TSINGHUA SCIENCE AND TECHNOLOGY ISSN 1007-0214 08/24 pp400-414 DOI: 10 . 26599 / TST. 2023 . 9010007
Cilt 29, Sayı 2, Nisan 2024

Nesnelerin İnterneti Kritik Altyapısı Kapsamında Federe Öğrenme Güvenliği ve Gizliliği Koruyan Algoritma ve Deney Araştırması

Nasir Ahmad Jalali ve Hongsong Chen*

Özet: Nesnelerin İnternetinin (IoTs) yaygın kullanımı ve yapay zeka teknolojilerinin hızlı gelişimi, uygulamaların ticari ve endüstriyel bant ortamlarını aşmasını sağlamıştır. Bu tür sistemlerde, ticari ve endüstriyel sistemlerle ilgili tüm katılımcılar iletişim kurmalı ve veri üretmelidir. Ancak, IoT cihazlarının küçük depolama kapasiteleri nedeniyle, üretilen verileri depolamak ve verilerini depolamak için tek bir nokta oluşturan "bulut" adı verilen üçüncü taraf varlığa aktarmaları gerekmektedir. Ancak katılımcı sayısı arttıkça üretilen verinin boyutu da artmaktadır. Bu nedenle, katılımcılar arasında veri toplama ve alışverişi için bu tür merkezi bir mekanizmanın güvenlik, gizlilik ve performans açısından çok sayıda zorlukla karşılaşması muhtemeldir. Bu zorlukların üstesinden gelmek için, Federated Learning (FL), istemcilerin artık gerçek verileri merkezi sunucuya aktarmasına ve depolamasına gerek kalmayan makul bir merkezi olmayan yaklaşım olarak önerilmiştir. Bunun yerine, yalnızca kendi özel veri kümeleri üzerinde eğitilen güncellenmiş eğitim modellerini paylaşırlar. Aynı zamanda FL, dağıtık sistemlerdeki istemcilerin makine öğrenimi modellerini eğitim verileri olmadan işbirliği içinde paylaşmalarını sağlayarak veri gizliliği ve güvenlik sorunlarını azaltır. Bununla birlikte, yavaş model eğitimi ve ek gereksiz iletişim turlarının yürütülmesi, FL uygulamalarının dağıtılmış bir sistemde düzgün çalışmasını engelleyebilir. Ayrıca, bu gereksiz iletişim turları sistemi güvenlik ve gizlilik sorunlarına karşı savunmasız hale getirmektedir, çünkü istemciler ve sunucular arasında alakasız model güncellemeleri gönderilmektedir. Bu nedenle, bu çalışmada, yerel bilgi gizliliğini koruyan işlevleri için model parametrelerini şifrelemek üzere Cheon-Kim-Kim-Song (CKKS) adı verilen tam homomorfik şifreleme için bir algoritma öneriyoruz. Önerilen çözüm, model eğitim süreci sırasında model yakınsamasını hızlandırmak için ivme terimini kullanır. Ayrıca, IoT cihazları ile sunucu arasında güvenli bir iletişim kanalı kurar. Ayrıca, iletişim ek yükünü azaltmak için hafif bir güvenli taşıma protokolü kullanıyoruz, böylece istemci ve sunucu arasında düşük iletişim gecikmesi ile iletişim güvenliğini ve verimliliğini artırıyoruz.

Anahtar kelimeler: Federe Öğrenme (FL); Nesnelerin İnterneti (IoTs); Hafif Taşıma Katmanı Güvenliği (iTLS); Cheon- Kim-Kim-Song (CKKS)

1 Giriş

İnsan yaşam kalitesini artırma hedefiyle, Nesnelerin İnterneti (IoT) destekli akıllı sistemler son yıllarda giderek daha popüler hale gelmiştir. Akıllı sistem konsepti, akıllı telefonlar ve Yapay Zeka (AI) gibi IoT cihazlarının yanı sıra bulut, uç bilişim ve büyük veri analitiği dahil olmak üzere ilgili depolama ve hesaplama mekanizmaları tarafından etkinleştirilmiştir. Bununla birlikte, bu tür akıllı sistemlerdeki düğümlerin artması büyük miktarda veri ürettiğinden, IoT cihazlarının sınırlı kapasitesi nedeniyle, tüm

• Nasir Ahmad Jalali ve Hongsong Chen, Pekin Bilim ve Teknoloji Üniversitesi (USTB), Bilgisayar Bilimleri Bölümü, Pekin 100083, Çin. E-posta: nasir.zehand@gmail.com; chenhs@ustb.edu.cn.
×Yazışmaların yapılacağı kişi.
Makale alındı: 2022-07-20; revize: 2022-12-07; kabul tarihi: 2023-02-13

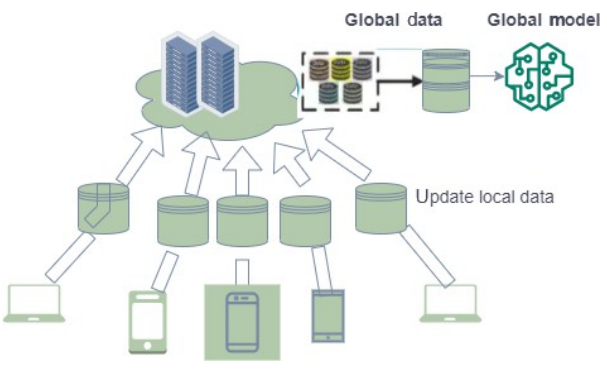
üretilen veriler üçüncü taraf (bulut) sunuculara aktarılmalı ve depolanmalıdır[1] . Bu kadar büyük miktarda verinin hesaplanması ve işlenmesi için son derece gelişmiş teknikler gereklidir. Makine Öğrenimi (ML) teknikleri, üç ana özelliği (büyük veri kullanılabilirliği, daha yeni öğrenme modeli hesaplama gücü ve derin öğrenme modelinin evrimi) yaygın kullanımlarına ve uygulamalarına önemli ölçüde katkıda bulunan oldukça başarılı tekniklerdir[2] . ML'nin büyük ve devasa veri analizi ve işlemede başarı göstermesine rağmen, çoğu alan bunları gerçek dünyada kullanmaya istekli değildir. Bir çalışma[3] günümüzün rekabetçi iş ortamlarında veri sahiplerinin güvenlik, gizlilik ve performans kaygıları nedeniyle verilerini merkezi bir sistemle paylaşmak istemediklerini bildirmektedir. Bunun nedeni, tek bir nokta oluşturmaları, dolayısıyla merkezi sunucunun bir saldırıya maruz kalması halinde depolanan tüm verilerin istismar edilecek olmasıdır. Veri sahipleri tarafından belirtilen diğer bazı nedenler arasında aşağıdakiler yer almaktadır:

- Kullanıcının veri gizliliğine ilişkin endişeler;
- Verilerin öğrenme için istemcilerden merkezi sunucuya aktarılmasından kaynaklanan gizlilik sorunları;
- Etkili öğrenme gelişimi için uç ağdaki önemli miktarda veri ve bilgi işlem kaynağının kullanılamaması.

Aşağıdakilerle ilgili diğer gizlilik ve güvenlik endişeleri

ML, sistem bazı şirketler tarafından kullanılırken veri sahiplerinin veri kümelerini hesaplama ve işleme için merkezi bir konuma aktaramaması gerçeğini içerir [4,5] . Şekil 1'de gösterildiği gibi, tüm veriler toplanır ve merkezi bir sunucuda saklanır ve model merkezi veri kümesi tarafından eğitilir.

Yukarıda belirtildiği gibi, tüm kullanıcılar üçüncü taraf bir bulut sunucusuna veri aktarırken veri gizliliği ve mahremiyeti konusunda endişe duymaktadır. İstemciler verilerini merkezi sunucuya gönderdiğinde, hızlı hesaplama elde edemeyebilir. İstemcilerin alması gerektiği gibi



Şekil 1 Merkezi veri kümesi ile makine öğrenimi modeli.

yanıtlarını anında alamadıklarından, bu kaynakları öğrenme iyileştirmeleri için etkili bir şekilde kullanamamaktadırlar. Bu zorlukların üstesinden gelmek için, ML kavramına dayanan Federated Learning (FL) en iyi ve en uygun fiyatlı çözüm olarak önerilmiştir[6] . Son zamanlarda yapılan çalışmalar FL'nin çok sayıda avantajı olduğunu göstermiş olsa da FL aynı zamanda güvenlik, gizlilik, verimlilik ve iletişim maliyeti sorunlarıyla da karşı karşıyadır. Örneğin, FL sisteminde gradyanlar merkezi sunucuya aktarılırken güvenilir olmayan model üretimi veya önceki verilerle ilgili bilgiler nedeniyle özel veriler sızdırılabilir[7] . Bu nedenle, FL sisteminde, Homomorfik Şifreleme (HE) şeması aracılığıyla verilerin şifrelenmesi ve sızıntıya karşı korunması makul bir seçimdir. Şimdiye kadar araştırmacılar, beklentilerine dayanarak FL sisteminde bilgi gizliliğini verimli bir şekilde korumak için çalışmışlardır. Ancak, FL sistemindeki yakınsama performansının iyileştirilmesini dikkate almamışlardır[8] . Ayrıca, FL sisteminin temel unsurları olan model doğruluğu, model yakınsama oranı ve düşük gecikmeli iletişim maliyeti açısından verimli performans, dağıtılmış katılımcıların veri kaynaklarının güvenliğini ve gizliliğini sağlamak için dikkate alınmalıdır.

2 İlgili Çalışma

Dağıtık öğrenme sistemlerinde FL kullanımı giderek artan bir ilgi görmektedir. Bununla birlikte, güvenlik ve gizlilik sızıntısı endişeleri, güvenli bir ortamda istemcilerle zamanında iletişim kurmak ve yanıt vermek için yüksek performans, düşük iletişim ek yükü ve düşük gecikme gereksinimi gibi bazı sorunlar henüz tam olarak çözülmemiştir. Son saldırılar, model istemciler ve merkezi sunucu arasında aktarılırken kişisel özel ve hassas verilerin sızdırılabileceğini ortaya koymaktadır. Diferansiyel gizlilik bir güvenlik ve gizlilik mekanizması olarak kullanılsa da, model eğitim maliyetlerinin artmasına ve model performans kalitesinin düşmesine neden olmuştur[9] . FL'deki IoT cihazları için güvenli iletişim çok önemlidir. Bununla birlikte, Refs [10] ve [11]'deki çalışmalara göre, Taşıma Katmanı Güvenliği (TLS) protokolü ve Datagram TLS (DTLS) protokollerini kullanan mevcut yaklaşımlar, ağır iletişim ek yükü, yüksek gecikme süresi, düşük güvenlik ve bir istemci ile sunucu arasında birçok gidiş-dönüş mesajı gerektiren taşıma protokolü değişimleri nedeniyle zorlanmaktadır. Güvenli çok partili hesaplama gibi farklı HE türlerinin, kullanıcıların ayrılmasına karşı sağlam olduğu gösterilmiştir; ancak bunlar aynı zamanda büyük iletişim ve

hesaplama ek yükü[12] . Bu sorunu çözmek için, RSA gibi diğer şifreleme algoritmalarına göre mükemmel bir şifreleme hızına sahip olan Cheom-Kim-Kim-Song (CKKS) şeması önerilmiştir. Şemamız, gizliliği korumak, model yakınsamasını hızlandırmak, iletişim ve hesaplama ek yükünü azaltmak, düşük gecikme süresi elde etmek ve iletişim mesajlarının gereksiz gidiş geliş sayısını azaltmak ve sınırlamak amacıyla model parametrelerini şifrelemek için hem CKKS algoritmasını hem de hafif Taşıma Katmanı Güvenliği (iTLS) protokolünü kullanır.

