır. Seyrekleştirme tekniği, tüm parametreler yerine önemli parametreleri göndermek için parametreleri seyrekleştirmeyi amaçlamaktadır. Seyrekleştirme teknikleri, seyrekliğin arandığı alana göre ham alan ve dönüştürülmüş alan olmak üzere iki kategoriye ayrılabilir. Örneğin, Dally ve arkadaşları, önemsiz elemanların değerlerini en üst %0,05 ile en alt %0,05 arasında olacak şekilde tanımlayan bir eşik aracılığıyla önemsiz elemanları sıfıra ayarlamak için bir parametre seyrekleştirme stratejisi önermiştir [24]. Wright ve arkadaşları ATOMO'yu önermişlerdir.

Daha fazla seyreklik elde etmek için düşük boyutlu yapıdan yararlanmak üzere parametreleri Tekil Değer Ayrıştırma (SVD) alanına dönüştürerek iletişim ek yüklerini azaltır [25].

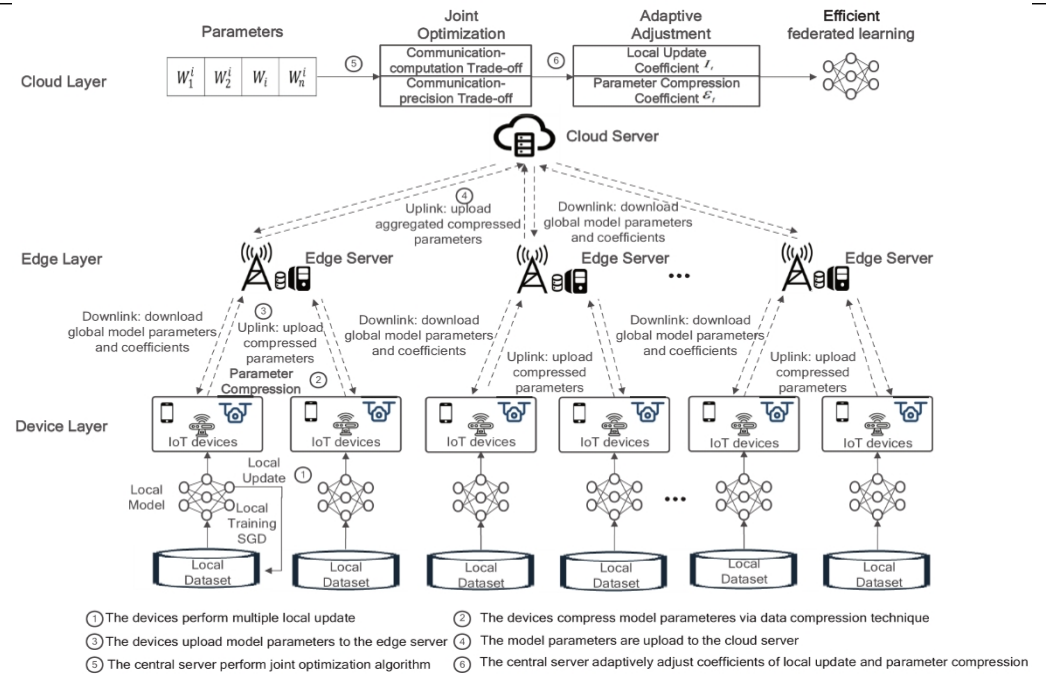
İlgili çalışmaların özeti Tablo 1'de listelenmiştir. Yukarıda bahsedilen yöntemler, iletişim darboğazlarının üstesinden gelmek ve federe öğrenmenin verimliliğini artırmak için ayrı ayrı çalışılmıştır. İki yaklaşımın entegre edilmesinin federe öğrenmeyi hızlandırmada daha etkili olması beklenmektedir. Ancak, iki yaklaşımın entegre edilmesi ve iletişim ile hesaplama/hassasiyet arasındaki ödünleşimlerin birlikte ele alınarak bunların federe öğrenmenin yakınsaması üzerindeki etkilerinin uyarlanabilir bir şekilde dengelenmesi ve ayarlanması konusu henüz çözüme kavuşturulmamıştır. Bu nedenle, mevcut yaklaşımları tamamlamak için iletişim, hesaplama ve hassasiyet arasındaki ödünleşimleri ortaklaşa dengeleyerek verimli bir uyarlanabilir federe optimizasyon algoritması tasarlamaya acil bir ihtiyaç vardır.

Tablo 1. Etkin federe öğrenme yaklaşımlarının özeti.

Ref.	Meydan Okuma	Teknik	Anahtar Fikir
[18]	İletişim frekansının azaltılması gerçeğeştirmesini sağlayın	Yerel güncelleme	Cihazların sabit bir yerel güncelleme ile birden fazla yerel güncelleme katsayısı
[19]	İletişim frekansının azaltılması benimsenmiştir	Yerel güncelleme	Yerel güncellemeyi ayarlamak için uyarlanabilir bir strateji dinamik olarak katsayı
[20]	İletişim frekansının azaltılması garantileri ile	Yerel güncelleme	[19]'a benzer şekilde, ancak IID olmayan ortamlar için yakınsama
[21]	Model güncellemeleri boyut azaltma	Sıkıştırma (niceleme)	Sunucu ve cihazlar arasında iletilen veri miktarını azaltmak için yerel model güncellemelerini sonlu sayıda bite sıkıştırın
[24]	Model güncellemeleri boyut azaltma	Sıkıştırma (ayrıklaştırma)	Bir model, önemsiz öğeleri sıfıra ayarlamak için ayrıştırma stratejisini günceller
[25]	Model güncellemeleri boyut azaltma	Sıkıştırma (ayrıklaştırma)	Model güncellemelerini Tekil Değer Ayrıştırma (SVD) alanına dönüştürün ve model güncellemelerini düşük boyutlu yapıya ayırın

3. Önbilgiler ve Problem Formülasyonu

Önerilen algoritmanın sisteme genel bakışı Şekil 1'de sunulmuştur. Önerilen algoritmanın ayrıntılarına girmeden önce, bu makale yerel güncelleme ve parametre sıkıştırmanın etkisini daha iyi tanımlamak için yerel güncelleme ve parametre sıkıştırma katsayılarının federe öğrenmeyi nasıl etkilediğine dair matematiksel bir analiz sunmaktadır. Ek olarak, problemimizi teorik olarak formüle ediyor ve yerel güncelleme ve parametre sıkıştırma dahil olmak üzere iki değişkeni birlikte dikkate alan federe öğrenmenin hata üst sınırını türetiyoruz.



Şekil 1. FedEAFO şemasının sisteme genel bakışı.

3.1. Federe Öğrenme

Kaynak kısıtlı IoT ortamlarında toplam N IoT cihazından oluşan bir federe öğrenme sistemi düşünüyoruz. Her cihaz kendi yerel veri kümesi üzerinde model eğitimi gerçekleştirir. Cihazlar aşağıdaki optimizasyon problemini ortaklaşa çözmeyi amaçlamaktadır.

$$\begin{aligned} \min_w F(w) &= \frac{1}{N} \sum_{n=1}^N F^n(w) \quad (1) \\ &\Leftrightarrow \sum_{n=1}^N F^n(w) \end{aligned}$$

Burada w global model ağırlıklarını, F^n n 'inci cihazın ağırlığını ve F^n n 'inci cihazın yerel hedefini temsil eder. Denklem (1), cihazlar ve merkezi sunucu arasında model parametrelerinin iteratif değişimi ile optimize edilebilir. Özellikle, t 'inci turda, n 'inci cihaz w^n 'nin yerel modeli şu şekilde gösterilebilir

$$w_{t,i+1}^n \leftarrow w_{t,i}^n - \eta_{t,i} \nabla F^n(w_{t,i}^n, \phi_{t,i}^n) \quad (2)$$

Burada $\phi_{t,i}^n$ veri örneklerine karşılık gelir, $\eta_{t,i}$ öğrenme oranını temsil eder ve i n 'inci yerel güncellemeler.

IoT'nin pratik uygulamalarında, farklı IoT cihazları kullanılacak ve farklı türde veriler toplanacaktır. FL, farklı veri türlerini işlemek için çok uygundur. IoT senaryolarında, farklı sensörlere ve işlemlere sahip çeşitli cihazlar veri üretir ve bu veriler oldukça heterojen olabilir. FL, bu cihazların ham verilerini merkezi bir sunucuya paylaşmadan ortak bir modeli işbirliği içinde öğrenmelerine olanak tanıyan dağıtılmış bir öğrenme çerçevesi sağlar. FL'nin IoT uygulamalarında farklı veri türlerini nasıl ele aldığı aşağıda açıklanmıştır: (i) Uyarlanabilir Hiperparametre Ayarlama: Bazı FL algoritmaları, modelin hiperparametrelerini her bir cihazın veri özelliklerine göre ayarlamasına olanak tanıyan uyarlanabilir optimizasyon teknikleri içerir. Bu uyarlanabilirlik, farklı veri dağılımlarına sahip cihazların verimli bir şekilde kullanılmasına yardımcı olur. (ii) Merkezi Olmayan Veri İşleme: FL'de her IoT cihazı kendi verileri üzerindeki kontrolü yerel olarak elinde tutar. Cihazlar, seçilen model mimarisiyle tutarlılık ve uyumluluk sağlamak için verilerini önceden işler ve dönüştürür. Bu merkezsizleştirme, cihazların kendi özel veri türlerini ve formatlarını bağımsız olarak işlemesine olanak tanır.

