Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9
DOI: https://doi.org/10.24867/24BE23Stanojevic

DISTRIBUIRANA PLATFORMA ZA FEDERATIVNO UČENJE ZASNOVANA NA RAZDELJENOM VIŠESLOJNOM BLOKČEJNU

A DISTRIBUTED PLATFORM FOR FEDERATIVE LEARNING BASED ON A SHARDED BLOCKCHAIN

David Stanojević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom radu će biti predstavljena distribuirana platforma za federativno učenje, zasnovana na razdeljenom blokčejn rešenju. Takođe, biće predstavljene i teorijske osnove koje stoje iza dva glavna aspekta platforme, federativnog učenja i tehnologija distribuirane glavne knjige.*

Ključne reči: *federativno učenje, blokčejn, tehnologija distribuirane glavne knjige, Hyperledger Fabric*

Abstract – *This work will present a distributed platform for federative learning based on a sharded blockchain solution. Furthermore, theoretical background of two main underlying aspects of the platform, the federative learning and distributed ledger technology, will be also presented.*

Keywords: *federative learning, blockchain, distributed ledger technology, Hyperledger Fabric*

1. UVOD

Sa povećanjem pristupačnosti i procesne moći umreženih uređaja poput mobilnih telefona i pametnih senzora, javlja se nova potreba za visokim iskorištenjem ogromne količine podataka koje pomenuti uređaji generišu. Termin koji pokušava da objedini pomenuti problem u praksi je „Internet svega“ [1] (eng. *Internet of Everything ili IoE).* „Internet svega“ podrazumeva distribuirani sistem klijenata, uređaja, podataka i procesa koji su zajedno umreženi i koji svojom participacijom u tom sistemu pružaju vredne i precizne informacije za razne poslovne grupe, industrije ili fizička lica. Centralizovano mašinsko učenje je široko priznato u industriji kao standardna tehnologija pri obučavanju modela za primene u pametnoj industriji (eng. *Smart industry*), pametnim mrežama(eng. *Smart grid*) i inteligentnom transportu(eng. *Intelligent transportation*).

Međutim, centralizovano mašinsko učenje podrazumeva prikupljanje podataka iz velikog broja uređaja u centralno skladište za dalju obradu. Ovo ne samo da povećava opterećenje na mrežu i potencijalna kašnjenja, nego uvodi i rizik od curenja privatnosti i zloupotrebe podataka. Za rešavanje ovih problema, Federativno učenje je predloženo kao obećavajuća paradigma treniranja modela na distribuiran način.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dušan Gajić, vanr. prof.

S obzirom da se federativno učenje oslanja na centralni server za orkestraciju kompletnog procesa učenja, postoji visok rizik od ciljanih napada na server, i on kao takav postaje jedinstvena tačka neuspeha (eng. *single point of failure*). Dodatno, tradicionalni pristup ne može da se odbrani od zlonamernih uređaja koji mogu da šalju netačan lokalni model i time utiču na preciznost globalnog modela. Sa strane efikasnosti, većina arhitektura federativnog učenja rade u sinhronom režimu, i trening će kao takav u većini slučajeva biti usporen potencijalnim zaostalim uređajima. U drugu ruku, puno asinhrono treniranje može uvesti tzv. ustajale modele, odnosno modele koji nisu ažurirani neko vrijeme i koji mogu uvesti nestabilnost u globalni model.

2. DISTRIBUIRANA GLAVNA KNJIGA

Distribuirani sistemi su sistemi autonomnih računara ili komponenti, koje su fizički udaljene, ali koje zajedno komuniciraju i koordinišu svoje akcije putem softvera distribuiranog sistema, koji se najčešće nalazi na nekom od komponenti. Svaka od komponenti, poseduje sopstvene resurse, a rezultati zadataka koje pojedina komponenta dobije se razmenjuju u obliku poruka. Ovo ih razlikuje od paralelnih sistema gdje se resursi najčešće dele između procesa. Na softveru distribuiranog sistema je da koordiniše komponente i podeli poslove na optimalan način, tako da spolja sistem izgleda kao jedna celina.