2.1 Motivasyon ve sorun bildirimi

FL, kullanıcı verilerini yerel olarak tutarak kullanıcı veri gizliliğini artıran bir dağıtıcı ML tekniğidir. Bununla birlikte, dağıtım ML sistemleriyle ilgili belirli güvenlik ve performans zorluklarına eğilimlidir. Gizlilik ve güvenlikle ilgili birçok çalışma yapılmış olmasına rağmen, gerçek hayattaki uygulamalar sırasında ortaya çıkan FL ile ilgili performans, gizlilik, doğruluk ve güvenlik zorluklarını tamamen ortadan kaldıramamıştır. Bu nedenle, yukarıda bahsedilen zorluklardan hareketle, dağıtık öğrenme makineleri için güvenli ve yüksek performanslı bir ortam sağlamayı amaçlıyoruz. Bu çalışmanın potansiyel katkıları iki yönlüdür. İlk olarak, yerel bilgi gizliliğinin korunması ve model eğitimi sırasında daha hızlı model yakınsaması ve hesaplaması için model parametrelerini şifrelemek üzere tam homomorfik bir algoritma, yani CKKS öneriyoruz. İkinci olarak, düşük gecikme süresiyle verimli ve güvenli iletişim ek yükünü teşvik etmek için hafif bir protokol kullanıyoruz.

2.2 Kağıt yapısı

Çalışmanın geri kalanı farklı bölümlere ayrılmıştır. Bölüm 2'de ilgili çalışmalar sunulmakta, Bölüm 3'te ise FL sistemindeki güvenlik ve gizliliğin korunmasına odaklanılarak FL modelinin görünümü incelenmektedir. Bölüm 4, IoT ortamı altında FL'deki zorlukları ele alırken, Bölüm 5 deneysel ortam ve yapılandırma dahil olmak üzere deneysel kısmı kapsamaktadır. Son olarak, Bölüm 6 tüm makalenin sonucunu sunmaktadır.

3 FL Modeli Hakkında Peyzaj

Araştırma topluluğu, veri gizliliği, gizlilik, yüksek hızlı iletişim ve yüksek hesaplama gücü gibi iyi eğitimli merkezi makine öğrenimi ile ilgili yukarıda belirtilen sınırlamaların ve sorunların üstesinden gelmek için FL çerçevesini geliştirmiş ve önermiştir. FL şudur IoT cihaz iletişimi için en iyi ve en uygun fiyatlı çözüm olarak kabul edilmektedir[13] . FL ayrıca, tüm verilerin merkezileştirilmesi ve modelin merkezi bir sunucuda eğitilmesi uygulamasına kıyasla yukarıda belirtilen zorlukları çözebilen bir ML tekniğidir. Bunun nedeni, FL modelinin tüm verileri merkezi bir sunucuya aktarmadan ve toplamadan merkezi olmayan verileri eğitebilmesidir. Şekil 2'de gösterildiği gibi, her istemci yerel verileri kullanarak modeli eğitir ve ardından yerel olarak eğitilmiş modelleri merkezi sunucuya gönderir. Sunucu yerel modelleri aldığında, bunları toplar, bir küresel model oluşturur ve bilgileri güncellemek için tüm katılımcılara geri gönderir. Katılımcılar bu faaliyetleri gerçek yerel verileri aktarmadan dolaştırırlar.

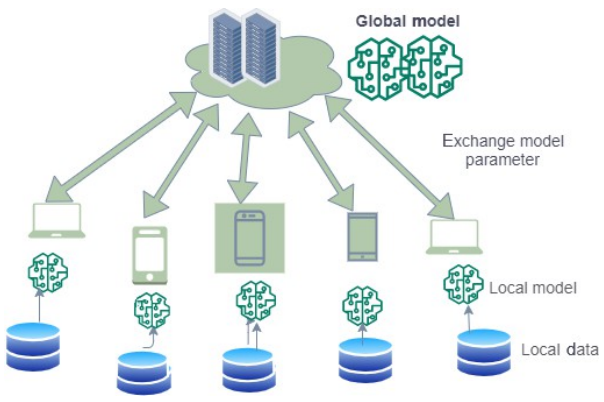
FL model güncellemesi için aşağıda açıklanan süreç, her bir iterasyonun merkezi ML modelini geliştirdiği iteratif bir süreçtir. Süreç aşağıdaki gibi üç ana adım halinde düzenlenmiştir:

Adım 1: Model seçimi. Global model, modeli tüm başlangıç parametreleriyle önceden eğitir ve çıktıyı FL ortamına dahil olan tüm istemcilerle paylaşır.

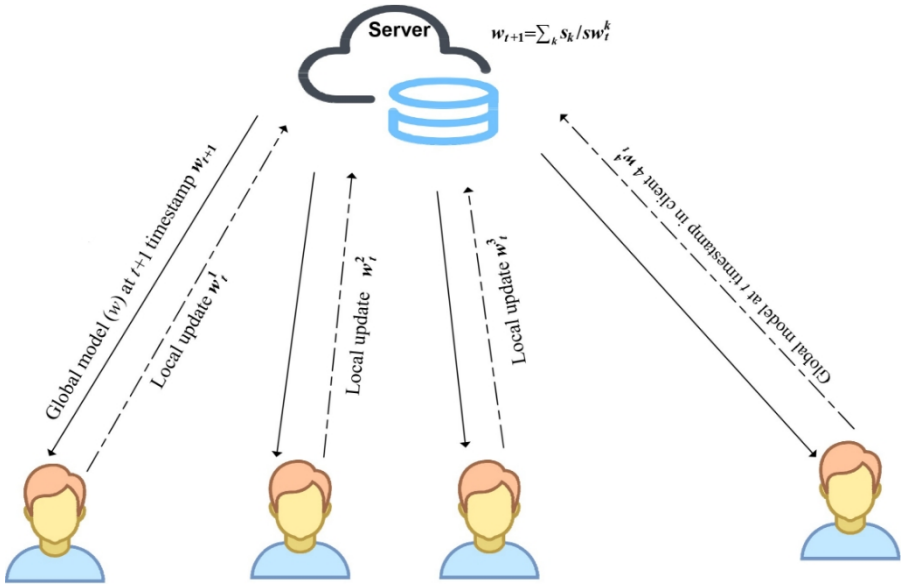
Adım 2: Yerel model eğitimi. Global bir model istemcilerle paylaşıldığında, her düğüm modeli kendi yerel veri kümesi üzerinde eğitir.

Adım 3: Yerel modeli topla. Bir model, istemci tarafında yerel olarak eğitildiğinde güncellenir. Ardından, modelin yeni sürümü merkezi nokta ile paylaşılır. Sunucu, global modeli yeni parametrelerle günceller ve toplar ve bilgileri güncelleyecek iterasyonu başlatmak için istemcilere geri gönderir.

Yukarıdaki adımlar, FL çerçevesinin ML modelini birkaç cihaz üzerinde eğittiğini ve gerçek verileri merkezi sunucuya (bulut) aktarmaya gerek olmadığını göstermektedir. Ayrıca, yalnızca eğitim modeli, modeli güncellemek için Adım 2 ve 3'ü dolaşarak verileri istemciler ve sunucular arasında paylaşacaktır. Şekil 3 nasıl çalıştığının ayrıntılarını göstermektedir



Şekil 2 Federe öğrenme sistemi.



Şekil 3 Federe öğrenme modeli güncelleme süreci.

model FL sisteminde güncellenir.

Şekil 3'te gösterildiği gibi, süreç global model r_t başlatma ile başlar ve model istemcilere gönderilir. Modeli aldıktan sonra, her "k" istemcisi modeli kendi yerel verileriyle eğitir; burada s_k istemciler tarafından tutulan eğitilmiş örneklerin sayısıdır ve s *istemciler* tarafından eğitilen toplam örnek sayısıdır. Daha sonra güncellenmiş model r^k sunucuya geri döner ve sunucu global modeli r_{tCl}[14] güncellemek için alınan tüm modelleri toplar. Bu şekilde, veri gizliliği istemcilerin ortamlarında yerel olarak korunur. FL çerçevesindeki tüm IoT cihazları, modeli eğitmek ve gizliliği verimli bir şekilde korumak için hesaplama işlemlerini gerçekleştirmekten sorumludur[15] .