(iii) Model Kişiselleştirme: FL, global modelin farklı cihazların veri özelliklerine uyarlanabildiği model kişiselleştirmeyi destekler. Cihazlar benzersiz veri dağılımlarına sahip olduğunda, kendi yerel verilerini kullanarak paylaşılan modeli kişiselleştirebilirler. Federated Averaging gibi model birleştirme mekanizmaları, kişiselleştirilmiş modellerin genel performansı iyileştirmek için birleştirilmesini sağlar.

n 'inci cihaz Denklem (2)'ye dayalı olarak yerel güncellemeleri gerçekleştirir gerçekleştirmez, yerel modellerin $l(w^n)$ birleştirilmiş güncellemeleri şu şekilde elde edilebilir

$$l(w^t) \leftarrow \sum_{i=1}^B l(w_{t,i}^n, \phi_{t,i}^n) \quad (3)$$

burada $l(w_{t,i}^n, \phi_{t,i}^n) = \nabla F^n(w_{t,i}^n, \phi_{t,i}^n)$, I_t ardışık yerel güncellemelerin sayısını temsil eder,

FL'nin hesaplama ve iletişimi arasındaki dengeyi sağlamak için ayarlanabilir. Cihazlar daha sonra yerel olarak birleştirilmiş güncellemelerini $l(w^t)$ ile $\hat{l}(w^t)$ olarak sıkıştırır.

ϵ_t olarak gösterilen seyreklik bütçesi, FL'nin iletişim ve model hassasiyeti arasındaki ödünleşimleri dengelemek için ayarlanabilir. Sunucu, aşağıdaki şekilde verilen sıkıştırılmış bir küresel model elde etmek için cihazlardan gelen tüm sıkıştırılmış yerel olarak toplanmış güncellemeleri toplar

$$\hat{l}(w_t) = \frac{1}{N} \sum_{n=1}^N l(w_t^n) \quad (4)$$

En son global model w_{t+1} stokastik gradyan iniş algoritması aracılığıyla şu şekilde güncellenebilir

$$w_{t+1} \leftarrow w_t - \eta l^*(w)_t \quad (5)$$

Merkezi sunucu güncellenmiş global modeli w_{t+1} bir sonraki eğitim turu için federe öğrenmeye dahil olan cihazlara iletir. Cihazların yerel modeli, alınan en son küresel ağırlıklar aracılığıyla güncellenebilir. Bu, hem yerel güncelleme hem de parametre sıkıştırmanın birlikte ele alındığı tam bir federe öğrenme sürecidir.

Sistemimizin eğitim sürecinin gecikmesinin analizi için, cihazların merkezi baz istasyonlarının kapsama alanı içinde eşit olarak dağıtıldığı dairesel bir küçük hücreli ağ düşünüyoruz. d^n baz istasyonu ile n cihazı arasındaki mesafeyi ve $L_{path}(d)$ yol kaybını gösterebilir. P^n ve P sırasıyla n cihazının ve baz istasyonunun iletim gücünü gösterebilir. Yukarı bağlantı kanalının ortogonal frekans bölme çoklu erişim (OFDMA) teknolojisini benimsediğini varsayalım. Cihazlar, yukarı bağlantı kanalında Slotted ALOHA protokolü kuralı altında yerel modelleri yüklemek için bir kaynak bloğu için rekabet eder. Baz istasyonu ise global modelleri aşağı bağlantı zaman dilimlerinde iletmek için tüm bant genişliğini kullanır. B sistemimizin toplam bant genişliğini ve B_U kaynak bloğu bant genişliğini gösterebilir. Böylece, yukarı bağlantı ve aşağı bağlantının sinyal-gürültü oranı kanalı $\rho^n = P^n - L_{path}(d^n) - N_0 B_U$ ve $\rho^n = P - L_{path}(d^n) - N_0 B$ ile gösterilebilir, Burada N_0 gürültünün spektral güç yoğunluğudur.

Yukarıdaki tanıma dayanarak, yerel model için iletişim gecikmesi T_n^U yüklemesi ve global model indirmesi $T_n^{n_D}$ -inci cihaz şu şekilde verilebilir

$$T_n^U = \frac{H}{B_U \log_2(1 + \rho^n_{D,U})}, \quad T_n^{n_D} = \frac{H}{B \log_2(1 + \rho^n_D)} \quad (6)$$

Burada H , modelin iletimi için toplam niceleme bitlerini gösterir.

n 'inci cihazın yerel modelini eğitmek ve güncellemek için hesaplama gecikmesi olarak hesaplanabilir

$$T_{C,t}^n = \frac{ECD_n}{f^n}, \quad T_{L,t}^n = \frac{Q}{f^n} \quad (7)$$

Burada E yerel güncelleme sayısını, C ve Q sırasıyla bir veri örneğini eğitmek ve yerel modeli güncellemek için gereken kayan nokta işlemlerinin sayısını ve f^n ise n 'inci cihazın CPU kapasitesini gösterir.

Böylece, t 'inci turdaki FL eğitim sürecinin uçtan uca gecikmesi şu şekilde verilebilir

$$T_t = \max_{U,t} \{T_{U,t}^n + T_{D,t}^n\} + \max_{C,t} \{T_{C,t}^n + T_{L,t}^n\} \quad (8)$$

3.2. Problem Formülasyonu

Amacımız, I_t ve ε_t dahil olmak üzere iki değişkeni birlikte ele almak ve federe öğrenmeyi hızlandırmak için bunları uyarlamalı olarak ayarlamaktır. Teorik olarak problem, farklı eğitim turlarında I_t ve ε_t için belirli bir sürede federe öğrenmenin hatasını en aza indiren en uygun çözümü bulmaktır. Problem matematiksel olarak şu şekilde formüle edilebilir

$$\begin{aligned} \min_{\{I_t\}_{t \in \mathcal{T}}, \{\varepsilon_t\}_{t \in \mathcal{T}}} & E_{\{\Phi_t\}_{t \in \mathcal{T}}} [min F(w_t)] \\ \text{s.t.} & \sum_{t=1}^{U_t} T_{comp} + T_{comm} = T \end{aligned} \quad (9)$$

Burada $F(w_t)$, Denklem (1)'de tanımlanan küresel öğrenme hedefine karşılık gelir. T , verilen zaman kısıtlamasını temsil eder. Denklem (9)'daki problemi çözmek için anahtar, I_t ve ε_t 'nin karşılıklı bağımlılığını tanımlayan federe öğrenme hatasının ifadesini türetmektir. Ancak, böyle bir ifadeyi elde etmek neredeyse imkansızdır. Ayrıca, FL'de I_t ve ε_t 'yi birlikte dikkate alan hata üst sınırını bulmak da zordur.

3.3. Öğrenme Hatası Üst Sınırı

Yerel güncelleme ve parametre sıkıştırma olmak üzere iki değişkeni birlikte dikkate alan federe öğrenmenin hata üst sınırını türetmek için, bu alt bölümde ilk olarak I_t cinsinden hata üst sınırı türetilmektedir. Ayrıca, I_t tarafından tanımlanan yerel güncelleme ve ε_t tarafından tanımlanan parametre sıkıştırma dikkate alınarak hata üst sınırı türetilmektedir.

Üst hata analizini sunmak için öncelikle aşağıdaki 1-3. Varsayımları yapıyoruz [19]'dan esinlenerek parametre sıkıştırması olmadan sınırlandırılmıştır.

- (c) Küresel kayıp fonksiyonu türevlenebilir ve L-düzgündür: $\|\nabla F(\mathbf{V}) - \nabla F(\mathbf{W})\| \leq L\|\mathbf{V} - \mathbf{W}\|$ ve bir F alt sınırı vardır_{inf}.
- (d) Mini partideki global ağırlık varyansı şu şekilde sınırlandırılmıştır: $E_{\{ \phi \}} \|\nabla l(w_t) - \nabla F(w_t)\|^2 \leq \lambda \|\nabla F(w_n)\|^2 + \delta$. δ , $\nabla l(w_n)$ arasındaki varyansa karşılık gelir. λ ve δ , mini parti boyutuyla ters orantılı sabitlerdir.
- (e) SGD, FGD'nin yansız tahmincisidir: $E_{\{ \phi \}} [\nabla l(w_t)] = \nabla F(w_t)$.

Theorem 1. Varsayımlar 1-3 geçerli olsun. $\eta L + \eta L^2 I_t (I_t - 1) \leq 1$ koşulunu sağlayan öğrenme oranını seçin. Böylece, Denklem (9)'da verilen T süresi içinde n turdan sonraki öğrenme hatası şu şekilde sınırlanır:

$$\frac{2[F(w_t) - F_{\inf}]}{\eta T} T_{\text{comp}} + \frac{T_{\text{comm}}}{I_t} + \frac{\eta L \delta}{\eta N L^2} + \eta N L^2 \delta (I_t - 1) \quad (10)$$

Kanıt. [19]'daki Ek'e bakınız.

Yukarıdaki Teorem 1, öğrenme hatasını en aza indirmek için hesaplama ve iletişim arasındaki değiş tokuşu açıklamaktadır. Denklem (10), yerel güncelleme katsayısının I_t hata üst sınırı ifadesinin pay ve paydasında olduğunu göstermektedir, bu da I_t değerlerinin çok küçük veya çok büyük olduğu durumlarda hata üst sınırının azalacağı anlamına gelmektedir. Bu nedenle, dengenin sağlanması gerekmektedir. Ayrıca, Denklem (10)'da yer alan dinamik olarak değişen kayıp fonksiyonu $F(w_k)$ nedeniyle dengenin çeşitli federe öğrenme turlarında dinamik olarak ayarlanması gerekir.