Jedna od bitnijih karakteristika distribuiranih sistema, koja je od velike važnosti za ovaj rad, jeste otpornost na otkaze i kvarove. Postoje mnoge vrste otkaza na koje utiču na sistem, kao što su prolazni, povremeni ili trajni otkazi. Njihovom pojavom dolazi do pada performansi distribuiranog sistema, vidljivim kroz metrike pouzdanosti i dostupnosti.

Druga stvar kod tradicionalnog federativnog učenja, jeste što ne postoji otpornost na kvarove, pogotovo na kvarove proizvoljnog, odnosno *vizantijskog* tipa [2]. Ovaj tip kvara se najčešće poistovećuje sa malicioznim procesima, iako uglavnom nije moguće pouzdano utvrditi da li je neka akcija bila benigna ili maliciozna. Zbog toga se pravi razlika između kvara usled propuštanja, gde komponenta ne preduzme akciju koja je trebala preduzeti, i kvara usled delovanja, gde komponenta preduzme akciju koju nije trebala preduzeti. U svakom slučaju, nameran kvar, bio on usled propuštanja ili delovanja, predstavlja ozbiljan bezbedonosni problem u sistemu federativnog učenja i često dovodi to pojave „trovanja modela“ (eng. *model poisoning*).

2.1. Blokčejn

Kada mreža dostigne konsenzus o validnosti najnovijeg stanja glavne knjige, transakcija koja je predložila izmenu se finalizira, enkriptuje i koristi kao osnova za sve buduće transakcije. Ovo je princip na kome se grade sve blokčejn (eng. *blockchain*) tehnologije, gde svaki blok izmena sadrži sopstveni vremenski otisak, kao i enkriptovane informacije o prethodnom bloku, time gradeći jedan veliki lanac blokova. Linearna struktura ovih lanaca podseća u mnogo čemu na jednostruko spregnutu listu, uvezanu kriptografskim primitivama kao što je heš (eng. *hash*) funkcija, što ih čini jako otpornim na bilo kakav vid izmene nakon što se blok upiše u lanac. Najpoznatiji predstavnik ove tehnologije je Bitcoin [3], koji uspostavlja konsenzus na osnovu algoritama baziranih na lutriji, poput dokaza posla [4].

2.2. Sistemi otporni na otkaze

Pored algoritama konsenzusa zanovanih na lutriji, postoji posebna klasa algoritama konsenzusa zasnovanih na glasanju. Ova klasa algoritama je od značaja kod *permissioned* blokčejn tehnologija, i biće u posebnom fokusu ovog rada zbog svoje primene u aplikacijama poslovnog (eng. *enterprise*) tipa, u poređenju sa aplikacijama javne prirode koje se zasnivaju na *permissionless* platformama poput *Ethereum*-a. [5] U kontekstu ovih sistema, postoje tri najbitnija tipa otkaza, odnosno kvarova. **Zaustavljajući kvarovi** (eng. *fail-stop*) – kvar se usled pada može pouzdano detektovati unutar neke predefinisane vremenske granice. **Bučni kvarovi** (eng. *fail-noisy*) – kvar se usled pada može eventualno, na kraju detektovati. **Proizvoljni kvarovi** (eng. *fail-arbitrary*) – kvarovi koji generišu odgovore koje ne bi trebali generisati, ali koje drugi procesi se ne mogu detektovati kao netačne. Maliciozni proces mogu čak sarađivati sa drugim procesima da proizvedu namerne netačne odgovore i podrivaju system.