3.1 FL'de güvenlik ve gizliliğin korunması

Günümüzün veri odaklı dünyasında, eğitim, sağlık, tıp, akıllı şehir ve finansal uygulamalar gibi çoğu hizmet ve uygulama, IoT cihazları gibi büyük ölçüde yapay zeka teknolojisine dayanmaktadır. Farklı uygulamalar, karmaşık bir makine öğrenimi modeli üzerinde çalışan çeşitli veri türleri üretebilir; bu nedenle, kullanıcılar veri koruma ve gizliliğe dikkat etmelidir[16] . Bununla birlikte, YZ teknolojisi henüz tam potansiyeline ulaşmamıştır ve YZ ve ML ile ilgili uygulamalar, merkezi depolama ve hesaplama ile ilgili birçok zorlukla karşılaşmaya devam etmektedir[17] . YZ teknolojileri, özellikle kişisel veriler olmak üzere veriler üretmekte ve ister son kullanıcı ister hizmet sağlayıcı ile ilgili olsun, sınırlı depolama kapasiteleri nedeniyle bunları bir veri silosunda saklamaktadır. Çoğu makine öğrenimi algoritmaları merkezi bir şekilde çalıştığından, eğitim verilerini entegre etmek, toplamak ve işlemek için merkezi sunucuya veri toplamaları gerekir[18] . Geleneksel bir makine öğrenimi algoritmasında, büyük miktarda verinin bir bulutta veya merkezi sunucuda toplanması tek bir hata noktasına yol açabilir ve veri ihlali gibi güvenlik risklerine neden olabilir. Bu merkezi veri işleme ve yönetimi, şeffaflık ve provenansta sınırlamalara yol açarak son kullanıcılar arasında güvensizliğe ve zor dağıtıma neden olabilir[19] . FL, bu tür zorlukların üstesinden gelmek için en iyi ve en uygun fiyatlı çözümdür. FL, algoritmanın çeşitli ayrılmış veri kümeleri üzerinde yürütüldüğü merkezi olmayan bir işbirlikçi öğrenme modelinde ML algoritmaları tarafından kullanılan bir tekniktir. IoT cihazlarından gelen tüm verilerin işlem için merkezi bir sunucuya toplanmasına gerek yoktur. Bunun yerine, katılımcılar yalnızca güncellenmiş modelleri merkezi bir koordineli sunucu [20,21] ile değiş tokuş eder. FL'nin geleneksel bulut merkezli ML'ye göre en büyük avantajı, birincisinin istemci tarafında yerel olarak işlenen verilerin gizliliğini sağlaması ve yalnızca model parametrelerinin değiş tokuş edilmesini gerektirmesidir. Bazı gizliliği koruyan ve güvenlik teknikleri, merkezi koordineli sunucular ve istemciler arasındaki bu aktarımı ve toplamayı güçlendirerek gelişmiş veri gizliliği ve güvenliği sağlar [22,23] . ML teknolojisi, doğrudan insan talimatları olmadan bir örnek kümesi (eğitim verileri) kullanarak matematiksel algoritmalar ve modellere (model eğitim süreci) dayalı olarak faaliyetlerini otomatik olarak öğrenen ve işleyen IoT cihazları gibi akıllı sistemler tasarlamak ve oluşturmak için kullanılır[18] .

3.2 FL'de iletişim maliyeti ve model doğruluğu

Bölüm 2'de belirtildiği gibi FL, çok sayıda IoT istemcisinden gelen dağıtılmış veri kümeleri üzerinde yüksek kalitede küresel bir modeli eğitmek için bir tür makine öğrenimi yaklaşımıdır. Bu IoT istemcileri güvenilmez ve yavaş ağ bağlantılarına sahip olabilir; bu nedenle, her istemci kendi yerel veri kümesine dayalı olarak mevcut modeli bağımsız olarak eğitebilir, güncelleyebilir ve hesaplayabilir. Ayrıca, yerel istemciler yeni bir model hesaplamak ve merkezi sunucuya iletişim kurmak veya geri dönmek için toplamları günceller, böylece iletişim maliyeti ve verimliliğinin önemini vurgular[24] . FL öğrenme sisteminde, tüm katılımcılar aktif bağlantılar elde etmek ve istenen doğruluğu elde etmek için merkezi sunucu ile iterasyon turlarını sürdürür. Ancak, gereksiz iterasyonlar nedeniyle, model parametreleri uygun ortamlarda bile sızdırılabilir. Buna karşılık, ML çok sayıda parametre içerir ve işler, bu da katılımcılar güncellemeleri yüklerken yüksek iletişim maliyetlerine ve gecikmelere neden olur[25] .

4 IoT Ortamında FL Uygulamasındaki Zorlukların Ele Alınması

Daha önce yapılan bir çalışmada[26] IoT cihazlarının sayısının dünya çapında 75 milyara ulaşacağı öngörülmüştü. Böylesine inanılmaz bir gelişme, bu IoT cihazları tarafından üretilen dağıtılmış verilerde muazzam bir büyümeye neden olacaktır. Bu tür veriler, tüm verilerin istemcilerden aktarıldığı ve toplu olarak sunucuda depolandığı merkezi depolamaya aktarılır. Bu nedenle, merkezi bir noktaya sahip olmak, veri güvenliği ve gizliliğinin tehlikeye atılması gibi güvenlik açıklarına yol açabilir. Son yıllarda Google, güvenlik ve gizlilik sorunlarını azaltmak için FL'yi makul bir çözüm olarak tanıttı. FL, bir istemcinin yerel verilerini merkezi sunucuya[27] aktarmadan tüm dağıtılmış istemcilerin ortak bir ML modelini işbirliği içinde öğrenmesini sağlar. Bununla birlikte, FL hala güvenlik ve gizlilik riskleriyle karşı karşıyadır çünkü yerel olarak eğitilmiş veri kümeleriyle ilgili bilgilerin modelin parametrelerinden sızması hala mümkündür. Bu nedenle, güvenliği artırmak ve gizlilik risklerini azaltmak için HE algoritmaları gibi özel güvenlik ve gizlilik algoritmalarının uygulanması önemlidir.

4.1 HE

Bu teknik, ML algoritmalarında gizlilik ve koruma elde etmek için kullanılır. HE, bir uygulamanın şifre metninin şifresini çözmeden şifrelenmiş veriler üzerinde bir hesaplama yapm a s ı n ı sağlar. Bu şifreleme türü, bir şifreli metnin *S(M)* teslim edilmesine izin verir.

düz metin mesajı *M ve şifreli metin S(f(M))* işlevi üzerindeki hesaplamayı *M* [18,28] mesajının şifresini çözmeden bir düz metin mesajı *M üzerinde* gerçekleştirir. Bu şifreleme işlemi üç adımda ilerler:

Adım 1: Anahtar üretimi. Bu adımda hem gizli anahtar (sk) hem de açık anahtar (pk) üretilir.

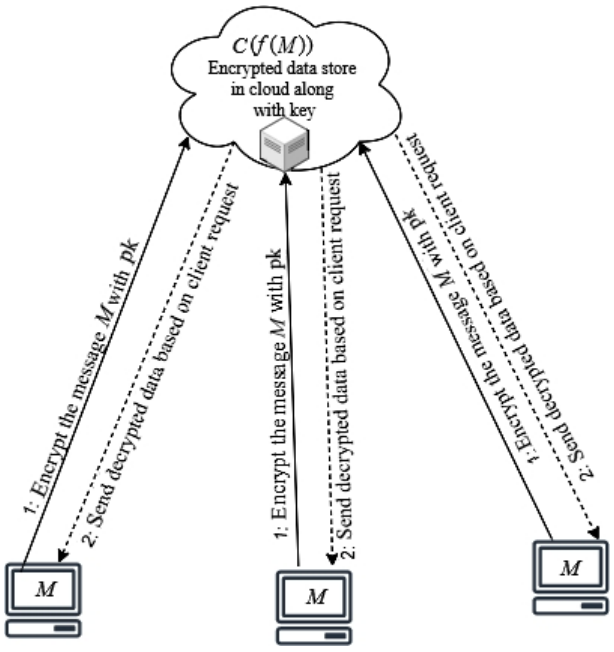
Adım 2: Şifreleme. Düz metin mesajı, şifreli metin üretmek için açık anahtar ile şifreleme gerçekleştirir.

Adım 3: Şifre çözme. Şifrelenmiş mesaj, düz metin mesajı elde etmek için gizli bir anahtar kullanılarak deşifre edilir. Buluttaki istemci ve sunucu arasındaki şifreleme sürecinin adımları Şekil 4'te gösterilmektedir. Adım 1'de istemci, buluta gönderilen şifreleme anahtarıyla birlikte bir düz metin mesajı *M* oluşturur ve ardından şifrelenmiş verileri anahtarla birlikte merkezi veritabanında saklar. İstemcinin bir işlem gerçekleştirmesi gerektiğinde, hizmet sağlayıcıya (bulut) bir talep göndermesi gerekir, ardından hizmet sağlayıcı istemcinin talebini işleme sunucusuna aktarır. Ardından, sunucu isteğe göre *S(f(M))* işlemini gerçekleştirir ve sonucu istemciye bir yanıt adımı olarak döndürür. Son olarak, istemci hizmet sağlayıcıdan (bulut) gelen sonucun şifresini gizli bir anahtarla çözer. HE üç ana kategoriye ayrılır veri üzerinde çalışmasına dayanmaktadır.

(1) Kısmi HE: Toplama veya çarpma işlemlerine izin verir, ancak yalnızca şifrelenmiş veriler üzerinde.

(2) Biraz HE: Bu, şifrelenmiş veriler üzerinde birden fazla işlem gerçekleştirir, ancak sınırlı bir ölçekte değildir.

Şekil 4 Homomorfik şifreleme yönteminin temel yapısı.



(3) Tam HE (FHE): Bu, şifrelenmiş veriler üzerinde işlem yapmak için birçok toplama ve çarpma işlemine izin verecektir.

4.1.1 FHE

FHE, üçüncü bir tarafın girdiyi ve hesaplamanın sonuçlarını bilmeden şifrelenmiş veriler üzerinde keyfi işlevler gerçekleştirmesine olanak tanıyan güvenli hesaplama için önemli bir teknolojik algoritmadır[29] . FHE yönteminde, yüksek dereceli polinomları farklı düşük dereceli polinomlara bölmek için bileşim polinomu kullanılır. Bu prosedürün konsepti, bir düz metin mesajını büyük halkalı bir şifreleme metni ile şifrelemektir. Daha sonra, bu mesaj daha küçük halkalı şifreleme metninde şifrelenmiş düz metinler üzerinde basit bir doğrusal fonksiyon olarak geri kazanılacaktır. Bu nedenle, büyük bir şifreleme metni yerine daha küçük bir şifreleme metninin aktarılması işlem verimliliğini artıracaktır[30] . Veri güvenliği ve gizliliği için güncellenmiş ve yeni bir HE türü olan CKKS'yi kullanıyoruz.

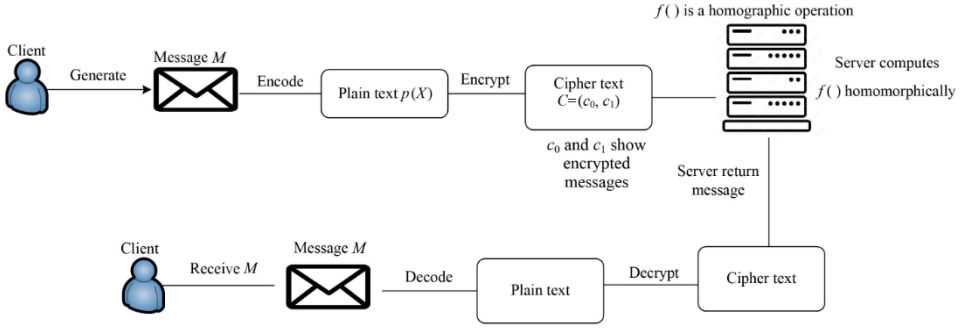
4.1.2 CKKS

Bölüm 3.3.2'de belirtildiği üzere, FHE şifrelenmiş veriler üzerinde hesaplama yapılmasına olanak sağlamaktadır. Araştırmacılar FHE ile veri şifreleme/şifre çözme işleminin hesaplama açısından pahalı olduğunu ileri sürmüşlerdir[31] . CKKS şeması, Şekil 5'te gösterilen Rivest-Shamir-Adleman (RSA)[32] gibi diğer HE şemalarına kıyasla şifreleme ve hesaplamayı hızlı bir şekilde gerçekleştirebilen yeni önerilmiş bir FHE şifreleme şemasıdır. Veri gizliliğini korumak ve FL'nin model eğitim verimliliğini artırmak için makul çözüm CKKS şeması veya HE for Arithmetic Approximate Numbers (HEAAN)[33] .