Yerel güncellenmenin yanı sıra, yerel olarak toplanmış ağırlıklara sıkıştırma getiren parametre sıkıştırma yaklaşımı, hata üst sınırının analizini iki açıdan karmaşıklatacaktır: (i) T_{comm} parametre sıkıştırma katsayısı ε_t tarafından etkilenecektir [26,27]. (ii) Denklem (10)'daki δ varyansı ε_t parametre sıkıştırma katsayısına bağlı olacaktır [28,29].

Daha önce de belirtildiği gibi, parametre sıkıştırmada sıkıştırılmış parametreler temel bileşenler tarafından yaklaşırlır. Seyreklik nedeniyle, ham parametrelerin yaklaştırılması açısından bazı bileşenler diğerlerinden daha önemlidir. Dolayısıyla sorun, δ varyansını en aza indirmek için temel bileşenleri tarafsız bir şekilde seçmektir. n'inci cihazdan gelen yerel olarak toplanmış ağırlıklar şu şekilde yeniden yazılabilir:

$$l(w_t^n) = \sum_{k=1}^K d^k(w_t^n) \alpha^k(w_t^n) \quad (11)$$

where K is the number of basic components, $\alpha^k(w_t^n)$ corresponds to the k -th basic component, and $d^k(w_t^n)$ is the corresponding weight. Our analysis is based on the fact that a matrix can be denoted as a combination of basic matrices, which is the atomic decomposition for sparse representation in compressed sensing. Thus, our analysis can be extended to nearly all unbiased compressions. For example, TernGrad [21] and QSGD [22] are special cases of Equation (8). Additionally, sparsification algorithms including ATOMO [25] also follow

Denklem (8). Formülasyon, $d^k(w^n)$ seçimine ilişkin bir sorun ortaya koymaktadır. Tarafsız seçim gereksinimini karşılamak için, [25]'ten esinlenerek aşağıdaki gibi bir tahmin edici öneriyoruz:

$$\hat{l}(w_t^n) = \sum_{k=1}^K \frac{d^k(w_t^n) e^k(w_t^n)}{p^k(w_t)} \alpha(w_t) \quad (12)$$

Burada $p^k(w_t^n) \in (0, 1]$ ve $e^k(w_t^n)$ Bernoulli dağılımına uyar. İki önemli sonuç sunuyoruz

Lemma 1 ve 2 aracılığıyla tahmin edici için özellikler.

Lemma 1. Denklem (12) ile verilen tahmin edicinin varyansı şu şekilde gösterilebilir:

$$E_{\{e(w_t^n)\}} \left\| \Gamma(w_t^n) - I(w_t^n) \right\|^2 = \sum_{k=1}^K d^k(w_t^n) \frac{1}{p^k(w_t)} - 1.$$

Kanıt. Denklem (9)'daki tahmin edicinin varyansı aşağıdaki gibi tanımlanır:

$$\begin{aligned} E_{\{e(w_t^n)\}} \left\| \Gamma(w_t^n) - I(w_t^n) \right\|^2 &= E_{\{e(w_t^n)\}} \sum_{k=1}^K \frac{d^k(w_t^n) e^k(w_t^n)}{p^k(w_t)} \frac{e^k(w_t^n) p^k(w_t)}{p(w_t)} \\ &= \sum_{k=1}^K d^k(w_t^n) \frac{e^k(w_t^n) p^k(w_t)}{p(w_t)} \\ &= \sum_{k=1}^K d^k(w_t^n) \frac{e^k(w_t^n) p^k(w_t)}{p(w_t)} \\ &+ \sum_{x,y \neq y} \frac{d^x(w_t^n) d^y(w_t^n)}{p(w_t)} \frac{e^x(w_t^n) p^x(w_t)}{p(w_t)} \frac{e^y(w_t^n) p^y(w_t)}{p(w_t)} \\ &\times E_{\{e(w_t^n)\}} \frac{e^x(w_t^n) p^x(w_t)}{p(w_t)} \frac{e^y(w_t^n) p^y(w_t)}{p(w_t)} \end{aligned} \quad (13)$$

nerede $E_{\{e(w_t^n)\}} \frac{e^k(w_t^n) p^k(w_t)}{p(w_t)}$ olarak gösterilebilir:

$$\begin{aligned} E_{\{e(w_t^n)\}} \frac{e^k(w_t^n) p^k(w_t)}{p(w_t)} &= \frac{1}{p(w_t)} \times \frac{e^k(w_t^n) p^k(w_t)}{p(w_t)} \\ &= \frac{1}{p(w_t)} - 1 \end{aligned} \quad (14)$$

ve $E_{\{e(w_t^n)\}} \frac{e^x(w_t^n) p^x(w_t)}{p(w_t)}$ ile elde edilebilir:

$$\begin{aligned} E_{\{e(w_t^n)\}} \frac{e^x(w_t^n) p^x(w_t)}{p(w_t)} &= (1 - p^x(w_t^n)) \frac{e^x(w_t^n) p^x(w_t)}{p(w_t)} + p^x(w_t^n) \frac{e^x(w_t^n) p^x(w_t)}{p(w_t)} \\ &= 0 \end{aligned} \quad (15)$$

ve aynı şekilde, $E_{\{e(w_t^n)\}} \frac{e^y(w_t^n) p^y(w_t)}{p(w_t)} = 0$.

Böylece, Denklem (13)-(15) ve $\|w^n\|_2 = 1$, tahmin edicinin varyansı Denklem (12)'deki değerler aşağıdaki gibi elde edilebilir:

$$E_{\{e(w_t^n)\}} \|\hat{w}^n - I(w_t^n)\|_2^2 = \sum_{k=1}^K d_k(w_t^n)^2 \frac{1}{p_k(w_t^n)} \quad (16)$$

Lemma 2. Denklem (12)'deki tahmin edici yansızdır, bu da şu anlama gelir: $E_{\{e(w_t^n)\}} \|\hat{w}^n - I(w_t^n)\|_2^2 = I(w_t^n)$.

Kanıt. Bu basitçe beklentinin tanımı ile kanıtlanabilir.

Varyansı minimize etmek için optimizasyon problemini formüle ediyoruz. Varyansı en aza indirmeye çalışmamızın nedeni, varyans azaldığında sıkıştırılmış parametrelerin orijinal parametrelere daha yakın olmasıdır. Varyansı minimize etme problemi şu şekilde verilebilir:

$$\min \sum_{k=1}^K d_k(w_t^n)^2 \quad \text{s.t. } 0 < p(w_t) \leq 1 \quad \text{ve} \quad \sum_{k=1}^K p(w_t) = \varepsilon_t \quad (17)$$

Before solving the optimization problem, we first provide the following Definition 1 of ε_t -balancedness:

Tanım 1. $I(w_t^n) = \sum_{k=1}^K d_k(w_t^n) \alpha^k(w_t^n)$ ε -dengesiz ise $d^k(w_t^n) \varepsilon_t > d_{w_t}^n \cdot \varepsilon_t$ dengeli $I(w_t^n)$ 'nin hiçbir elemanının ε_t -dengesiz olmadığı duruma karşılık gelir.

Böylece, optimizasyon probleminin çözümü için teorem şu şekilde verilebilir:

Teorem 2. $I(w_t^n) \varepsilon_t$ -dengeli olduğunda, yukarıda bahsedilen problemin çözümü şöyle olabilir tarafından elde edildi:

$$p(w_t) = \frac{d(w_t^n)_{\varepsilon t}}{\|d(w_t^n)\|_1} \quad (18)$$

İspat. Teorem 2, Lagrangian çarpımı [30] ile kanıtlanabilir.

Lemma 3. Sıkıştırılmamış parametreler $I(w_t^n)$ ile sıkıştırılmış parametreler arasındaki fark $\hat{I}(w_t^n)$ ve tarafından verilebilir:

$$E_{\{e(w_t^n)\}} \|\hat{I}(w_t^n) - I(w_t^n)\|_2^2 = \frac{\delta_{1,n}^2}{\varepsilon_t^2} + \quad (19)$$

burada $\delta_{1,n} = \sum_{k=1}^K d_k(w_t^n) d(w_t^n)$ ve $\delta_{2,n}^2 = \sum_{k=1}^K d_k(w_t^n)^2$.

İspat. Lemma 1 ve Teorem 2'ye dayanarak, $E_{\{e(w_t^n)\}} \|\hat{I}(w_t^n) - I(w_t^n)\|_2^2$ ile gösterilebilir:

$$\begin{aligned} E_{\{e(w_t^n)\}} \|\hat{I}(w_t^n) - I(w_t^n)\|_2^2 &= \sum_{k=1}^K d_k(w_t^n)^2 \frac{1}{p_k(w_t^n)} - 1 \\ &= \frac{1}{\varepsilon_t^2} \sum_{k=1}^K d_k(w_t^n)^2 \frac{1}{p_k(w_t^n)} - 1 \end{aligned} \quad (20)$$

$$\text{burada } \delta_{1,n}^t = \sum_{k=1}^K d_k(w_t)^n \|d(w_t^n)\|_{t-1} \quad \text{ve} \quad \delta_{2,n}^t = - \sum_{k=1}^K d_k(w_t)^{n^2}.$$

Tahmin edicinin varyansının olasılığını türettikten sonra, sıkıştırmanın varyans δ ve iletişim süresi T üzerindeki etkilerine erişebiliriz_{comm}. Yüklenecek model parametrelerinin Y olarak gösterildiğini varsayalım. Cihazlar orijinal parametreleri göndermek yerine, aşağıdaki gibi gösterilen sıkıştırılmış yaklaşık parametreleri yükleyebilir

\hat{Y} seyreklik temsili ile sadece ε_t sayıda temel bileşeni yükler.