2.2. Raft konsenzus algoritam

Pošto su prvobitni algoritmi za sprečavanje bučnih kvarova, poput *Paxos* [6] algoritma, bili prekomplikovani, odnosno upravljanje njihovim greškama i graničnim slučajevima je dosta teško za ispratiti i implementirati. Stoga je osmišljen *Raft* [7] konsenzus algoritam sa ciljem da se čitav proces pojednostavi što je više moguće. Od tada, Raft postaje široko prihvaćen u mnogim popularnim alatima poput Apache Kafka [8]. Raft, u svojoj osnovi, sastoji se iz dva glavna procesa: izbor lidera (eng. *leader election*) i replikacija loga (eng. *log replication*). **Izbor lidera** – U *Raft*-u postoji samo jedan lider po klasteru. Ostale replike mogu imati ulogu kandidata ili ulogu pratioca (eng. *follower state*). Lider na vremenske intervale šalje znake života (eng. *heartbeats*) ostalim replikama. Ukoliko replike ne prime *heartbeat* neki određeni vremenski period, pretpostaviće da se lider srušio i započeće proces reelekcije lidera. **Replikacija loga** – U početnoj fazi, klijent šalje zahtev za izmenom stanja lideru, lider uloguje izmenu ali je ne komituje. Lider potom šalje predlog izmene replikama, i ukoliko se većina replika složi sa izmenom, lider komituje izmenu u svoj log, obaveštava klijentu da je izmena prihvaćena i na kraju, zahteva od svih replika da takođe komituju izmenu u svoje logove.

3. FEDERATIVNO UČENJE

Federativno (kolaborativno) učenje (eng. *federated learning*, *FL*) predstavlja decentralizovan pristup treniranju *ML* modela, koji podrazumeva više nezavisnih sesija treniranja na udaljenim uređajima, nad njihovim lokalnim podacima. Ovaj pristup treniranja ML modela je u kontrastu u poređenju sa tradicionalnim centralizovanim pristupom gde se svi lokalni skupovi podataka spajaju u jednu trening sesiju. Takođe, ne podrazumeva činjenicu da su svi lokalni uzorci podataka identično distribuirani, što je generalno i priroda kod heterogenih izvora podataka. Federativno učenje adresira kritične probleme poput privatnosti podataka, sigurnosti podataka i prava pristupa podataka. Postoji više strategija federativnog učenja: **Centralizovano federativno učenje** – centralni server se koristi za orkestraciju različitih koraka *FL* i koordinaciju svih čvorova tokom procesa učenja. Server je takođe zadužen za selekciju čvorova na početku procesa učenja i agregaciju dobijenih izmena modela. Kao što se može videti, server ovde može postati usko grlo sistema i potencijalni problem. **Decentralizovano federativno učenje** – svi čvorovi imaju sposobnost da koordinišu sebe putem algoritma konsenzusa i time zajednički dođu do globalnog modela. Ovaj pristup sprečava probleme kao što su jedinstvena tačka neuspeha (eng. single point of failure), pošto se izmene šalju samo između međusobno uvezanih čvorova bez orkestracije centralnog servera. Jedna od mogućih mana ovog pristupa je loš odabir algoritma konsenzusa koji može da preopteretи komunikaciju u mreži i time smanji performanse učenja.

3.1 Nezavisnost i jednaka distribuiranost uzoraka

U većini slučajeva, pretpostavka da su uzorci u svim lokalnim čvorovima nezavisno i jednako distribuiran, ne važi u federativnom učenju. Uzmimo u razmatranje sledeći slučaj, za nadgledani *FL* zadatak sa obeležjima x i labelama y. Statistički model federativnog učenja podrazumeva dva nivoa uzorkovanja: pristup podacima prvo podrazumeva uzorkovanje klijenta $i \sim Q$, iz distribucije svih dostupnih klijenata, a potom uzorkovanje primera $(x, y) \sim P_i(x, y)$ iz lokalne distribucije podataka klijenta i. Neidentične distribucije klijenata, odnosno $P_i \neq P_j$ za različite klijente i, j, možemo karatkerisati na dva najčešća načina. **Iskrivljenje distribucije obeležja** (eng. *covariate shift*) – marginalne distribuicije $P_i(x)$ mogu da variraju po klijentima, čak i ako je $P(y|x)$ deljen[1]. Na primer, kod prepoznavanja rukopisa, različiti klijenti imaju različite načine pisanja istih karaktera. **Iskrivljenje distribucije labela** (eng. *prior probability shift*) – marginalne distribucije $P_i(y)$ mogu varirati po klijentima, čak i ako je $P(x|y)$ isti. Na primer, kada klijenti iz različitih geografskih lokacija imaju drugačije vrste slika ili različite medicinske ustanove uzimaju roentgen snimke sa drugačijim nijansama boja zbog raznih aparata koji postoje.