Bir iletişim sisteminde iki taraf arasında veri alışverişi yapıldığında, aralarında veri güvenliğinin (gizlilik ve bütünlük) garanti edilmesi gerekir. Bunun nedeni, IoT cihazlarında güvenlik güvencesinin gizlilik ve hizmet kalitesine ulaşmada daha önemli olmasıdır[34] . Tam homomorfik CKKS şemasının çalışması birkaç adımdan oluşur. İlk olarak, bilgisayar/istemci tarafından oluşturulan mesaj M, $p(X)$ düz metnine kodlanır, ardından mesajın c_0 ve c_1 açık anahtarları ile şifrelendiği, çeşitli şifreli mesajları gösteren, örneğin iki mesajımız var ve bunlar $S = c_0 + c_1$ şifreli metnine şifrelenmiştir, böylece düz metinden şifreye dönüşmüştür. Daha sonra sunucu, şifrelenmiş veri üzerinde homomorfik bir $f(\ )$ işlemi gerçekleştirmek için mesajı saklar. İstemcinin talebine bağlı olarak, sunucu şifrelenmiş veriyi istemciye iade eder ve mesajın veya verinin şifresini çözmek için ek adımlar (şifrelemenin ters adımları) gerçekleştirir. Daha sonra, istemci gizli anahtarıyla verinin şifresini çözerek $p = f(\ )$ düz metnine dönüştürür, ardından şifreyi çözerek $f(M)$ mesajına veya veriye dönüştürür. HE'nin arkasındaki ana konsept, katılımcıların şifreli metin üzerinde doğru bir şekilde şifre çözme ve şifre çözme işlemlerini gerçekleştirmelerini ve işlemler uygulandıktan sonra çıktıyı sağlamalarını sağlayan kodlayıcı, şifreleyici, şifre çözücü ve kod çözücü gibi HE özelliklerine sahip olmaktır. Önerilen CKKS şeması polinomlarla çalışır çünkü diğer standart hesaplamalara göre güvenlik ve verimlilik arasında iyi bir iletişim vardır. Veri alışverişi sırasında iletişim kanallarını güvence altına almak için kullanılan birçok başka protokol vardır; ancak, düşük gecikme süresiyle iletişim ek yükünü azaltamazlar ve bir istemci ile sunucu arasındaki yineleme turlarını kontrol edemezler. Hafif güvenlik protokolü ve simetrik yük şifreleme, düşük gecikme süresi ile iletişim ek yükünü en aza indirmek ve hesaplama yapmak ve iletişim kapasitesinden tasarruf etmek için kullanılır[35] .

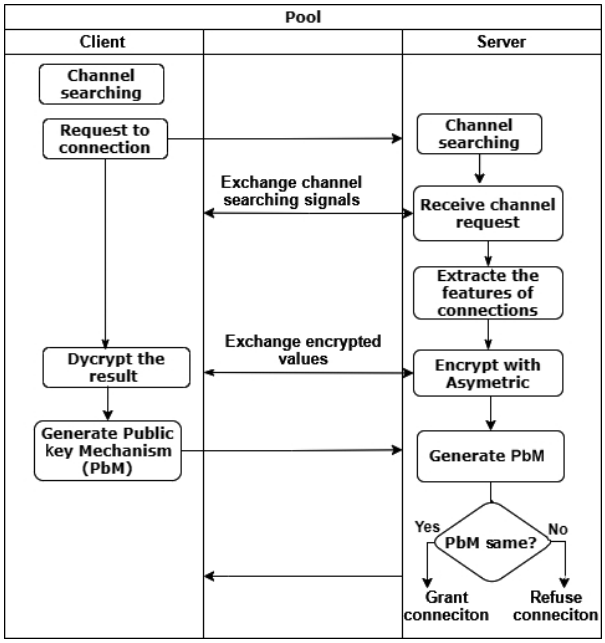
4.2 Hafif güvenli aktarım protokolü

IoT, siber dünyayı fiziksel cihazlarla birbirine bağlayan ve giderek büyüyen bir teknolojidir. IoT teknolojilerini ve uygulamalarını sağlık, eğitim, ulaşım gibi günlük hayatımızın farklı alanlarında bulabiliriz. Tüm IoT tabanlı sistemlerin faydaları vardır, ancak aynı zamanda IoT cihazlarının iletişim maliyeti, güç kapasitesi, hesaplama ve depolama kapasitesi ile ilgili bazı iletişim ve performans zorluklarıyla da karşı karşıyadırlar. Bu nedenle, düşük gecikmeli güvenli iletişim ile yüksek hız



Birçok IoT uygulaması için diğer katılımcılarla zamanında bilgi güvenliği alışverişi sağlamak önemlidir. Bununla ilgili olarak, TLS ve DTLS gibi hesaplama açısından hafif çeşitli protokoller önerilmiş ve uygulanmıştır. Bununla birlikte, mevcut TLS kimlik doğrulama yöntemi, kısıtlı IoT ortamlarında ağır ek yük, yüksek iletişim maliyetleri, gecikme ve diğer güvenlik sorunlarından muzdarip olabilir. Bu protokoller yerine, düşük iletişim gecikmesi ve daha az ek yük ile korunan verileri sunmak için bir iTLS protokolü öneriyoruz [36,37] . Hafif protokol, küçük ve daha yalın bir yüke sahip olan ve bir ağ bağlantısı üzerinden kullanılan ve iletilen bir protokol türüdür. Hafif protokolün özellikleri, istemci/sunucu tabanlı ağ sistemlerindeki diğer iletişim protokollerinden daha hızlı, daha basit ve kullanımı ve yönetimi daha kolay olmasıdır. Hafif protokoller, gerekli olmayan iletişim adımlarını ve verileri dışarıda bırakır veya bir ağ bağlantısını etkileyen veri sıkıştırma tekniklerini kullanabilir. Bu nedenle iTLS, istemci-sunucu bağlantıları için kullanıldığında aynı özelliklere sahiptir, çünkü aynı zamanda bir iTLS protokolü olarak kabul edilir. iTLS protokolü, istemciler tarafından sağlanan alakasız güncellemeleri dinamik olarak belirleyebildiği ve gereksiz verilerin aktarılmasını yasakladığı için iletişim ek yükünü azaltır. Söz konusu protokol ayrıca, ilk bağlantıda istemcileri dinamik olarak tespit etmek ve aktarım protokollerine kıyasla katılımcılar üzerinde güvenlik filtreleme adımları uygulamak için iç tekniklerle donatılmıştır ve aynı zamanda iletişim maliyeti ve gecikme nedeniyle bazı bağlantı adımlarını entegre eder [38,39] . iTLS, açık anahtar ile açık anahtarı sunan varlık arasında doğal bir bağ kurmak için asimetrik şifreleme veya açık anahtar şifreleme mekanizması olarak bilinen kimlik tabanlı kriptografiyi kullanır. IBAKA şeması, paylaşılan bir anahtar oluştururken farklı taraflar arasındaki iletişimin kimliğini doğrular; bu süreçte bağlantı kurulurken iTLS ile artık bir iletişim ve doğrulama sertifikası kullanılmasına gerek kalmaz. iTLS, ilk iletişimdeki verileri korumak amacıyla ilk anahtarı dinamik olarak oluşturmak için her iki tarafın kimliğini kullanır[10] . Kimlik doğrulama akış şeması Şekil 6'da gösterilmektedir. Görülebileceği gibi, bir iletişim kanalı kurmak için farklı adımlar izlenir. Bunlar iki ana aşamaya ayrılır.

- Başlatma: Bu aşama aşağıdakileri başlatır veya arar kanal arama sinyallerini değiş tokuş etmek için bağlantı talebi.
- El sıkışma: Başlatma aşaması tamamlandığında, iTLS el sıkışma protokolü şunları üretir



Şekil 6 Hafif iTLS aktarım protokolünün akış şeması.

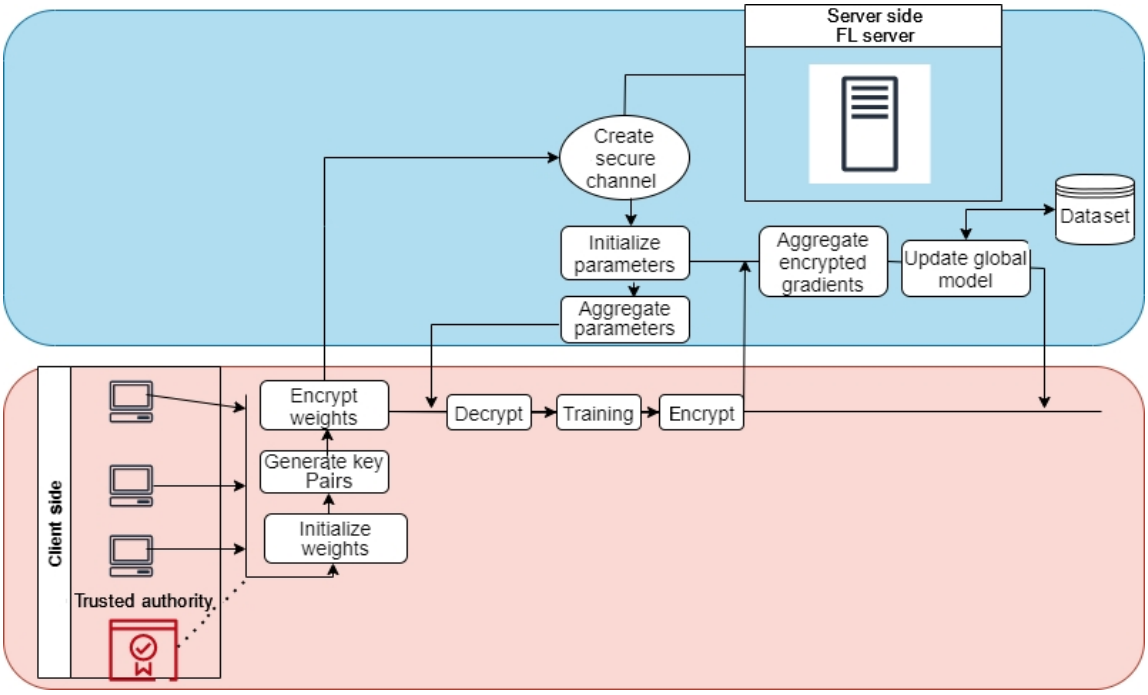
şifreleme anahtarı ve şifreleme değerlerini değiş tokuş eder. Daha sonra iTLS katılımcılar arasında güvenli bir bağlantı kurar.

