Зборник радова Факултета техничких наука, Нови Сад

UDK: 004.9
DOI: https://doi.org/10.24867/29BE22Stanisic

# ПРИМЕНА ФЕДЕРАТИВНОГ УЧЕЊА У РЕШАВАЊУ ПРОБЛЕМА БИНАРНЕ КЛАСИФИКАЦИЈЕ

# IMPLEMENTING FEDERATED LEARNING FOR BINARY CLASSIFICATION SOLUTIONS

Андрија Станишић, *Факултет техничких наука, Нови Сад*

**Област – ЕЛЕКТРОТЕХНИЧКО И РАЧУНАРСКО ИНЖЕЊЕРСТВО**

**Кратак садржај –** *У овом раду је представљено решење које користи федеративно учење за решавање проблема бинарне класификације. Првенствено су анализиране теоријске основе дистрибуираних система и федеративног учења, након чега је представљена имплементација система за федеративну обуку неуронских мрежа. Развијени систем омогућава доделу и агрегацију тежина модела на више дистрибуираних чворова, чиме се елиминишу ризици повезани са централизованим складиштењем података. Рад обухвата преглед коришћених технологија, опис структуре федеративног система, као и евалуацију модела. Овај приступ пружа значајан допринос разумевању федеративног учења и може послужити као основа за даља истраживања и развој напреднијих дистрибуираних система.*

**Кључне речи:** *дистрибуирани системи, бинарна класификација, федеративно учење, неуронске мреже*

**Abstract** – *This paper presents a solution that employs federated learning to address binary classification problems. The theoretical foundations of distributed systems and federated learning are primarily analyzed, followed by the implementation of a system for federated neural network training. The developed system enables the allocation and aggregation of model weights across multiple distributed nodes, thereby eliminating risks associated with centralized data storage. The paper includes a review of the employed technologies, a description of the federated system's structure, and an evaluation of the model. This approach significantly contributes to the understanding of federated learning and can serve as a foundation for further research and the development of more advanced distributed systems.*

**Keywords:** *distributed systems, binary classification, federated learning, neural networks*

## 1. УВОД

Савремени свет је незамислив без рачунара, који су од средине 20. века доживели експоненцијални раст захваљујући микропроцесорима и рачунарским мрежама. Локалне мреже (енгл. *local-area networks - LAN*), метрополитанске мреже (енгл. *metropolitan-area networks - MAN*) и мреже широког подручја (енгл. *wide-area networks - WAN*) омогућиле су напредак у телекомуникацијама, образовању, здравству и финансијама.

Овај развој је створио основу за сложене системе који ефикасно управљају и размењују информације на глобалном нивоу. Данас, један од најузбудљивијих трендова у развоју рачунарских система је свакако вештачка интелигенција. Алгоритми вештачке интелигенције омогућавају рачунарима да уче из података и самостално доносе одлуке. Многе индустрије почињу да користе вештачку интелигенцију за извршавање послова који могу варирати од релативно једноставних до изузетно комплексних. Међутим, ништа од овога не би било могуће да претходно није развијена моћна рачунарска инфраструктура која чини базу вештачке интелигенције. Велики удео ове инфраструктуре чине савремени дистрибуирани системи (енгл. *distributed systems*). Дакле, системи чије се компоненте налазе на различитим, неретко географски удаљеним рачунарима који међусобном комуникацијом чине кохерентну целину. Поред неопходне компутационе снаге, алгоритми вештачке интелигенције захтевају и мноштво релевантних података како би креирали примењиве производе. Међутим, велики обим података и законска ограничења довели су до потребе за новим методама обраде и складиштења података.

Федеративно учење (енгл. *federated learning - FL*) представља револуционарни приступ који омогућава обуку неуронских мрежа на децентрализованим скуповима података. Овај приступ смањује ризике и трошкове повезане са преносом и складиштењем података на централизованим серверима, омогућавајући ефикаснију употребу података и ресурса. Федеративно учење такође омогућава коришћење различитих варијација података који су раније били недоступни због законских и техничких ограничења.