Böylece, T_{comm} iletişim süresi $T_{comm} = \gamma \varepsilon_t$ olarak yeniden yazılabilir, burada γ her bir cihazın iletişim süresine karşılık gelir. δ varyansına gelince, sıkıştırılmış yerel olarak toplanmış parametrelerin varyansına ilişkin teorem Teorem 2'ye dayanmaktadır ve şu şekilde verilebilir:

Teorem 3. Sıkıştırılmış yerel olarak toplanmış parametrelerin varyansı için sınır aşağıdaki gibidir:

$$E_{\{\phi_{t,i}\}, \{e_{wt}^k\}} \left\| \hat{I}(w_t) - \nabla F(w) \right\|_{t-1}^2 \leq \lambda \left\| \nabla F(w) \right\|_{t-1}^2 + \frac{\delta_1}{\varepsilon_t} + \delta_2 \quad (21)$$

Burada λ , δ_1 , ve δ_2 mini partinin boyutuyla ters orantılı sabitlerdir.

Kanıt. $E_{\{\phi_{t,i}\}, \{e_{wt}^k\}} \left\| \hat{I}(w_t) - \nabla F(w) \right\|_{t-1}^2$ olarak tanımlanabilir:

$$\begin{aligned} & E_{\{\phi_{t,i}\}, \{e_{wt}^k\}} \left\| \hat{I}(w_t) - \nabla F(w) \right\|_{t-1}^2 \\ &= E_{\{\phi_{t,i}\}, \{e_{wt}^k\}} \left\| \hat{I}(w_t) - I(w_t) + I(w_t) - \nabla F(w_t) \right\|_{t-1}^2 \\ &\leq E_{\{\phi_{t,i}\}, \{e_{wt}^k\}} \left\| \hat{I}(w_t) - I(w_t) \right\|_{t-1}^2 + E_{\{\phi_{t,i}\}, \{e_{wt}^k\}} \left\| I(w_t) - \nabla F(w_t) \right\|_{t-1}^2 \end{aligned} \quad (22)$$

Denklem (22)'deki ilk terim için, $E_{\{\phi_{t,i}\}, \{e_{wt}^k\}} \left\| \hat{I}(w_t) - I(w_t) \right\|_{t-1}^2$ ile elde edilebilir:

$$\begin{aligned} & E_{\{\phi_{t,i}\}, \{e_{wt}^k\}} \left\| \hat{I}(w_t) - I(w_t) \right\|_{t-1}^2 \\ &\leq \frac{1}{N} \sum_{n=1}^N d_k(w_t)^2 \frac{1}{\rho_k(w_t^n)} - 1 = \frac{1}{N} \sum_{n=1}^N \frac{\delta_{1,n}}{\varepsilon_t} + \delta_{2,n} \end{aligned} \quad (23)$$

burada $E_{\{\phi_{t,i}\}, \{e_{wt}^k\}} \left\| \hat{I}(w_t) - I(w_t) \right\|_{t-1}^2 = \sum_{k=1}^K d_k(w_t)^2 \frac{1}{\rho_k(w_t^n)} - 1$ Lemma 1'den ve

$$\sum_{k=1}^K d_k(w_t)^2 \frac{1}{\rho_k(w_t^n)} - 1 = \frac{\delta_{1,n}}{\varepsilon_t} + \delta_{2,n} \quad \text{Lemma 3'ten alınmıştır.}$$

Denklem (22)'deki ikinci terim için, $E_{\{\phi_{t,i}\}, \{e_{wt}^k\}} \left\| I(w_t) - \nabla F(w_t) \right\|_{t-1}^2$ şu şekilde elde

edilebilir

$\lambda \left\| \nabla F(w) \right\|_{t-1}^2 + \delta$ [19] tarafından takip $E_{\{\phi_{t,i}\}, \{e_{wt}^k\}} \left\| \hat{I}(w_t) - \nabla F(w) \right\|_{t-1}^2$ de-

Böylece, Denklem (22)-(23)'e göre, $E_{\{\phi_{t,i}\}, \{e_{wt}^k\}} \left\| \hat{I}(w_t) - \nabla F(w) \right\|_{t-1}^2$ olabilir.

tarafından yazıldı:

$$\begin{aligned} & E_{\{\phi_{t,i}\}, \{e_{wt}^k\}} \left\| \hat{I}(w_t) - I(w_t) \right\|_{t-1}^2 \\ &\leq \frac{1}{N} \sum_{n=1}^N d_k(w_t)^2 \frac{1}{\rho_k(w_t^n)} - 1 = \frac{1}{N} \sum_{n=1}^N \frac{\delta_{1,n}}{\varepsilon_t} + \delta_{2,n} \end{aligned} \quad (24)$$

burada $\delta = \max_{k=1}^N \delta_k$ ve $\delta = \max_{k=1}^N \delta_k + \delta$.

Elde edilen yeni iletişim süresi ve varyans ile Teorem 1'deki Varsayım 2, Denklem (25) aracılığıyla yeniden yazılabilir. Parametre ile hata üst sınırını türetmek için

Sıkıştırma, Teorem 1'i yeni varyans ve iletişim süresi ile uyarlıyoruz, bu da parametre sıkıştırmanın iletişim süresi ve varyans üzerindeki etkisinin federe öğrenmenin yakınsaması üzerindeki etkisini dikkate alıyor. Güncellenmiş Teorem aşağıdaki gibidir:

Theorem 4. *The bound error upper bound with parameter compression is as follows:*

$$u_t(I_t, \varepsilon_t) = \frac{2[F(w_t) - F]_{\inf}}{\eta T} T_{comp} + \frac{\gamma \varepsilon_t}{I_t} + \frac{\eta L (\frac{\delta_1}{2} + \delta)}{N} + \eta^2 L^2 \frac{\delta_1}{\varepsilon_t} + \delta_2)(I_t - 1) \quad (25)$$

Burada $u_t(I_t, \varepsilon_t)$ yerel güncelleme ve parametre sıkıştırma dikkate alan hata üst sınırına karşılık gelir.

Kanıt. 19]'daki Ek'e bakınız, ancak yeni iletişim süresi T_{comp} yerine $\gamma \varepsilon_t$ ve δ varyansı $1 + \delta_2$ ile değiştirilir.

$u_t(I_t, \varepsilon_t)$, I_t ve ε_t tarafından belirlenen iletişim ve hesaplama/hassasiyet arasındaki değiş tokuş dinamiklerini gösterir. $u_t(I_t, \varepsilon_t)$ ifadesindeki ilk terim, I_t payda olduğu için $u_t(I_t, \varepsilon_t)$ değerinin I_t arttıkça azaldığını ve ε_t payda olduğu için $u_t(I_t, \varepsilon_t)$ değerinin ε_t azaldıkça azaldığını göstermektedir. Dolayısıyla, Denklem (22)'deki ilk terimin analizine dayanarak I_t ve ε_t 'nin azaltılması gerekir. Ancak, Denklem (22)'nin üçüncü terimi I_t 'nin küçük kalmasını gerektirir çünkü I_t paydadır. Denklem (22)'nin ikinci ve üçüncü terimlerinin büyük kalması gerekir çünkü ε_t paydadır. Yukarıdaki analiz, $I_t = 1$ veya $I_t \gg 1$ seçeneklerinden ilkinin gereksiz iletişim ek yüklerine yol açması, ikincisinin ise daha az iletişimden kaynaklanan yerel modeller arasındaki büyük tutarsızlıklar nedeniyle yakınsamanın uzamasından muzdarip olması nedeniyle optimal bir seçim olmadığını göstermektedir. Hem $\varepsilon_t = 1$ hem de $\varepsilon_t \gg 1$ optimum seçimler değildir, çünkü ilki uzayan yakınsamaya neden olan kesin olmayan model parametreleri gönderirken, ikincisi sık iletişimle sonuçlanır. Bu nedenle, bu makale iletişim ve hesaplama/hassasiyet arasındaki dengeleri ayarlamak için en uygun dengeyi bulmayı amaçlamaktadır.

4. Önerilen FedEAFO Algoritması

Yukarıdaki teorik analizler, hata üst sınırının I_t ve ε_t tarafından yönetildiğini göstermektedir. Bu makale, yerel güncelleme ve parametre sıkıştırma dahil olmak üzere iki değişkeni birlikte dikkate alarak öğrenme hatasını en aza indiren verimli bir uyarlanabilir federe optimizasyon (FedEAFO) algoritması önermektedir. FedEAFO, iletişim ve hassasiyet/hesaplama arasındaki değiş tokuşları uyarlamalı olarak ayarlar ve dengeler. Şekil 1, FedEAFO şemasına genel bir bakış sunmaktadır. Matematiksel olarak FedEAFO, Denklem (25)'teki hata üst sınırını en aza indirmek için en uygun dengeyi bulur, bu da şu şekilde gösterilebilir:

$$I_t^*, \varepsilon_t^* = \underset{I_t, \varepsilon_t}{\operatorname{argmin}} u_t(I_t, \varepsilon_t) \quad (26)$$

Bu makale, $u_t(I_t, \varepsilon_t)$ 'nin dışbükeyliğinin ispatı için teorik analiz sağlamak üzere Teorem 5'i ve ardından Denklem (26) problemini çözmek için en uygun çözümleri bulan Teorem 6'yı sunmaktadır.

Theorem 5. L , T ve η burada tanımlanmış olsun. Varsayımları seçin (4) $I_t \geq 2$, (5) $\eta^5 \approx 0$, (6) $L^4 T \delta_1 / 2\alpha(F(w_t) - F_{\inf})\varepsilon_t^4 \leq \infty$, (7) $2\eta^2 L T \delta_1 I_t \geq \alpha N \varepsilon_t^2 (F(w_t) - F_{\inf})$, dolayısıyla $u_t(I_t, \varepsilon_t)$ dışbükeydir.