[1] Pišemo „$\mathcal{P}_i(y|x)$ je deljen" kao skraćenu verziju $\mathcal{P}_i(y|x) = \mathcal{P}_j(y|x)$ za sve klijente i, j

3.2 Algoritmi federativnog učenja

Od velikog izbora algoritama federativnog učenja koji imaju za zadatak agregirati lokalne modele i kreirati validan globalan model na nivou mreže, spomenućemo jedan koji je veoma bitan za ovaj rad. **Federativno uprosečavanje** (eng. *FedAvg*) – je generalizacija *FedSGD* [9]-a, koja dozvoljava lokalnim čvorovima da obave više od jednog paketnog ažuriranja modela nad lokalnim podacima, razmenjujući ažurirane težine (parametre), a ne njihove gradijente (izvode). Razlog za ovakav pristup je, da ako u *FedSGD*-u svi čvorovi počnu od istog stanja (inicijalizacije), onda je uprosečavanje parametara modela identično sa uprosečavanjem gradijenata parametara modela. Dodatno, uprosečavanje težina koje počinju od iste inicijalizacije uglavnom ne spušta performanse tako agregiranog modela.

4. OPIS REŠENJA

Platforma za federativno učenje sastoji se iz dvoslojnog blokčejna u kom se prvi sloj koji se zove *podlanac* (eng. *subchain layer*), sastoji iz višestrukih udeljenih blokčejnova (eng. *sharded subchains*), a drugi sloj ili sloj glavnog lanca (eng. *mainchain layer*) sastoji od jednog blokčejna zasnovanog na algoritmu acikličnog grafa. Za sloj *podlanca* razmotrena je klasična višepristupni scenario za primenu u pametnom IoT-u za podršku IoT uređajima sa slabijim performansama, gde se *ivični čvorovi* (eng. *edge nodes*), odnosno uređaji sa dovoljnim komputacionim resursima, particionišu u višestruke nezavisne grupe koje nazivamo *delovima* (eng. *shards*). Da bi se ispunili zahtevi kontrole pristupa IoT uređaja, za *podlanac* prisvojeno je rešenje konzorcijum blokčejna, poput *Hyperledger*-a. S druge strane, glavni lanac može biti raspoređen na više distribuiranih ivičnih čvorova ili pouzdanih komputacionih platformi zarad validacije transakcija dostavljenih od strane *delova*, odnosno *podlanaca*.

4.1. Sloj uređaja

Ovaj sloj se sastoji iz IoT uređaja koji učestvuju u zadacima vezanih za proces federativnog učenja, kao što su pametni telefoni, vozila, uređaji pametnih kuća i ostalih, koji su odgovorni za održavanje sakupljenih podataka i treniranje lokalnih modela. Dodatno, uređaji imaju obavezu da upakuju ažuriran lokalni model unutar transakcije sa dodatnim metapodacima poput autorizacionih informacija i vremenskog otiska, i takvu transakciju dostave svom *podlancu*.

4.2. Sloj podlanaca

Podlanci su raspoređeni u svakom *delu* kao nezavisni blokčejn, svaki zadužen za koordinaciju IoT uređaja unutar svog dela i završavanje *FL* zadataka u sinhronom režimu. Zbog toga, *Raft* konsenzus protokol je uveden u svakom *podlancu*, radi visokih performansi i opsega komunikacije potrebnog unutar svakog *dela*. Posledica toga je da ivični čvorovi unutar svakog dela spadaju u jednu od dve kategorije. **Lider čvor podlanca** (eng. *subchain leader node, SLN*) – čvor koji je izabran za lidera po principu Raft algoritma. Pored standardnih operacija uspostavljanja konsenzusa među čvorovima, *SLN* je takođe odgovoran za odabir uređaja koji će vršiti treniranje i autorizovati im pristup *podlancu*. Takođe, *SLN* mora da agregira lokalne modele i dostavi ažuriran model na nivou dela na kraju iteracije glavnom lancu, kao i kreiranje osnovnog iteracionog modela dobavljenog od glavnog lanca na početku nove iteracije. **Pratioc čvor podlanca** (eng. *subchain follower node, SFN*) – čvor koji ima zadatak da validira autentičnost kao i preciznost lokalnih modela dostavljenih kroz transakciju od strane IoT uređaja, i prosledi validne transakcije *SLN*-u. Takođe, zadužen je zajedno sa ostalim čvorovima pratiocima unutar jednog *dela* da uspostavi konsenzus o bloku izgenerisanom od strane *SLN*-a.