5 Deney

Bu bölümde, güvenlik ve gizlilik algoritmasını gerçek dünyadaki gibi FL çerçevelerinden biriyle bir IoT ortamında değerlendiriyoruz. Bunu yapmak, istemciler ve koordineli sunucular arasında değiş tokuş edilen bilgilerin bütünlüğünü sağlar. Modeli "MNIST" adlı standart bir veri kümesi üzerinde eğitmek için IID olmayan bir dağıtım (yani Bağımsız Özdeş Dağıtım) türü öneriyoruz ve yerel bilgi gizliliğinin korunması için model parametrelerini şifrelemek üzere tam homomorfik algoritma uyguluyoruz. Ayrıca, model eğitimi sırasında model performansını hızlandırmak, iletişim ek yükünü azaltmak ve gelişmiş iletişim verimliliği için gecikmeyi azaltmak için iTLS protokolünü kullanıyoruz.

5.1 Deneysel ortam

Deneylerimizin amacı, IoT ortamında FL'deki yerel bilgilerindeki model parametreleri için gizlilik koruması sağlarken, istemcilerin ve sunucuların güvenli iletişim kurmasını sağlamak ve iletişim ek yükünü ve gecikmesini azaltmak için hafif protokoller ve FHE algoritmaları desteğiyle Python tabanlı bir FL tanımlamaktır. Mantıksal diyagram, CKKS'ye dayalı istemci-sunucu iletişimini sunan Şekil 7'de gösterilmektedir ve



Şekil 7 CKKS ve iTLS protokolü altında istemci-sunucu iletişiminin mantıksal diyagramı.

iTLS. Bu iletişim, başlatma, güvenli iletişim oluşturma, şifreleme anahtarları oluşturma ve her iki tarafta da model eğitimi (istemci/yerel eğitim ve sunucu/küresel eğitim) gibi çeşitli bölümlerden oluşur. Her bir bölümü aşağıda kısaca açıklıyoruz.

(1) Başlatma: iTLS protokolleri, model güncellemesi için bir mesaj gönderilmesi gerekip gerekmediğini belirlemek üzere başlatılır. Eğer gerekliyse, iTLS protokolleri iletişim katılımcıları arasında güvenli bir kanal oluşturur. Buna ek olarak, güvenilir bir otorite CKKS şifreleme şemasına dayalı bir anahtar çifti (açık anahtar ve gizli anahtar) oluşturur. Her bir istemci kendi yerel modelinin ağırlık parametrelerini (yerel ağırlıklar) başlatır ve birleştirme için merkezi veya bulut sunucusuna göndermeden önce pk genel anahtarı ile şifreler. Parametreler alındığında, sunucu alınan tüm parametreleri toplayarak global model üzerinde hesaplamaya başlar. Böylece, sunucu global modeli, enc (global, ağırlıklar), istemcilere gönderir.

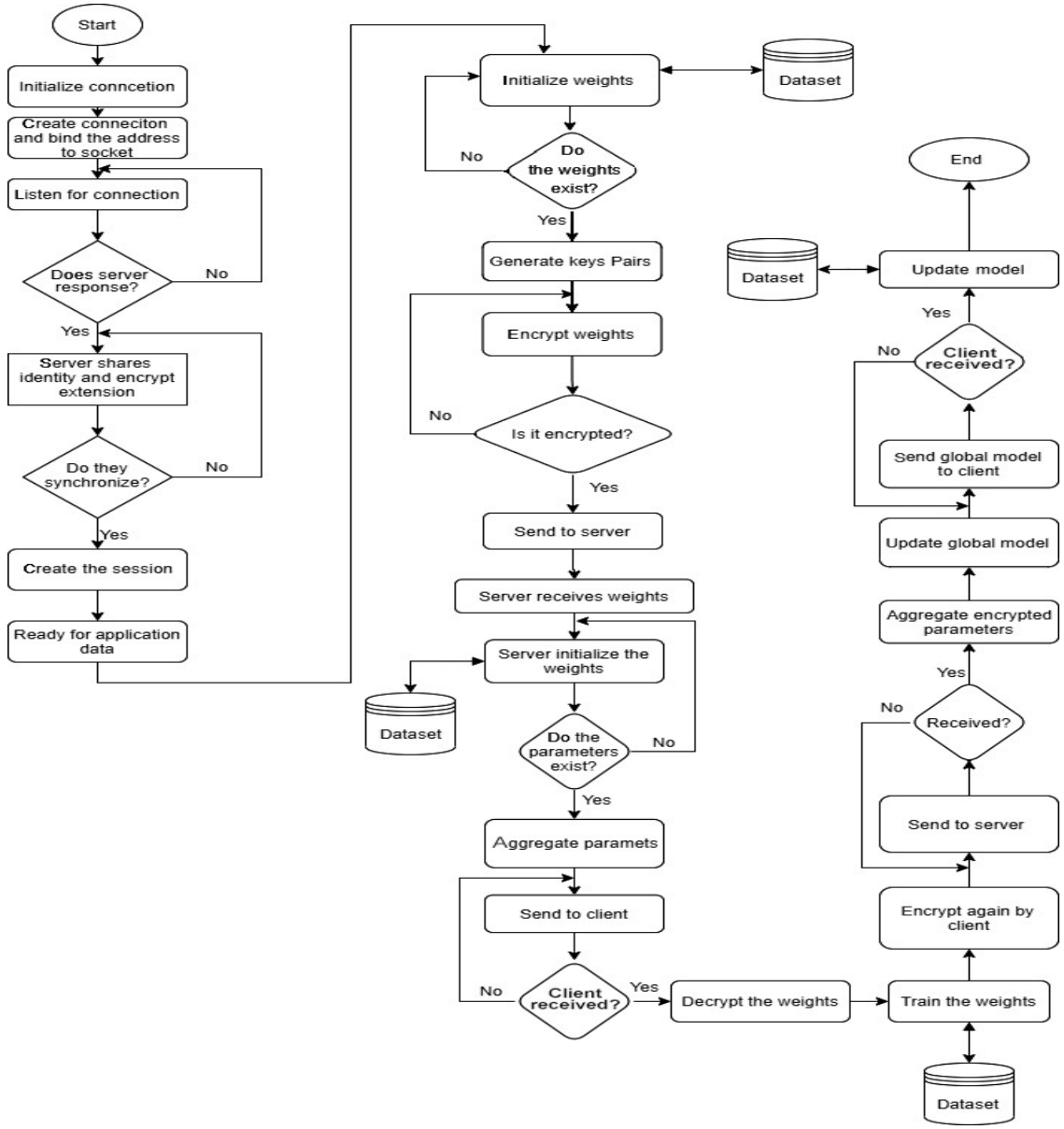
(2) Yerel model eğitimi: Bir istemci şifrelenmiş global model parametrelerini enc (global ağırlıklar) aldıktan sonra, global modeli elde etmek için gizli bir anahtarla (sk) şifre çözme işlemi gerçekleştirir. Ardından, istemci global modeli kendi yerel modeline yükler ve özel veri kaynakları aracılığıyla yerel model gradyanlarını hesaplar. Gradyanların hesaplanması tamamlandığında, model açık anahtarla tekrar şifrelenir ve daha fazla işlem için bulut sunucusuna geri gönderilir.

(3) Global model eğitimi: Merkezi veya bulut sunucusu tüm dağıtık istemcilerden şifrelenmiş gradyanları (yerel ağırlıklar) alır almaz, tüm şifrelenmiş gradyanları hesaplar, global modeli günceller ve ardından tüm dağıtık istemcilere geri gönderir.

5.2 Deneysel yapılandırma ve sonuç

Mantıksal diyagram, Şekil 7'de gösterilen IoT ortamı altında FL istemcileri ve sunucuları arasındaki iletişimi göstermektedir. Bu, iletişim ek yükünü azaltmak ve düşük gecikme süresi ve veri gizliliği (bütünlük ve gizlilik) elde etmek için CKKS ve iTLS uygulamasıyla yapılan deneylerle doğrulanmıştır. Simülasyonun gerçekleştirilmesinde birçok aşama yer almaktadır. Daha iyi bir açıklama için, Şekil 8'de gösterilen sistem yapısındaki ana adımı ve alt adımları göstermek için tüm program akış şemasını tasarlıyoruz. Özellikle, Şekil 8 tüm program yapısının adımlarını ve mantıksal tasarımla ilgili işlevleri göstermektedir. Akış şeması mantıksal tasarımla ilgili iki süreci göstermektedir. Bunlardan ilki iTLS kullanılarak istemci-sunucu bağlantısının kurulmasıdır. Bağlantı kurulduktan sonra, CKKS şifreleme şeması gizliliğin korunması ve güvenlik için verileri ve eğitim modelini şifreleme işlevini başlatır.

Akış şeması, mantıksal diyagramdaki bazı kritik özelliklere göre tasarlanmıştır, Programla ilgili bazı parametreler açıklamalarıyla birlikte aşağıda sunulmuştur



Şekil 8 FL sisteminde CKKS ve iTLS protokolünün uygulanması ile istemci-sunucu iletişim akış şeması.

Tablo 1.

(1) Başlatıcı özniteliği: İlk olarak, hem istemciler hem de sunucu aracıları eğitim modelinin yinelenmesi için başlatılır ve ardından bir model güncellemesinin gerekli olup olmadığını belirlemek için değerlendirme yapılır. Eğer gerekliyse, iTLS katılımcılar arasında bir kanal oluşturmak ve anahtarları birbirleriyle değiştirmek için bayrak mesajı alışverişinde bulunur. Bu aşama aynı zamanda her bir istemci için doğru ve gerekli miktarda veri setini dağıtır. Başlatıcı aşaması oluşturulduktan sonra simülasyonu başlatmak için simülasyon ( ) çağrılır.

(2) Simulation.py ve config.py: Başlatma aşamasından sonra, simülasyon ( ) her ikisini de çağırır istemci ve sunucu aracısı. Burada bir istemci, istemci aracı adları için bir harita oluşturur ve bunu diğer istemcilerle paylaşırken sunucu, _katılan tüm istemcilerden gelen istek değeri ( ) ile çağrılır. Her istemci ağırlıkları oluşturmaya çalışır ve o iterasyon için kendi veri kümesi *D* boyunca öğrenme modelini eğitir. Ardından, eğitilen modelin bir kopyasını güvenlik ofseti ile sunucuya gönderir. Sunucu eğitilmiş modeli aldığında, tüm modelleri toplamaya devam eder ve bunları istemcilere geri gönderir. İstemciler her bir iterasyon için toplam model ağırlıklarını aldıktan sonra, kendi yerel modeline kıyasla federasyon modelinin doğruluğunu hesaplar. Bu faaliyet belirli bir sayıya kadar *yinelenecektir.*

Tablo 1 Parametrelerin yapılandırılması.