Kanıt. $u_t(I_t, \varepsilon_t)$ Hessian matrisinin pozitif yarı sonsuz olması $u_t(I_t, \varepsilon_t)$ koşuludur. konveks olması gerekir. $u_t(I_t, \varepsilon_t)$ Hessian matrisi aşağıdaki gibi türetilir:

$$H(u_t(I_t, \varepsilon_t)) = \begin{bmatrix} \frac{2\eta\alpha\varepsilon_t}{I_t^3} & -\frac{\eta L}{I_t^2} & -\frac{\eta^2 L^2}{\varepsilon_t^2} \\ -\frac{\eta L}{I_t^2} & -\frac{\eta^2 L^2}{\varepsilon_t^2} & 2P(I_t - 1) \end{bmatrix} \quad (27)$$

burada $X = \frac{2[F(w_0) - F_{\inf}]}{\eta^2}$, $Z = \frac{\eta L}{N}$ ve $P = \eta^2 L^2$. Pozitif diyagonal elemanlar ve determinant, Hessian matrisinin pozitif yarı sonlu olması koşuludur ve bu da $u_t(I_t, \varepsilon_t)$ 'nin dışbükeyliğinin kanıtını sağlayabilir. Açıkçası, köşegen elemanlar ve determinant Varsayım 4-7'ye göre pozitiftir.

Teorem 6. Hata üst sınırını en aza indirmek için en uygun çözümler $u_t(I_t, \varepsilon_t)$ ile verilebilir:

$$I_t = \frac{2\alpha[F(w_t) - F_{\inf}]\varepsilon_t^2}{\eta^3 L^2 (\delta_1 + \delta_2 \varepsilon_t)}, \quad \varepsilon_t = \frac{\delta_1 \eta^2 L T (1 - \eta L (I_t - 1)) I_t}{2\alpha[F(w_t) - F_{\inf}]} \quad (28)$$

Varsayım 8 ($\delta_1 \ll \delta_2 \varepsilon_t$), Varsayım 9 ($\eta L (I_t - 1) \ll 1$) ve Varsayım 10 ($F_{\inf} = 0$) ile I_t ve ε_t yaklaşık olarak hesaplanabilir:

$$\frac{I_{t+1}}{I_t} \approx \frac{F(w_{t+1})}{F(w_t)} \frac{\varepsilon_{t+1}}{\varepsilon_t} \quad (29)$$

$$\frac{\varepsilon_{t+1}}{\varepsilon_t} \approx \frac{F(w_t)}{F(w_{t+1})} \frac{I_{t+1}}{I_t} \quad (30)$$

İspat. Bu, Varsayım 8-10'un benimsenmesi ve $u_t(I_t, \varepsilon_t)$ kısmi türevlerinin 0 olarak ayarlanmasıyla kanıtlanabilir.

Denklem (29) ve (30)'da I_t ve ε_t değerleri birbirine bağlıdır ve Denklem (29), Denklem (30)'da aşağıdaki şekilde değiştirilerek ayrıştırılabilir:

$$\frac{I_{t+1}}{I_t} = \frac{F(w_{t+1})}{F(w_t)}, \quad \frac{\varepsilon_{t+1}}{\varepsilon_t} = \frac{F(w_t)}{F(w_{t+1})} \quad (31)$$

$I_0, \varepsilon_0, F(w_0)$ başlangıç değerleri ile Denklem (32) şu şekilde yeniden yazılabilir:

$$I_t = \frac{F(w_t)}{F(w_0)}, \quad \varepsilon_t = \frac{F(w_0)}{F(w_t)} \quad (32)$$

Denklem (32), federe öğrenmenin eğitim sürecinde kayıp değeri $F(w_t)$ azaldıkça, I_t 'nin azalması ve ε_t 'nin artması gerektiğini göstermektedir.

Algoritma 1, FedEAFO algoritmasını gerçekleştirirken eğitim sürecinin ayrıntılarını açıklamaktadır. Algoritma 1'in tam akışı aşağıdaki adımlar olarak tanımlanabilir: (i) Sunucu, hesaplanan yerel güncelleme katsayısını I_t , sıkıştırma katsayısını ε_t ve en son ağırlıkları seçilen cihazlara yayınlar. Yerel güncelleme katsayısı yerel eğitimdeki parametre hesaplamalarının sayısını belirlerken, parametre sıkıştırma katsayısı parametre sıkıştırma oranını belirler. (ii) Cihazlar, alınan yerel güncelleme katsayısına I_t dayalı olarak çoklu yerel eğitim gerçekleştirir ve sıkıştırılmış yerel olarak toplanmış model parametrelerini sıkıştırma katsayısı ε_t tarafından belirlenen belirli bir sıkıştırma oranında yükler. (iii) Sunucu, küresel modeli güncellemek için alınan tüm sıkıştırılmış model parametrelerini toplar. (iv) Sunucu, Teorem 6 aracılığıyla kaybın en son değerine göre yerel güncelleme katsayısı ve parametre sıkıştırma katsayısı dahil olmak üzere iki değişkeni birlikte optimize eder ve ayarlar. En son yerel güncelleme katsayısı, parametre sıkıştırma katsayısı ve en son ağırlıklar sunucu tarafından bir sonraki iterasyon için seçilen cihazlara yayınlanır.

Algoritma 1: Verimli Uyarlanabilir Federasyon

Optimizasyonu 1: **Sunucu Yürütür:**
2: **küresel model parametresinin başlatılması:** w_0
3: **for** round $t = 1$ to T **do**
4: Sunucu, Denklem (32) aracılığıyla I_t ve ϵ_t 'yi birlikte optimize eder: Sunucu I_t ve ϵ_t 'yi seçilen cihazlara yayınlar
6: **cihaz için** $n = 1$ ila N **do**
7: Cihazlar I_t ve ϵ alır,
8: yerel eğitim için $i = 1$ 'e **evet**
9: Cihazlar I^{wn} , ϕ^n hesaplar,
10: Cihazlar wn 'yi Denklem (2) aracılığıyla günceller
11: **icin son**
12: Cihazlar $I^{(wn)}$ 'yi Denklem (3) aracılığıyla hesaplar
13: Cihazlar $I^{(wn)}$ 'yi Denklem (12) ve (17) aracılığıyla sıkıştırır
14: Cihazlar sıkıştırılmış $I^*(w_t)$ 'yi sunucuya yükler
15: **icin son**
16: Sunucu Denklem (4) aracılığıyla sıkıştırılmış güncellemeleri toplar
17: Sunucu küresel modeli Denklem (5) aracılığıyla günceller
18: En son global model w_{t+1} cihazlara yayınlanır
19: **end for**

5. Deneysel Değerlendirme

Bu bölümde FedEAFO'nun federe öğrenmeyi hızlandırma performansını değerlendirmek için deneyler gerçekleştirilmekte ve Fashion- MNIST ve CIFAR-10 görüntü sınıflandırma görevleri üzerinde en son algoritmalarla karşılaştırılmaktadır. Deneyimizde, bu makale algoritmayı Nvidia TITAN RTX GPU'lu bir centos7 sunucusu aracılığıyla uygulamaktadır.

5.1. Deney Düzenliği

Eğitim Veri Kümeleri: 31,32]'yi takiben, bu makale Fashion-MNIST ve CIFAR-10'dan hem IID hem de IID olmayan veri kümeleri oluşturmaktadır. Fashion-MNIST veri kümesi, 70.000 28×28 görüntü içeren 10 moda öğesi kategorisinden oluşmaktadır. CIFAR-10 veri kümesi, uçaklar, kuşlar ve yarış arabaları gibi 10 sınıfta 32×32 renkli piksellere sahip 60.000 örnekten oluşmaktadır.

Uygulama ayarları: Eğitim ayarları için, bu makale yineleme süresini 200, yerel epoğu 10, yığın boyutunu 32 ve öğrenme oranını 0.01 olarak başlatmaktadır. Önerilen tasarım, Fashion-MNIST ve CIFAR-10 veri kümeleri üzerindeki görüntü sınıflandırma görevleri üzerinde incelenmiştir. Model mimarileri için, bu makale sırasıyla 32 ve 64 kanallı iki evrişimli katmana sahip Evrişimsel Sinir Ağı (CNN) ve Fashion-MNIST üzerinde tam bağlı ve softmax çıkış katmanını ve CIFAR-10 üzerinde hafif ağırlıklı bir ResNet18'i benimser.

Karşılaştırma temelleri: Bu makale, önerilen FedEAFO'yu son teknoloji yöntemlerin iki kategorisi ile karşılaştırmaktadır: (i) Model parametrelerinin sık iletimini azaltmak için cihazların hesaplama kapasitesinden tam olarak yararlanmayı amaçlayan yerel güncelleme. Bu makale, yerel güncelleme katsayılarını uyarlanabilir stratejilerle dinamik olarak ayarlayan ADACOMM'u [19] temel olarak ele almaktadır. (ii) Veri sıkıştırma teknikleri aracılığıyla iletilecek veri miktarını azaltan parametre sıkıştırma. Karşılaştırmalar için parametreleri sıkıştırmak amacıyla parametrelerin seyrekliğinden yararlanan ATOMO'yu [25] benimsiyoruz.