4.3. Sloj glavnog lanca

Asinhroni konsenzus mehanizam baziran na arhitekturi direktnog acikličnog grafa, odnosno tangle konsenzusa, prihvaćen je u glavnom lancu za svrhu rukovanja inter-akcijama sa *podlancima*. U DAG glavnom lancu, čvorovi grafa predstavljaju transakcije (blokove), dok ivice grafa označavaju odobrenje neke druge transakcije. Svaka transakcija predstavlja jedan model istreniran od strane jednog *dela*. Transakcije koje nisu odobrene ni od jedne druge transakcije zovu se *vrhovi*. Za razliku od drugih blokčejn sistema, kao što je to slučaj sa PoW-baziranim blokčejnovima, glavni lanac se ne oslanja na jedan lanac koji predstavlja jedinstveni izvor stanja, upravo zbog njegove strukture grafa. Glavni lanac koji inherentno toleriše grananja lanca (eng. *forks*), u mogućnosti je procesirati transakcije asinhrono. Stoga, ova platforma ima sposobnost efektivnog skaliranja bez prevelikog uticaja na performanse sistema: sve što novi IoT uređaj treba da uradi jeste da se pridruži postojećem *delu*, ili eventualno da se uskladi sa drugim ivičnim uređajima da kreira novi *deo*.

4.4. Proces federativnog učenja

Platforma zbog svoje višeslojne arhitekture, podržava i sinhoro i asinhrono učenje, što je čini veoma efikasnom i skalabilnom. Stoga, uvode se dva tipa transakcija, trans-akcija *podlanca* i transakcija glavnog lanca. Transakcija *podlanca* se kreira od strane IoT uređaja i *SLN*-ova i koristi se unutar jednog *dela*. Transakcija glavnog lanca je kreirana od strane *SLN*-a i sadrži agregiran model čitave iteracije *dela* i u upotrebi je unutar glavnog lanca. Faze koje detaljnije opisuju ovaj proces su sledeće:

5.4.1. Faza 1: Objavljivanje zadatka

Glavni lanac na zahtev spolja, započinje FL zadatak, tako što analizira sve nepotvrđene transakcije, odnosno *vrhove*, i bira podskup vrhova koji potom filtrira i sortira na osnovu preciznosti njihovih agregiranih modela. Isfiltrirane transakcije grupiše zajedno i agregira njihove parametre modela i kreira osnovi iteracioni model, koji spakuje u transakciju i pošalje je zainteresovanim podlancima koji su spremni za sledeću iteraciju treniranja. Ukoliko ne postoje *vrhovi*, kreira se početna (eng. *genesis*) transakcija sa nasumičnim parametrima.

5.4.2. Faza 2: Treniranje unutar delova

Transakcija se dovlači od strane SLN-a, koji izvlači korisne informacije poput parametara modela, broja epoha, trenutne preciznosti, i čuva ih unutar distribuirane glavne knjige. Potom, za svaku trening rundu, SLN bira kandidate među IoT uređajima na osnovu kvaliteta njihovih performansi, gde će oni uređaji sa optimalnim stanjem baterije i kvalitetom mreže biti izabrani za trening rundu. Odabrani uređaji dobijaju autorizaciju da mogu

skinuti osnovi model trenutne runde sa glavne knjige i prosleđuju svoje lokalne podatke. Važna napomena je da se modeli čuvaju van lanca (eng. *off-chain*), u obliku heš vrednosti koja se koristi kao identifikator za dobavljanje datoteke modela skladištene na IPFS distribuiranom skladištu. Ovaj način čuvanja je neophodan zbog očiglednih problema sa performansama s obzirom da modeli mogu imati veličinu od par stotina megabajta do par gigabajta, u zavisnosti od odabranog algoritma dubokog učenja.