Циљ овог рада је да кроз практичну имплементацију објасни најбитније аспекте федеративног приступа у обучавању неуронских мрежа. Разматрају се кључни аспекти дистрибуираних система и федеративног учења, њихове предности и недостаци, као и технологије и методе коришћене у имплементацији система за бинарну класификацију. На крају, рад

**НАПОМЕНА:**
**Овај рад проистекао је из мастер рада чији је ментор био др Душан Гајић, ванр. проф.**

предлаже правце за даљи развој овог иновативног приступа.

## 2. ТЕОРИЈСКЕ ОСНОВЕ

Савремени рачунарски системи се посматрају као колекција рачунара међусобно повезаних у рачунарску мрежу. Почетна идеја умрежавања рачунара произашла је из потребе за повезивањем постојећих система ради побољшања перформанси и ефикасности. Први приступ умрежавања је интегративни, где се постојећи системи повезују како би формирали моћан систем који омогућава колаборативно решавање комплексних проблема. Други приступ је проширујући, који укључује додавање нових рачунара постојећим системима, чиме се постиже боља отпорност на отказе и повећава доступност.

Централизовани системи имају једну тачку контроле и управљања, што олакшава одржавање и обезбеђује високу поузданост. Међутим, како се комплексност задатака повећава, прелазак на дистрибуиране или децентрализоване системе постаје неопходан.

Дистрибуирани системи распоређују процесе и ресурсе на више рачунара, што омогућава већу поузданост, скалабилност и отпорност на отказе. Децентрализовани системи, пак, елиминишу централну тачку контроле, при чему сваки чвор у мрежи има значајну улогу у обављању функција система, обезбеђујући висок степен сигурности и независности од појединачних тачака отказа.

### 2.1. Основни принципи дистрибуираних система

Дистрибуирани систем је систем у којем компоненте егзистирају на мрежно повезаним рачунарима, који су неретко и географско удаљени, док је целокупна комуникација и координација система заснована на међусобној размени порука [1]. Из ове дефиниције могуће је издвојити неколико значајних карактеристика дистрибуираних система:

- *Конкурентност* - Дистрибуирани системи омогућавају истовремено, ефикасно и безбедно коришћење ресурса од стране више корисника.

- *Независни отказ компоненти* - Отказ једног дела система не сме утицати на остале делове. Систем треба да настави рад без прекида, аутоматски откривајући и отклањајући отказе.

- *Непостојање глобалног сата* - Географска удаљеност чворова онемогућава коришћење глобалног сата. Стога се за синхронизацију користе логички сатови и размена порука.

### 2.2. Класификација дистрибуираних система

У односу на начин пројектовања и њихову намену дистрибуирани системи се могу поделити на три основне категорије: дистрибуирани рачунарски системи високих перформанси (енгл. *High-performance distributed computing*), дистрибуирани информациони системи (енгл. *Distributed information systems*) и прожимајући системи (енгл. *Pervasive systems*) [3].

- *Дистрибуирани рачунарски системи високих перформанси* - Ови системи су дизајнирани за решавање захтевних проблема који захтевају употребу мултипроцесорских рачунара са заједничким меморијским простором. Захваљујући дистрибуцији процеса и ресурса, они могу извршавати комплексне рачунарске задатке са високом ефикасношћу. Ова категорија обухвата грид рачунаре, кластере и рачунарство у облаку.

- *Дистрибуирани информациони системи* - Ова класа дистрибуираних система повезује велики број мрежно увезаних апликација ради ефикасног управљања подацима. Углавном се састоји од сервера и база података, са изазовом интероперабилности.

- *Прожимајући системи* - Интегрисани у свакодневни живот, прожимајући системи комбинују карактеристике децентрализованих и дистрибуираних система. Пример ових система су сензорске мреже, свеприсутни рачунарски системи, као и мобилни рачунарски системи.