Değerlendirme ölçütleri: Bu makale, deneyim sonuçlarını doğru bir şekilde analiz etmek için farklı boyutları dikkate almaktadır: (1) Tur tamamlama süresi (verimlilik ölçütü). Bu makale, tur tamamlama süresini, bir federe öğrenme turunda harcanması gereken süre olarak tanımlamaktadır. Yineleme/iletişim turu yerine tur tamamlama süresini benimsememizin nedeni, farklı yineleme sürelerinin neredeyse aynı olduğu geleneksel makine öğreniminin aksine, yineleme turunun yakınsama hızını yansıtmaması ve farklı turların yineleme sürelerinin farklı olmasıdır. FL'nin yakınsama hızı için kilit faktörler tur tamamlama süresi ve yineleme turudur. Tur tamamlanma süresi, zamanla değişen üç gecikme bileşeni nedeniyle tur boyunca değişir: (i) Farklı aşağı bağlantı hızlarına sahip cihazlardan global model parametrelerinin indirilme gecikmeleri. (ii)

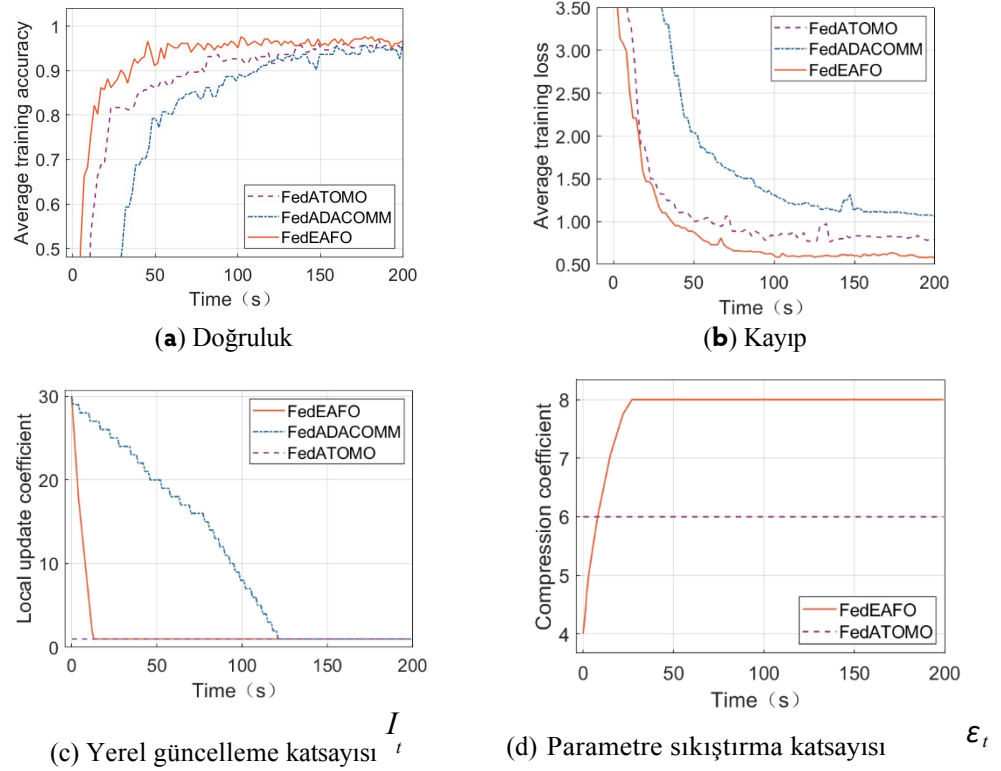
Farklı sıkıştırma hızlarına sahip cihazlardan gelen sıkıştırılmış yerel model parametrelerinin iletim gecikmeleri.

siyon oranları ve yukarı bağlantı hızları. (iii) Farklı yerel güncelleme katsayılarına ve donanım kısıtlamalarına sahip cihazlardan gelen yerel model güncellemesinin hesaplama gecikmeleri. (2) Küresel Öğrenme doğruluğu (fayda metriği). Bu makale, FedEAFO'nun FL'yi hızlandırmadaki etkinliğini değerlendirmek için küresel öğrenme doğruluğunu küresel modelin ortalama doğruluğu olarak ele almaktadır. (3) Trafik Tüketimi (iletişim verimliliği metriği). İletişim verimliliği, federe öğrenme süreci sırasında IoT cihazları ile merkezi sunucu arasında değiş tokuş edilen veri miktarını ölçmektedir. İletişim kaynaklarının genellikle sınırlı olduğu IoT bağlamında, bu metrik çok önemli hale gelmektedir. İletişim açısından daha verimli bir algoritma, iletilen veri miktarını azaltarak daha düşük bant genişliği tüketimine ve cihazlar için enerji harcamasına yol açar. Uyarlanabilir optimizasyon tekniğinin diğer performans unsurlarını korurken veya iyileştirirken iletişim verimliliğini nasıl etkilediğini değerlendirmek önemlidir.

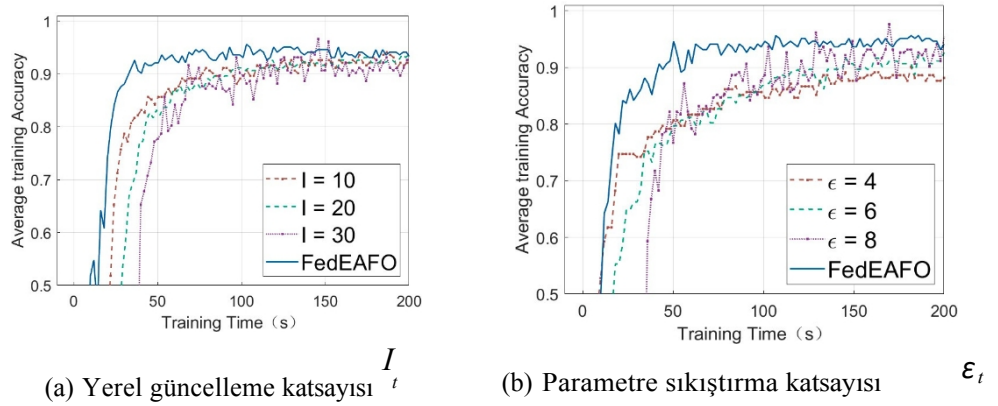
5.2. Sonuçlar ve Analiz

Bu makale, önerilen FedEAFO'yu yerel güncelleme kategorisindeki ADACOMM [19] ve parametre sıkıştırma kategorisindeki ATOMO [25] dahil olmak üzere iki son teknoloji algoritma kategorisiyle karşılaştırmaktadır. Şekil 2'deki karşılaştırmalı deney sonuçları, Fashion-MNIST veri kümesi üzerinde 32 cihaz için tur tamamlama süresi üzerinden doğruluk, kayıp, yerel güncelleme katsayısı I_t ve parametre sıkıştırma katsayısı ϵ_t değerlerini sunmaktadır. Şekil 2a, ATOMO ve ADACOMM ile karşılaştırıldığında önerilen FedEAFO'nun verimlilik ve yüksek doğruluk açısından üstünlüğünü göstermektedir, bu da tur tamamlama süresi boyunca daha yüksek doğruluklara daha hızlı ulaşmaktadır. Şekil 2b algoritmaların eğitim kaybını göstermektedir. Şekil 2c, eğitim süresi boyunca I_t değerlerini göstermektedir; burada ATOMO'nun yerel güncellemesi, birden fazla yerel güncelleme gerçekleştirmediği için bir olarak atanmıştır. Şekil 2d, ADACOMM hariç ϵ_t değerlerini göstermektedir çünkü parametre sıkıştırmasını benimsemekten ATAMO sabit bir ϵ_t değeri ile parametre sıkıştırmasını benimser. Önerilen EAFO'nun daha verimli olmasının nedeni, FedEAFO'nun federe öğrenme sırasında I_t ve ϵ_t değerlerini dinamik olarak ayarlamasıdır. I_t için, önerilen FedEAFO daha fazla yerel güncelleme yapılmasını sağlamak için yüksek değerlerle başlar, ancak daha düşük değerlerle ve yüksek doğrulukla sona erer. ϵ_t için, önerilen FedEAFO kaba parametreler nedeniyle düşük değerlerle başlar ve yine de düşük doğrulukla modelin doğruluğunu etkili bir şekilde geliştirmeye katkıda bulunabilir. Önerilen FedEAFO, Şekil 2d'de daha ince taneli parametreler ve yüksek doğruluk sağlayan yüksek değerlerle sona ermektedir. İletişim/hesaplama değiş tokuşu ve iletişim/hassasiyet değiş tokuşu iki keşif-kullanım değiş tokuşu olarak görülebilir. Yerel güncelleme katsayısının ve parametre sıkıştırma katsayısının her iki ayarı da ilgili keşif-kullanma ödünleşimini çözmek için kullanılabilir. Önerilen FedEAFO, büyük I_t ve küçük ϵ_t aracılığıyla yüksek keşif ile başlar ve FedEAFO, küçük I_t ve büyük ϵ_t aracılığıyla saf sömürü ile sona erer, bu da kararlı duruma ulaşıldığı anlamına gelir. Hem yukarı bağlantı hem de aşağı bağlantının veri hızları 100 Kbps olarak ayarlanmıştır. Veri kümesi IID'dir. $I_t \in \{1, \dots, 40\}$ ve $\epsilon_t \in [4, 8]$ olduğunu unutmayın.

İkinci simülasyon, önerdiğimiz FedEAFO algoritmasının performansının sabit yerel güncelleme katsayısı şemaları ve sabit parametre sıkıştırma katsayısı şemaları ile karşılaştırılmasını sunmaktadır. Şekil 3a, yerel güncelleme katsayısı I_t 'nin farklı değerleri altında zaman içindeki eğitim doğruluğunu göstermektedir, 10, 20 ve 30 olarak ayarlanmıştır. Şekil 3b, ϵ_t parametre sıkıştırma katsayısının 4, 6 ve 8 olarak ayarlanmış farklı değerleri altında zaman içindeki eğitim doğruluğunu göstermektedir. Şekil 3'te gösterildiği gibi, önerdiğimiz FedEAFO algoritması, sabit yerel güncelleme ve parametre sıkıştırma şemalarından daha hızlı ve daha düzgün yakınsamaktadır, bu da önerilen algoritmanın etkinliğini doğrulamaktadır.