Na osnovu osnovnog modela runde dobijenog od podlanca, svaki uređaj će obučavati svoje lokalne modele nad sopstvenim podacima. Nakon što dostigne, ili broj epoha ili određenu ciljnu vrednost konvergencije metrike evaluacije, uređaj šalje lokalni model nazad nekom od čvorova *podlanca*. Nakon što su svi lokalni modeli primljeni, čvorovi *podlanca* validiraju tačnost dobijenih modela na osnovu testnog skupa podataka. *SLN* će zatim agregirati validne lokalne modele na osnovu *FedAvg* algoritma, i zatim objaviti *podlancu* kao novi osnovni model runde. Čitav proces se ponavlja dok se ne dostigne broj rundi zadat u metapodacima zadatka, odnosno transakcije glavnog lanca.

5.4.3. Faza 3: Kreiranje transakcije glavnog lanca

Kada se izvrši zadati broj rundi treniranja, najnoviji osnovni model se pakuje unutar transakcije glavnog lanca i šalje glavnom lancu od strane *SLN*-a.

6. ZAKLJUČAK

U ovom radu predložena je višehijerarhijska platforma za federativno učenje, zasnovana na blokčejn tehnologiji. Platforma značajno poboljšava performanse i sigurnost FL sistema unutar inherentno nepoverljive okoline ivičnih uređaja. Implementirana je razdeljena arhitektura za paralelizam između blokčejnova, zarad mnogo bolje skalabilnosti sistema. Unutar *podlanaca*, implementiran je Raft konsenzus algoritam za koordinaciju naučenih lokalnih modela, dok je na glavnom lancu implementiran konsenzus baziran na direktnom acikličnom grafu, što omogućava istovremenu sinhronost i asinhronost sistema, efektivno uklanjajući probleme ustajalih modela i zaostalih uređaja. Štaviše, robusnost platforme na napade tipa „trovanja modela“, od strane zlonamernih čvorova, poboljšana je višehijerarhijskim konsenzusom na oba sloja.

Za buduće radove, u planu je istraživanje ka algoritmu modela baziran na arhitekturi transformera [10], zarad poboljšanih performansi usred nebalansiranih skupova podataka, kao i implementacija dubljih verzija neuronskih mreža zarad dostizanja veće tačnosti na težim skupovima podataka poput CIFAR10 [11], a pritom očuvavajući postojeće performanse na jednostavnijim modelima. Takođe, u razmatranju su razne optimizacije na nivou modela, poput odabira normalizacije podataka i odabira funkcije gubitka, više prikladne federativnom pristupu učenju modela.

7. LITERATURA

[1] “Internet of Everything,” [Online]. Available: https://ioe.org/.

[2] R. S. M. P. Leslie Lamport, “The Byzantine Generals Problem,” 1982.

[3] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008.

[4] A. Back, “Hashcash - A Denial of Service Counter-Measure,” 2002.

[5] V. Buterin, “Ethereum Whitepaper”.

[6] L. Lamport, “The Part-Time Parliament,” May 1998.

[7] J. O. Diego Ongaro, “In Search of an Understandable Consensus Algorithm”.

[8] Apache Foundation, “Apache Kafka,” [Online]. Available: https://kafka.apache.org/.

[9] Wikipedia, “Stohastic Gradient Descent,” [Online]. Available: https://en.wikipedia.org/wiki/Stochastic_gradient_descent.

[10] Y. L. Y. C. Ze Liu, “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows”.

[11] A. Krizhevsky, “The Cifar-10 dataset,” [Online]. Available: https://www.cs.toronto.edu/~kriz/cifar.html.

Kratka biografija:

David Stanojević rođen je u Banjaluci 1997. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Softversko inženjerstvo i informacione tehnologije, odbranio je 2020.god.

Kontakt: davidstanojevic43@gmail.com