Parametre	Açıklama
iTLS protokolü	Bu güvenli bir aktarım iletişimidir. Eğer durumu "True" ise, yeni soketler ve istemci-sunucu Gerçek veri iletimine hazırlanmak üzere birincil bayrak mesajlarını değiş tokuş etmek için bağlantılar kurulacaktır. Aksi takdirde, iletişim devresi oluşturulmayacaktır.
Güvenlik	anahtarı "Doğru" ise, istemci şifreleme yöntemi için ortak anahtarlar oluşturmak üzere simülasyonun çevrimdışı bölümünde diğer katılımcılarla anahtar değişimleri gerçekleştirir.
Diferansiyel Gizlilik (DP) ekleyin	"Doğru" ise, CKKS odak noktasını içeren önemli bitten sonra gürültü eklenmesine izin verir mesaj. Ayrıca, bu şifreleme gürültüsü yaklaşık hesaplama yapılırken bir hata olarak kabul edilir ve böylece güvenlik sertliği varsayımı sağlanmış olur.
DP'yi kaldırın	Eğer "True" ise, FL sisteminde istemci tarafından veri alınırken DP kaldırılacaktır. Aksi takdirde, istemci sunucu tarafından hesaplanan modeli kullanacaktır.
Client	dropped-outEğer "True" ise, FL modelindeki her iterasyondan ağırlıkları aldıktan sonra istemci düşer ve simülasyon istemciler olmadan devam eder.
Simülasyon ve gecikme	"Doğru" ise, sistem kullanıcı tanımlı protokolün her bir adımının ne kadar süreceğini simüle eder. config.py dosyasındaki iletişim gecikmesi alacaktır. Aksi takdirde, bilgiler görüntülenmeyecektir.

yinelemeler.

(3) Mesaj. py: Tüm katılımcılar iletişim kurmalı, birbirlerini çağırmalı ve iletişim için mesaj göndermelidir. Mesaj, iletişim ve gövde nitelikleri hakkındaki tüm meta verileri içerir.

(4) Client-agent.py: Bu, öğrenme modelini eğiten simülasyonun ana örneğidir. Ajan, simülasyon sırasında belirlenecek hem ajan adına hem de numarasına sahiptir.

Sunucu aracıyı çağırdığında, aşağıdaki iki ana yöntemi kullanacaktır:

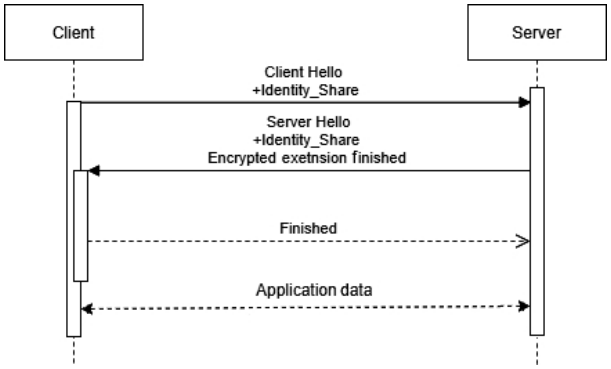
- Ağırlıkları oluşturun: Bu aşamada, istemci öğrenme modelini her iterasyon için kendi veri kümesi üzerinde generate-weights (self, message) kullanarak iletir.
- Ağırlıkları al: Receive-weights (self, message) sunucu, toplanmış federasyon ağırlıklarını istemciye döndürmek istediğinde kullanılır. Bu mesaj, alınan mesajın iterasyon numarasını, ağırlığını ve simüle edilmiş zamanını içerir. İstemcinin ağırlıkları birleştirilmiş ağırlıklara yakınsarsa, yöntem doğru olarak kabul edilir; aksi takdirde yanlış olarak kabul edilir. Bırakılan istemci yöntemi simülasyondaki her iterasyonun sonunda kullanılır ve bu mesaj bırakılan istemcilerin bir listesini ve simüle edilen zamanı içerir.

(5) Sunucu temsilcisi: Daha önce de belirtildiği gibi
sunucu aracısı üçüncü tarafın bir örneğidir ve sadece eğitim modeli ve bilgi alışverişi için tüm katılımcı istemcileri koordine eder. Sunucu aracısı bu faaliyeti gerçekleştirmek için istek değerini ( ) kullanır.

(6) CKKS şifreleme: Bu adım, istemciler diğer katılımcılarla veri veya eğitim modeli alışverişi yapmak istediklerinde başlatma adımından sonra yapılır. CKKS şifreleme, aşağıdakileri sağlamak için kullanılan FHE şemasıdır gizlilik ve bütünlük (güvenlik ve gizlilik). FL sistemindeki çeşitli katılımcılar arasında, özellikle de istemci ve sunucu arasında alışveriş yapan bir şifreleme eğitim modelidir. Bu işlevi yerine getirmek için CKKS şeması iki tür anahtar (gizli ve açık anahtarlar) üretir. İstemci mesajı açık anahtarla şifreler ve sunucuya gönderir, ardından sunucu farklı istemcilerden alınan tüm şifreli mesajları toplar. Bu tür işlemler şifrelenmiş veriler üzerinden yapılır. Bir istemcinin isteğine bağlı olarak, sunucu birleştirilmiş şifrelenmiş verileri istemciye döndürür. İstemci gizli anahtarı kullanarak veri şifresini çözer ve modeli öğrenmeye devam eder. Programla ilgili bazı parametreler açıklamalarıyla birlikte Tablo 1'de sunulmuştur.

5.3 Sonuçlar

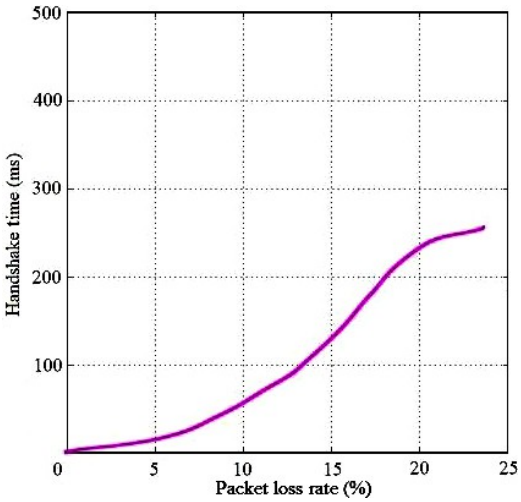
Simülasyon çalıştığında, iTLS ve CKKS algoritmaları yukarıda simülasyonun yapılandırma ortamında açıklanan adımlara göre işlevlerine başlar. Bu noktada iTLS protokolü katılımcılar arasındaki iletişim kanalını sağlar ve istemcilere kimliklerini diğerleriyle paylaşma şansı verir, böylece katılımcılar arasındaki gereksiz iletişim adımlarını azaltır. Şekil 9'd a k i iTLS iletişim sırası diyagramında gösterildiği gibi, iTLS protokolü istemci ve sunucu tarafından bir İstemci Merhaba ve sunucu mesajı altında kullanılan yeni bir uzantı kullanır. Amaç, kimlik bilgilerini ve paylaşılan sırlar için kullanılan kriptografik parametreleri aralarında değiş tokuş etmektir. Buna ek olarak, sertifikalara ve sertifika doğrulama mesajlarına ait tüm bileşenler göz ardı edilir çünkü bunlar



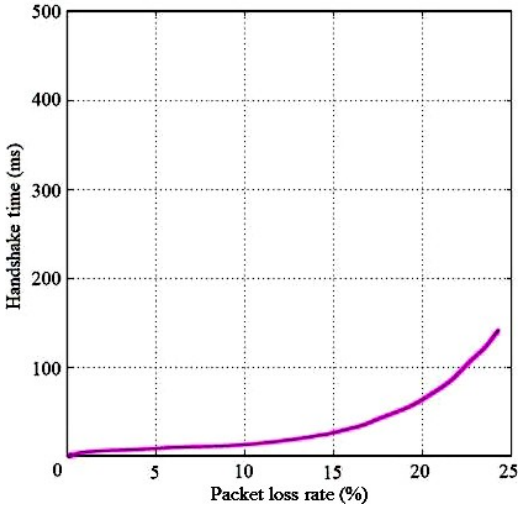
Şekil 9 iTLS protokolü iletişim adımları.

TLS gibi diğer protokoller tarafından değiştirildiğinde daha fazla iletişim adımı ve iletişim ek yükünü artırır. Bu nedenle, iletişim adımlarındaki bu azalma performansı artırır ve iletişim ek yükünü ve gecikmeyi azaltır.

iTLS ve TLS arasındaki gecikme ve gereksiz gidiş-dönüş mesajlarının paket kaybı yüzdesine göre karşılaştırmaları Şekil 10 ve 11'de sunulmuştur. Özellikle Şekil 10, TLS'nin mesaj sayısını artırdığını ve gecikme süresini hızla artırdığını, dolayısıyla istemci ile sunucu arasındaki bağlantıyı derhal etkilediğini göstermektedir. Buna karşılık, Şekil 11'de gereksiz mesajları önleyen ve paket kayıpları ile gecikmeyi azaltan CKKS şemalı iTLS gösterilmektedir.



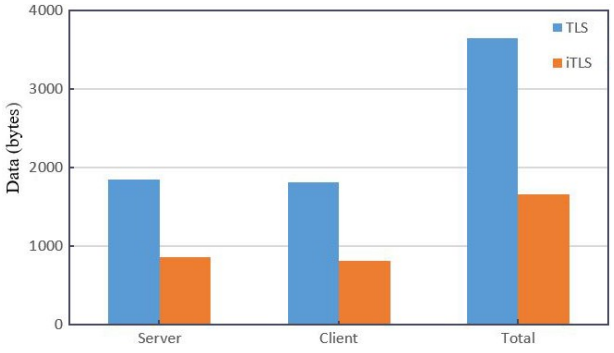
Şekil 10 TLS'de el sıkışma gecikmesi (paket kaybı).



Şekil 11 iTLS'de el sıkışma gecikmesi (paket kaybı).

TLS'nin aksine istemci-sunucu bağlantısının daha az etkilendiği anlamına gelir. Ayrıca, Şekil 11 önerilen şemanın hesaplama sürecini hızlandırdığını göstermektedir. Tablo 2, FL güvenliği ve gizliliği ile hesaplama süreci için kullanılan çeşitli şema ve algoritmaların karşılaştırmasını göstermektedir. Hesaplama süreçleri ve gidiş-dönüş mesajları nedeniyle, farklı bir doğruluk ölçüsüne sahiptirler. Sonuçların da gösterdiği gibi, önerdiğimiz şema gereksiz gidiş-dönüş mesajlarını sınırlamakta, hesaplama sürecini hızlandırmakta ve doğruluğu artırmaktadır.