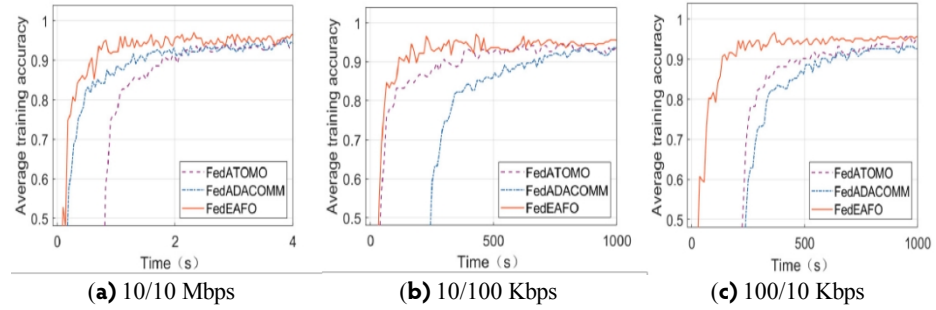
Şekil 2. Tur tamamlama süresi boyunca eğitim doğruluğu, eğitim kaybı, I_t ve ϵ_t değerleri.



Şekil 3. Tur tamamlama süresi boyunca yerel güncelleme katsayısı I_t ve parametre sıkıştırma katsayısı ϵ_t 'nin farklı değerleri ile eğitim doğruluğu değerleri.

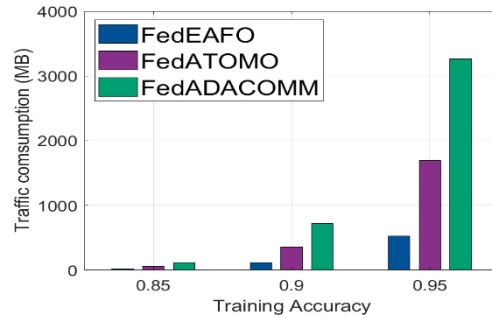
Bu makale, ADACOMM ve ATOMO ile karşılaştırmalı olarak önerilen FedEAFO'nun farklı yukarı bağlantı/aşağı bağlantı veri hızlarındaki performansını değerlendirmektedir. Şekil 4a-c'deki sonuçlar sırasıyla 10/10 Mbps, 10/100 Kbps ve 100/10 Kbps yukarı bağlantı/aşağı bağlantı veri hızları için doğruluğu göstermektedir. Bu makale, parametre sıkıştırmanın nispeten yüksek veri hızlarında hesaplamadan daha az önemli olduğu Şekil 4a'daki 10 Mbps gibi yüksek yukarı bağlantı/aşağı bağlantı veri hızlarını dikkate almaktadır. Bu durumda hesaplama, federe öğrenmeyi ölçeklendirmek için iletişimden daha büyük bir darboğaz haline gelmiştir. Şekil 4a'da ADACOMM'nin 10/10 Mbps veri hızlarında çoklu yerel güncellemeler nedeniyle parametre sıkıştırma benimsen ATOMO'dan daha iyi performans gösterdiği ve önerilen FedEAFO'nun parametre sıkıştırma sağladığı faydalar nedeniyle ADACOMM'den daha iyi performans gösterdiği gösterilmektedir. Şekil 4b'de, yukarı bağlantı iletişiminin darboğaz haline geldiği 10/100 Kbps veri hızları dikkate alındığında, ATOMO yukarı bağlantıdaki parametreleri sıkıştırdığı için ADACOMM ile yarışı kazanmaktadır. Buna ek olarak, önerilen FedEAFO çoklu yerel yukarı bağlantıdan sağladığı fayda nedeniyle ATOMO'dan daha iyi performans göstermektedir.

tarihlerini göstermektedir. Aksine, Şekil 4c'de, 100/10 Kbps veri hızları dikkate alındığında, aşağı bağlantı iletişimi darboğaz haline gelmiştir. Dolayısıyla, ATOMO'nun performansı, yukarı bağlantıdaki parametre sıkıştırmasına rağmen ADACOMM'un performansına benzerdir.



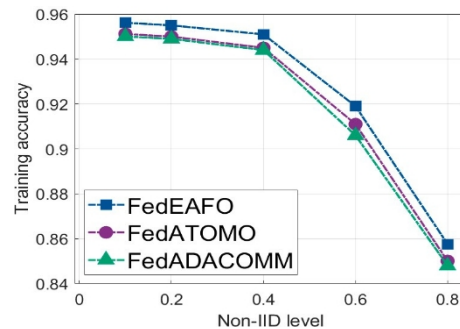
Şekil 4. Zaman içinde farklı yukarı bağlantı/aşağı bağlantı veri hızları ile doğruluk değerleri.

Şekil 5, farklı algoritmaların farklı hedef doğruluğuna ulaştıklarında ağ trafiği tüketimini göstermektedir. Tüm veri kümeleri için tüm algoritmaların ağ trafiği tüketiminin artan doğrulukla birlikte arttığını görebiliriz. Bununla birlikte, FedEAFO her zaman minimum ağ trafiğini tüketebilir, bu da FedEAFO'nun iletişim verimliliğini gösterir. Buna ek olarak, model sıkıştırma algoritmaları, model sıkıştırması olmayan algoritmalarla göre çok daha fazla ağ trafiği tasarrufu sağlayabilir.



Şekil 5. Hedef doğruluğa ulaşırken algoritmaların ağ trafiği tüketimi.

Bu makale ayrıca FedEAFO'nun ve temel hatların performansını IID olmayan veriler üzerinde değerlendirmektedir. Somut olarak, farklı bölümlere şemaları ile Fashion-MNIST veri kümesinin IID olmayan verilerini oluşturuyoruz. Her cihaz 10 sınıfta benzersiz bir sınıfa sahiptir ve diğer cihazlar kalan örnekleri eşit olarak paylaşmıştır. Şekil 6, p değerinin 0,1, 0,2, 0,4, 0,6 ve 0,8 olarak ayarlandığı IID olmayan seviyelerin farklı algoritmaların test doğruluğu üzerindeki etkilerini göstermektedir. Sonuç, tüm algoritmaların IID olmayan seviyedeki artışla birlikte doğruluk kaybına uğradığını, FedEAFO'nun ise diğer temel hatlardan daha dayanıklı olduğunu göstermektedir.



Şekil 6. Algoritmaların IID olmayan veriler üzerindeki doğruluk değerleri.

6. Sonuçlar

FL, cihazların gizliliğe duyarlı veriler yerine yerel model parametrelerini paylaşarak makine öğrenimi modellerini işbirliği içinde eğitmelerini sağlama özelliği nedeniyle IoT'de büyük ilgi görmüştür. Bununla birlikte, FL-IoT uygulaması, kaynak kısıtlı IoT ortamlarında verimsizlik ve sınırlı yakınsama performansından muzdariptir. FL eğitiminin verimliliğini artırmak için yeni bir verimli uyarlanabilir federasyon optimizasyon algoritması öneriyoruz. Önerilen FedEAFO, yerel güncelleme ve parametre sıkıştırma katsayısından oluşan iki değişkeni birlikte değerlendirerek öğrenme hatasını en aza indirir. Buna ek olarak, önerilen FedEAFO, federe öğrenmenin iletişim, hesaplama ve model hassasiyeti arasındaki dengeleri uyarlanabilir ve dinamik olarak ayarlamasını sağlar. Değerlendirme sonuçları, FedEAFO'nun verimlilik ve doğruluk açısından üstünlüğünü göstermektedir. Gelecekteki çalışmalarda, FedEAFO'nun kararsız ortamlarda dağıtılmış optimizasyona olanak tanıyan mesaj geçirme protokolleriyle etkileşimlerini inceleyeceğiz.

Yazar Katkıları: Kavramsallaştırma, Z.C. ve E.W.; metodoloji, Z.C. ve E.W.; yazılım, Z.C. ve X.Y.; doğrulama, Z.C. ve H.C.; resmi analiz, Z.C.; araştırma, Z.C. ve X.Y.; kaynaklar, H.C.; veri kütürlüğü, Z.C. ve X.Y.; yazım-orijinal taslak hazırlama, Z.C.; yazım-inceleme ve düzenleme, E.W. ve H.C.; görselleştirme, Z.C. ve X.Y.; denetim, H.C.; proje yönetimi, Z.C. ve H.C.; fon sağlama, H.C. Tüm yazarlar makalenin yayınlanan versiyonunu okumuş ve kabul etmiştir.

Finansman: Bu araştırma, Hibe No. 62171049 altında Çin Ulusal Doğa Bilimleri Vakfı ve Hibe No. 2020YFB1807805 altında Çin Ulusal Anahtar Araştırma ve Geliştirme Programı tarafından finanse edilmiştir.

Veri Kullanılabilirlik Beyanı: Bu çalışmada yeni veri oluşturulmamış veya analiz edilmemiştir. Veri paylaşımı bu makale için geçerli değildir.

Çıkar Çatışmaları: Yazarlar herhangi bir çıkar çatışması beyan etmemektedir.

Referanslar

1. Haddad Pajouh, H.; Dehghantanha, A.; Parizi, R.M.; Aledhari, M.; Karimipour, H. Nesnelerin interneti güvenliği üzerine bir araştırma: Gereksinimler, zorluklar ve çözümler. *IEEE Internet Things J.* **2021**, *14*, 100129.
2. Liu, Y.; Wang, J.; Li, J.; Niu, S.; Song, H. Nesnelerin İnterneti Cihazlarının Tespiti ve Tanımlanması için Makine Öğrenimi: Bir Anketi. *IEEE Internet Things J.* **2021**, *9*, 298-320.
3. Ghosh, A.M.; Grolinger, K. Nesnelerin İnterneti veri analitiği için uç-bulut bilişim: derin öğrenme ile uca zeka gömme. *IEEE Trans. Ind. Inform.* **2021**, *17*, 2191-2200.
4. Khan, L.U.; Saad, W.; Han, Z.; Hossain, E.; Hong, C.S. Nesnelerin İnterneti için Federe Öğrenme: Son Gelişmeler, Taksonomi, ve Açık Zorluklar. *IEEE İletişim. Surv. Eğitmen.* **2021**, *23*, 1759-1799.
5. Zhang, T.; Gao, L.; He, C.; Zhang, M.; Krishnamachari, B.; Avestimehr, A.S. Nesnelerin İnterneti için Federe Öğrenme: Uygulamalar, Zorluklar ve Fırsatlar. *IEEE Internet Things Mag.* **2022**, *5*, 24-29.
6. Kandati; Reddy, D.; Gadekallu, T.R. COVID-19 Tespiti için Genetik Kümelenmiş Federe Öğrenme. *Elektronik* **2022**, *11*, 2714.
7. Posner, J.; Tseng, L.; Aloqaily, M.; Jararweh, Y. Araç Ağlarında Federe Öğrenme: Fırsatlar ve Çözümler. *IEEE Netw.* **2021**, *35*, 152-159.
8. Mills, J.; Hu, J.; Min, G. İletişim açısından verimli federe öğrenme için istemci tarafı optimizasyon stratejileri. *IEEE İletişim. Mag.* **2022**, *60*, 60-66.
9. Nguyen, C.; Ding, M.; Pathirana, P.N.; Seneviratne, A.; Li, J.; Vincent Poor, H. Nesnelerin İnterneti için Federe Öğrenme: Kapsamlı Bir Araştırma. *IEEE İletişim. Hayatta Kalma. Eğitmen.* **2021**, *23*, 1622-1658.
10. Sun, H.; Li, S.; Yu, F.R.; Qi, Q.; Wang, J.; Liao, J. Toward Communication-Efficient Federated Learning in the Internet of Things with Edge Computing. *IEEE Internet Things J.* **2020**, *7*, 11053-11067.
11. Ullo, S.L.; Sinha, G.R. Uzaktan algılama ve tarım uygulamaları için IoT ve akıllı sensörlerdeki gelişmeler. *Remote Sens.* **2021**, *13*, 2585.
12. Qin, Z.; Li, G.Y.; Ye, H. Federe öğrenme ve kablosuz iletişim. *IEEE Wirel. Commun.* **2021**, *28*, 134-140.
13. Nguyen, H.T.; Schwag, V.; Hosseinalipour, S.; Brinton, C.G.; Chiang, M.; Vincent Poor, H. Fast-Convergent Federated Learning. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 201-218.
14. Shah, S.M.; Lau, V.K.N. İletişim Verimli Federasyonlu Öğrenme için Model Sıkıştırma. *IEEE Trans. Sinir Ağları. Öğrenmek. Syst.* **2021**, *3*, 1-15.

15. Reisizadeh, A.; Mokhtari, A.; Hassani, H.; Jadbabaie, A.; Pedarsani, R. Fedpaq: Periyodik Ortalama Alma ve Niceleme ile İletişim Verimli Birleştirilmiş Öğrenme Yöntemi. Uluslararası Yapay Zeka ve İstatistik Konferansı Bildirileri (PMLR), Sanal Konferans, 26-28 Ağustos 2020; s. 2021-2031.
16. Mitra, A.; Jaafar, R.; Pappas, G.J.; Hassani, H. Federe öğrenmede doğrusal yakınsama: İstemci heterojenliği ve seyrek gradyanlarla başa çıkma. *Adv. Neural Inf. Süreç. Syst.* **2021**, *34*, 14606-14619.
17. Zheng, S.; Meng, Q.; Wang, T.; Chen, W.; Yu, N.; Ma, Z.M.; Liu, T.Y. Gecikme Telafisi ile Asenkron Stokastik Gradyan İnişi. Uluslararası Yapay Zeka ve İstatistik Konferansı Bildirileri (PMLR), Fort Lauderdale, FL, ABD, 20-22 Nisan 2017; s. 4120-4129.
18. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Heterojen ağlarda federasyon optimizasyonu. *Proc. Mach. Learn. Syst.* **2020**, *2*, 429-450.
19. Wang, J.; Joshi, G. Yerel güncellemeli SGD'de en iyi hata-çalışma süresi dengesini elde etmek için uyarlanabilir iletişim stratejileri. *Proc. Mach. Learn. Syst.* **2019**, *1*, 212-229.
20. Wang, S.; Tuor, T.; Salonidis, T.; Leung, K.K.; Makaya, C.; He, T.; Chan, K. Kaynak Kısıtlı Kenar Bilişim Sistemlerinde Uyarlanabilir Federe Öğrenme. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 1205-1221.
21. Wen, W.; Xu, C.; Yan, F.; Wu, C.; Wang, Y.; Chen, Y.; Li, H. Terngrad: Dağıtılmış derin öğrenmede iletişimi azaltmak için üçlü gradyanlar. *Adv. Neural Inf. Süreç. Syst.* **2017**, *4*, 30-43.
22. Alistarh, D.; Grubic, D.; Li, J.; Tomioka, R.; Vojnovic, M. QSGD: Gradyan niceleme ve kodlama yoluyla iletişim açısından verimli SGD. *Adv. Neural Inf. Süreç. Syst.* **2017**, *30*, 47-60.
23. Seide, F.; Fu, H.; Droppo, J.; Li, G.; Yu, D. 1-bit stokastik gradyan inişi ve konuşma DNN'lerinin veri-paralel dağıtılmış eğitimine uygulanması. Uluslararası Konuşma İletişimi Derneği On Beşinci Yıllık Konferansı Bildirileri, Singapur, 14-18 Eylül 2014; s. 20-27.
24. Lin, Y.; Han, S.; Mao, H.; Wang, Y.; Dally, B. Derin Gradyan Sıkıştırma: Dağıtılmış Eğitim için İletişim Bant Genişliğini Azaltma. Uluslararası Öğrenme Temsilleri Konferansı (ICLR) Bildirileri, Vancouver, BC, Kanada, 30 Nisan-3 Mayıs 2018; s. 40-48.
25. Wang, H.; Sievert, S.; Liu, S.; Charles, Z.; Papailiopoulos, D.; Wright, S. Atom: Atomik ayrıştırma yoluyla iletişim açısından verimli öğrenme. *Adv. Neural Inf. Süreç. Syst.* **2018**, *31*, 10-24.
26. Xu, J.; Du, W.; Jin, Y.; He, W.; Cheng, R. İletişim Verimli Federe Öğrenme için Üçlü Sıkıştırma. *IEEE Trans. Sinir Ağları. Öğrenmek. Syst.* **2022**, *33*, 1162-1176. [CrossRef] [PubMed]
27. Haijian, S.; Ma, X.; Hu, R.Q. Yukarı bağlantı NOMA'da gradyan sıkıştırma uyarlamalı federasyonlu öğrenme. *IEEE Trans. Veh. Teknol.* **2020**, *69*, 16325-16329.
28. Zha, Z.; Yuan, X.; Wen, B.; Zhang, J.; Zhou, J.; Zhu, C. Ortak Yama-Grup Tabanlı Seyrek Temsil Kullanarak Görüntü Restorasyonu. *IEEE Trans. Görüntü İşleme.* **2020**, *29*, 7735-7750.
29. Sattler, F.; Müller, K.R.; Samek, W. Kümelenmiş federe öğrenme: Gizlilik kısıtlamaları altında modelden bağımsız dağıtılmış çoklu görev optimizasyonu. *IEEE Trans. Neural Netw. Öğrenmek. Syst.* **2020**, *32*, 3710-3722.
30. Ananduta, W.; Nedich, A.; Ocampo-Martinez, C. Bağlantı Tabanlı Kaynak Paylaşımı için Dağıtılmış Artırılmış Lagrangian Yöntemi Çok Ajanlı Sistemlerin Problemleri. *IEEE Trans. Otom. Kontrol.* **2021**, *8*, 10-23.
31. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-mnist: Makine öğrenimi algoritmalarını kıyaslamak için yeni bir görüntü veri kümesi. *arXiv* **2017**, arXiv:1708.07747.
32. Zheng, Y.; Lai, S.; Liu, Y.; Yuan, X.; Yi, X.; Wang, C. Federe öğrenme için toplama hizmeti: Verimli, güvenli ve daha dirençli bir gerçekleştirme. *IEEE Trans. Dependable Secur. Hesaplama.* **2022**, *20*, 988-1001.

Feragatname/Yayıncının Notu: Tüm yayınlarda yer alan ifadeler, görüşler ve veriler MDPI ve/veya editör(ler)e değil, yalnızca yazar(lar)a ve katkıda bulunan(lar)a aittir. MDPI ve/veya editör(ler), içerikte atıfta bulunulan herhangi bir fikir, yöntem, talimat veya üründen kaynaklanan insanlara veya mallara gelebilecek herhangi bir zarar için sorumluluk kabul etmez.