TLS ve iTLS arasındaki iletişim ek yükünün CKKS şeması protokolü ile karşılaştırılması Şekil 12'de gösterilmektedir. Grafikte gösterildiği gibi, her ikisi için de



Şekil 12 iTLS ve TLS arasındaki iletişim ek yükü karşılaştırması.

Tablo 2 Mevcut ve önceki şema arasındaki doğruluk karşılaştırması.

Şema	Doğruluk (%)	Sebep	Okulu Bırakma
MPC	82.0	Kayan noktalı işlemlerden modüler hesaplamalara dönüşüm nedeniyle azaldı	Evet
DP	77.0	Gürültü ve ek iletişim turları nedeniyle azaldı	Evet
FedAvg/SecAgg	73.0	Ek hesaplama ve iletişim süreçleri nedeniyle daha az doğruluk	-
Bizim planımız	95.3	Ek iletişim turlarının önlenmesi nedeniyle arttı; daha hızlı hesaplama süreci	Hayır

protokollerinin el sıkışması sırasında TLS 1840 bayt üretmiştir. Bunun nedeni iTLS kullanıldığında istemci ve sunucunun herhangi bir ek sertifika bileşeni ve doğrulama mesajı alışverişi yapmasına gerek kalmamasıdır. Daha sonra, iTLS iletişim ek yükünü yaklaşık 65 bayt azalttığı için 850 bayt boyutunda üretirler. Buna ek olarak, CKKS algoritması, model güncellemesi için sunucuya gönderilmesi gereken modelin gradyanlarını şifrelemek için kendi mekanizmasını başlatır, böylece bilgileri potansiyel saldırganlara karşı korur ve bilgi bütünlüğü ve gizliliğinden yararlanma faaliyetlerini azaltır. Bu süreç aynı zamanda model eğitim güvenliğine zarar veren risklerin azaltılmasına da yardımcı olur. Daha önce de belirtildiği gibi, FL sisteminin model eğitim verimliliğini ve güvenliğini artırmak için önerilen yöntem, Şekil 5'te gösterildiği gibi, dağıtılmış istemcinin veri kaynağını korumak ve model parametrelerini şifrelemek için bir CKKS şifreleme şeması kullanır. CKKS şifreleme yöntemi aşağıdaki üç fonksiyondan oluşmaktadır:

(1) Anahtar üretimi: Güvenlik parametreleri ve halka derecesi *n ile* tanımlanır, şifreli metin modülü *P* ile gösterilir. Bu süreçte mesaj kodlanır ve düz metne dönüştürülür. Böylece, güven otoritesi CKKS tarafından tanımlanan kriptosistem standardına dayalı olarak anahtar çifti (pk ve sk) üretir.

(2) Model şifreleme: Bu işlem, güvenilir bir otorite tarafından oluşturulan açık anahtarı kullanarak model parametrelerini şifreler.

(3) Bulut sunucusunda model toplama: Bu adımda, bulut sunucusu tüm dağıtılmış istemcilerden şifrelenmiş mesajları veya gradyanları alır, ardından tüm gradyanları toplar ve küresel bir model oluşturur / günceller.

5.4 Son teknoloji

FL, küresel bir modelin dağıtılmış veri kümeleri üzerinde eğitildiği ve istemcilerin veri gizliliğini artırmak için gerçek veriler olmadan merkezi bir sunucuya göndermeden önce yerel modellerini yalnızca kendi özel veri kümeleri üzerinde eğitmeleri gereken dağıtılmış ve işbirlikçi bir yapay zeka yöntemidir. Bununla birlikte, bağlı cihazların sayısı arttıkça, FL sistemleri yüksek hızlı iletişimin yanı sıra düşük iletişim ek yükü ve gecikmesinin yanı sıra gizlilik, doğruluk ve en iyi hesaplama gücü. Çok sayıda çalışma FL gizliliğini ve iletişim ek yükünü araştırmıştır, ancak hiçbiri FL sisteminin gerektirdiği gibi etkili bir şekilde çalışmamıştır. Bu nedenle, mevcut makale, Tablo 3'te gösterildiği gibi, önceki araştırmaları bizim araştırma planımızla karşılaştırmaktadır.

6 Sonuç

Bu bölümde, bu makalede yürütülen araştırmaya göre varılan sonuç sunulmaktadır. Burada, her ikisi de veri ve model güvenliği ve gizliliğinde önemli rol oynayan iTLS aktarım protokolünü ve CKKS FHE şemasını kullanıyoruz. Bunlar aynı zamanda FL sistemlerinde çeşitli katılımcılar arasında veri alışverişi yapılırken veri işleme performansını artırır, iletişim ek yükünü azaltır ve gecikmeyi azaltır. Veri güvenliğini sağlamak için Gizlilik Bütünlük Kullanılabilirlik (CIA) üçgeninin gerekli uygulaması da bu araştırmada tamamlanmıştır. Bir istemci model güncellemesi için model ağırlıklarını ve gradyanlarını sunucuya göndermek istediğinde, öncelikle model güncellemesinin gerekli olup olmadığını ve bir istemcinin gradyanları sunucuya ne zaman gönderebileceğini belirlemelidir. Böylece, bu çaba katılımcıların istemci ve sunucu arasında aktarılan verilerin kullanılabilirliğini teyit etmelerini sağlar. Ayrıca, ihtiyaç duyulmadığı takdirde veriler periyodik olarak aktarılmayacak, böylece iletişim performansını artırmak ve iletişim ek yükünü azaltmak için bir istemci-sunucu kanalı oluştururken iletişim adımlarının sayısı azaltılacaktır. CKKS şifreleme şeması, modeli şifrelemek için mekanizmasını başlatır, böylece bilgileri potansiyel saldırganlara karşı korur. Bu aynı zamanda model eğitim güvenliği için zararlı olan bilginin bütünlüğüne ve gizliliğine zarar verebilecek bilgisayar korsanlığı faaliyetlerini de engeller. Bu güvenlik ve gizlilik gerçeği ve iletişim performansı, kullanıcıların bulut yapısını tıbbi sistemler, eğitim, bankacılık ve veri güvenliğine ihtiyaç duyan diğer yapılar gibi insan yaşamının farklı alanlarında işlem yapmak için kullanmasına olanak tanır

Tablo 3 Mevcut şemanın referans alınan şema ile karşılaştırılması.

Şema	Doğruluk ve hesaplama süreci	İletişim ek yükü ve gecikme	Operasyon türü	Daha fazla eğitim ve iletişim yinelemesi	Model performansında düşüş
Bizimki	Yüksek (ekleme ve çarpma)	ve hızlı	Daha az Karmaşık	Basit (ya ekleme ve ya	
Diğerleri	Daha az ve daha yavaş[40]		Daha fazla [10;41]		

çarpma [42;46]

Azaltılmış Hayır

İndirgenmemiş [36;44] Evet [9;43]

ve tutarlılık.

Bulut tabanlı yapılarda veri güvenliğini artırmak için, gelecekteki araştırmalar, çeşitli anahtarlar altında şifrelenmiş şifreli metin üzerinde matematiksel işlemleri gerçekleştirebildiğinden, Çok Anahtarlı HE (MKHE) şemasını dikkate alabilir.

Teşekkür

Bu çalışma, Çin Ulusal Anahtar Araştırma ve Geliştirme Programı (No. 2018YFB0803403) ve Çin Eğitim Bakanlığı'nın Merkezi Üniversiteler için Temel Araştırma Fonları (No. FRF-AT-20-11 ve FRF-AT-19-009Z) tarafından desteklenmiştir.

Referanslar

[1] Y. Zhao, J. Zhao, L. Jiang, R. Tan, D. Niyato, Z. Li, L. Lyu, and Y. Liu, Privacy-preserving blockchain-based federated learning for IoT devices, *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1817-1829, 2021.

[2] D. Chen, V. Tan, Z. Lu ve J. Hu, OpenFed: A kapsamlı ve çok yönlü açık kaynaklı federe öğrenme çerçevesi, arXiv ön baskı arXiv: 2109.07852, 2023.

[3] L. Zhang, Z. Zhang, ve C. Guan, Gizliliğin hızlandırılması-preserving momentum federated learning for industrial cyber-physical systems, *Complex Intell. Syst.*, vol. 7, no. 6, pp. 3289-3301, 2021.

[4] B. Jeon, S. M. Ferdous, M. R. Rahman ve A. Walid, Federe öğrenme için gizliliği koruyan merkezi olmayan toplama, *Proc. IEEE INFOCOM 2021-IEEE Conf. Computer Communications Workshops,* Vancouver, Kanada, 2021, s. 1-6.

[5] M. Asad, A. Moustafa, and C. Yu, A critical evaluation of privacy and security threats in federated learning, *Sensors,* vol. 20, no. 24, p. 7182, 2020.

[6] M. Alazab, S. Priya, M. Parimala, P. K. R. Maddikunta, T. R. Gadekallu, ve Q. V. Pham, Siber güvenlik için federe öğrenme: Concepts, challenges, and future directions, *IEEE Trans. Ind. Inform.*, vol. 18, no. 5, pp. 3501-3509, 2022.

[7] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang ve H. Qi, Sınıf temsilcilerini çıkarmanın ötesinde: User-level privacy leakage from federated Learning, in *Proc. IEEE INFOCOM 2019-IEEE Conf. Computer Communications,* Paris, Fransa, 2019, s. 2512-2520.

[8] Y. Aono, T. Hayashi, L. T. Phong, ve L. Wang, Gizlilik-homomorfik şifreleme yoluyla dağıtılmış veri kaynakları ile lojistik regresyonun korunması, *IEICE Trans. Inf. Syst.*, cilt.

E99-D, no. 8, pp. 2079-2089, 2016.

[9] D. Stripelis, H. Saleem, T. Ghai, N. Dhinagar, U. Gupta, C. Anastasiou, G. V. Steeg, S. Ravi, M. Naveed, P. M. Thompson ve diğerleri, Homomorfik şifreleme ile federe öğrenme kullanarak güvenli nörogörüntüleme analizi, arXiv ön baskı arXiv: 2108.03437, 2021.

[10] P. Li, J. Su ve X. Wang, iTLS: IoT için minimum gecikmeli ve mükemmel hafif taşıma katmanı güvenlik protokolü forward secrecy, *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6828-6841, 2020.

[11] R. Hummen, J. H. Ziegeldorf, H. Shafagh, S. Raza ve K. Wehrle, Towards viable certificate-based authentication for the internet of things, in *Proc. 2nd ACM Workshop on Hot Topics on Wireless Network Security and Privacy*, Budapest, Hungary, 2013, pp. 37-42.

[12] M. Hao, H. Li, G. Xu, S. Liu, and H. Yang, Towards efficient ve gizliliği koruyan federe derin öğrenme, *Proc. IEEE Int. Konf. Communications* (*ICC*), Şangay, Çin, 2019, s. 1-6.

[13] T. Li, A. K. Sahu, A. Talwalkar, ve V. Smith, Federated öğrenme: Challenges, methods, and future directions, *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50-60, 2020.

[14] Y. Gao, M. Kim, S. Abuadbba, Y. Kim, C. Thapa, K. Kim, S. A. Camtep, H. Kim, and S. Nepal, Nesnelerin interneti için birleştirilmiş öğrenme ve bölünmüş öğrenmenin uçtan uca değerlendirilmesi, in *Proc. 2020 Int. Symp. Reliable Distributed Systems* (*SRDS*), Şangay, Çin, 2020, s. 91-100.

[15] C. Shen ve W. Xue, Federasyon üzerine bir deney çalışması öğrenme test yatağı, in *Proc. SmartCom 2021*, Singapur, 2021,
s. 209-217.

[16] M. Yang, Y. He, and J. Qiao, Federated learning- based privacy-preserving and security: Anket, *Proc. Computing, Communications and IoT Applications* (*ComComAP*), Shenzhen, Çin, 2021, s. 312-317.

[17] S. Sav, A. Pyrgelis, J. R. Troncoso-Pastoriza, D. Froelicher, J. P. Bossuat, J. S. Sousa, and J. P. Hubaux, POSEIDON: Privacy-preserving federated neural network learning, arXiv preprint arXiv: 2009.00349, 2021.

[18] N. Truong, K. Sun, S. Wang, F. Guitton ve Y. Guo, Gizlilik federe öğrenmede koruma: Insights from the GDPR perspective, arXiv ön baskı arXiv: 2011.05411, 2021.

[19] N. B. Truong, K. Sun, G. M. Lee, ve Y. Guo, GDPR-Uyumlu kişisel veri yönetimi: A blockchain-based solution, *IEEE Transactions on Information Forensics and Security,* vol. 15, pp. 1746-1761, 2020.

[20] J. Konecˇny´, H. B. McMahan, D. Ramage ve P. Richtárik, Federe optimizasyon: Cihaz üzerinde zeka için dağıtılmış makine öğrenimi, arXiv ön baskı arXiv: 1610.02527v1, 2016.

[21] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, ve K. Seth, Practical secure aggregation for privacy-preserving machine learning, in *Proc. 2017 ACM SIGSAC Conf. Bilgisayar ve İletişim Güvenliği*, Dallas, TX, ABD, 2017,
pp. 1175-1191.

[22] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, and H. V. Poor, Federated learning with
diferansiyel gizlilik: Algoritmalar ve performans analizi,
IEEE Transactions on Information Forensics and Security, vol. 15, pp. 3454-3469, 2020.

[23] J. Konecˇny´, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, ve D. Bacon, Federe öğrenme: Strategies for improving communication efficiency, arXiv ön baskı arXiv: 1610.05492, 2017.

[24] M. Asad, A. Moustafa, T. Ito, and M. Aslam, Evaluating the

federasyon öğrenme algoritmalarında iletişim verimliliği, *Proc. IEEE 24th Int. Conf. Computer Supported Cooperative Work in Design*, Dalian, Çin, 2021, pp. 552-557.

[25] Z. Chai, Y. Chen, A. Anwar, L. Zhao, Y. Cheng ve H. Rangwala, FedAT: Asenkron katmanlara sahip yüksek performanslı ve iletişim açısından verimli bir federe öğrenme sistemi, in *Proc. SC21: Int. Conf. High Performance Computing, Networking, Storage and Analysis,* St. Louis, MO, ABD, 2021, s. 1-17.

[26] J. Ma, S. A. Naas, S. Sigg ve X. Lyu, Çok anahtarlı homomorfik şifrelemeye dayalı gizliliği koruyan federe öğrenme, arXiv ön baskı arXiv: 2104.06824v1, 2021.

[27] V. Mugunthan, A. Peraire-Beuno, ve L. Kagal, PrivacyFL: A simulator for privacy-preserving and secure federated learning, in *Proc. 29th ACM Int. Konf. Information and Knowledge Management*, Virtual Event, 2020, s. 3085-3092.

[28] K. Rangasami and S. Vagdevi, Comparative study of homomorphic encryption methods for secured data operations in cloud computing, in *Proc. 2017 Int. Konf. Electrical, Electronics, Communication, Computer, and Optimization Techniques* (*ICEECCOT*), Mysuru, India, 2017, pp. 1-6.

[29] A. Acar, H. Aksu, A . S. Uluagac, and M. Conti, Homomorfik şifreleme şemaları üzerine bir araştırma: Teori ve uygulama, *ACM Comput. Surv.*, vol. 51, no. 4, p. 79,
2018.

[30] M. Ogburn, C. Turner, and P. Dahal, Homomorphic encryption, *Procedia Comput. Sci*., cilt 20, s. 502-509, 2013.

[31] J. He, B Gong, ve J. Yang, ASCFL: Accurate and speedy semi-supervised clustering federated learning, *Tsinghua Science and Technology*, vol. 28, no. 5, pp. 823-837, 2023.

[32] H. Chen, W. Dai, M. Kim ve Y. Song, Kayıtsız sinir ağı çıkarımına uygulama ile paketlenmiş şifre metinleri i 1 e verimli çok anahtarlı homomorfik şifreleme, *Proc. 2019 ACM SIGSAC Conf. Bilgisayar İletişim Güvenliği,* Londra, İngiltere, 2019, s. 395-412.

[33] J. H. Cheon, A. Kim, M. Kim, and Y. Song, Yaklaşık sayıların aritmetiği için homomorfik şifreleme, in *Proc. 23rd Int. Konf. Kriptoloji ve Bilgi Güvenliği Teorisi ve Uygulamaları,* Hong Kong, Çin, 2017, s. 409- 437.

[34] D. Shehada, A. Gawanmeh, C. Fachkha, and H. A. Damis, Performance evaluation of a lightweight IoT authentication protocol, in *Proc. 3rd Int. Konf. Signal Processing and Information Security* (*ICSPIS*), DUBAI, Birleşik Arap Emirlikleri, 2020, s. 1-4.

[35] A. Diro, H. Reda, N. Chilamkurti, A. Mahmood, N. Zaman, and Y. Nam, Lightweight authenticated-encryption scheme for internet of things based on publish-subscribe communication, *IEEE Access,* vol. 8, pp. 60539-60551, 2020.

[36] M. N. Khan, A. Rao ve S. Camtepe, Lightweight cryptographic protocols for IoT-Constrained devices: A survey, *IEEE Internet Things J.,* vol. 8, no. 6, pp. 4132-4156, 2021

[37] P. Liu, X. Xu, ve W. Wang, Tehditler, saldırılar ve savunmalar
federe öğrenmeye: Sorunlar, taksonomi ve perspektifler, *Cybersecurity*, cilt 5, no. 1, s. 4, 2022.

[38] O. Shahid, S. Pouriyeh, R. M. Parizi, Q. Z. Sheng, G. Srivastava, and L. Zhao, Communication efficiency in federated learning: Başarılar ve zorluklar, arXiv ön baskı arXiv: 2107.10996v1, 2021.

[39] L. Wang, W. Wang ve B. Li, CMFL: Hafifletme federasyon öğrenimi için iletişim ek yükü, *Proc. IEEE 39th Int. Konf. Distributed Computing System* (*ICDCS*), Dallas, TX, ABD, 2019, s. 954-964.

[40] J. Loya ve T. Bana, Gizliliği koruyan tuş vuruşu analizi using fully homomorphic encryption & differential privacy, in *Proc. Int. Konf. Cyberworlds* (*CW*), Caen, Fransa, 2021, s. 291-294.

[41] E. Rescorla, The Transport Layer Security (TLS) Protocol version 1.3, Internet Engineering Task Force (IETF), RFC 8446, https://www.rfc-editor.org/info/rfc8446, 2018.

[42] K. Lauter, M. Naehrig, ve V. Viakuntanathan, Can homomorfik şifreleme pratik olabilir mi? in *Proc. Association for Computing Machinery* (*ACM*) CCSW, 2011, s.113- 124.

[43] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, Deep learning with differential privacy, in *Proc. ACM SIGSAC Konf. Bilgisayar ve İletişim Güvenliği,* Viyana, Avusturya, 2016, s. 308-318.

[44] U. Gupta, D. Srtipelis, P. K. Lam, P. M. Thompson, J. L. Ambite, ve G. Ver Steeg, Membership inference attacks on deep regression models for neuroimaging, *Mach. Learn. Res.*, cilt 143, s. 228-251, 2021.

Nasir Ahmad Jalali, Afganistan'daki Kabil Üniversitesi'nden bilgisayar bilimi (bilgi teknolojisi) alanında MEng derecesi almıştır. Çin'deki Pekin Bilim ve Teknoloji Üniversitesi'nde (USTB) doktora adayıdır. 2012'den beri Afganistan'daki Ghazni Üniversitesi'nde yardımcı doçent olarak görev yapmaktadır. Araştırma alanı ağ güvenliği, bilgi güvenliği, kablolu ve kablosuz ağ, makine öğrenimi ve büyük veridir. Beş akademik makale yayınlamıştır. USTB'de 2022 yılında üstün öğrenci ödülünü almıştır. Ayrıca, *MPLS-VPN Impacts on VoIP-QoS* ve *Afganistan'da Yükseköğretim Bilgi Güvenliği için Çerçeve Geliştirme* başlıklı iki kitabın yazarıdır.

Hongsong Chen, 2006 yılında Harbin Teknoloji Enstitüsü, Çin'den bilgisayar bilimleri alanında doktora derecesini almıştır. 2008'den beri Pekin Bilim ve Teknoloji Üniversitesi (USTB) Bilgisayar Bilimleri Bölümü'nde profesör olarak görev yapmaktadır. Purdue Üniversitesi Bilgisayar Bilimleri Bölümü'nde misafir araştırmacı olarak bulunmuştur. Üniversitesi, ABD'de 2013-2014 yılları arasında görev yapmıştır. Çin Bilgisayar Federasyonu'nun üst düzey bir üyesidir. Şu anda IEEE üyesidir. Araştırma alanları arasında yapay zeka ve bilgi güvenliği, kablosuz ağ ve yaygın hesaplama ve güven hesaplama bulunmaktadır. USTB'de 2009 yılında Mükemmel Genç Akademik Makale Ödülü'nü almıştır. 60'tan fazla akademik makale ve 6 kitap yayınlamıştır.