### 2.3. Архитектуре дистрибуираних система

За успешну реализацију било ког дистрибуираног система неопходно је одредити његову софтверску и хардверску архитектуру. Софтверска архитектура (енгл. *software architecture*) се односи на логичку апстракцију компоненти система. Она дефинише начин представљања компоненти и њихову међусобну комуникацију. Са друге стране, хардверска архитектура (енгл. *hardware architecture*) одговорна је за распоређивање софтверских компоненти на физичке рачунаре чиме се постиже иницијализација дистрибуираног система.

Архитектура софтвера представља структуру дистрибуираног система у смислу одвојено специфицираних компоненти и њихових међусобних односа. Главни циљ је да се обезбеди да структура испуни садашње и будуће захтеве. Кључни аспекти су поузданост, управљивост, прилагодљивост и економичност система [1]. Неке од најзначајнијих архитектура обухватају:

- *Клијент-сервер архитектура* - Основни елементи су клијент и сервер, где сервер пружа услуге, а клијент их потражује слањем захтева. Ова архитектура је изузетно примењива и једноставна за имплменетацију, али има проблеме са скалабилношћу и јединственом тачком отказа.

- *P2P мреже* - Централна идеја ових система је равноправност и равномерно укључивање свих процеса у решавање одређеног задатка. Сви чворови покрећу исти програм и нуде идентичне интерфејсе, што омогућава отпорнију и скалабилнију обраду.

Дистрибуирани системи играју кључну улогу у обради великих скупова података, омогућавајући паралелну обраду и обуку комплексних модела машинског учења. Њихова способност паралелизације убрзава процес обраде и повећава поузданост, јер

отказ једног сервера не утиче на целокупан систем. Скалабилност омогућава додавање нових сервера како расту потребе за обрадом. Осим тога, дистрибуирани системи омогућавају рад са хетерогеним подацима на различитим локацијама, интегришући их без премештања на централизовани сервер, чиме се подстиче колаборација и иновација у машинском учењу.

### 2.4. Федеративно учење

Федеративно учење се ослања на дистрибуирану архитектуру обуке у којој је процес тренирања модела подељен између *K* клијентских уређаја са локалним подацима, а координише га централни сервер. Локалност података побољшава њихову безбедност и минимизује трошкове складиштења. Осим тога, примена федеративног учења омогућава бољу искоришћеност разноликих података, што доводи до креирања робуснијих и прецизнијих модела.

Процес обуке обухвата више федеративних рунди које укључују локално рачунање градијената модела на клијентским уређајима и глобалну агрегацију на централном серверу. На почетку сваке рунде *t*, случајно се бира скуп од *C* клијената којима сервер шаље глобално стање алгоритма, односно тренутне параметре глобалног модела које су означене са $\omega_t$ . Сваки изабрани клијент локално рачуна нове параметре модела на основу глобалног стања и приватног скупа података. Клијенти затим шаљу измене назад централном серверу који их агрегира у ново глобално стање.

Процес агрегације параметара локалних модела на централном серверу назива се федеративна оптимизација. Проблем федеративне оптимизације може се математички формулисати на следећи начин:

$$\min_{w \in R^d} f(w) \text{ где је} f(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$$

Типично, за проблем машинског учења, функција циља се може дефинисати као $f_i(\omega) = L(xi\ ,\ yi\ ;\ \omega)$, што представља функцију губитка модела на основу узорка ($x_i$ , $y_i$ ) из локалног скупа података клијента и параметара модела ω [2].

Уз претпоставку да постоји *K* клијената са локалним подацима, целокупан скуп података величине *n* је подељен на партиције при чему партиција $P_k$ представља скуп индексираних података на клијенту *k*. С тим на уму, функција циља федеративне оптимизације може се реформулисати на следећи начин:

$$f(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w) \text{ где је} F_k(w) = \frac{1}{n_k}\sum_{i \in P_k} f_i(w)$$

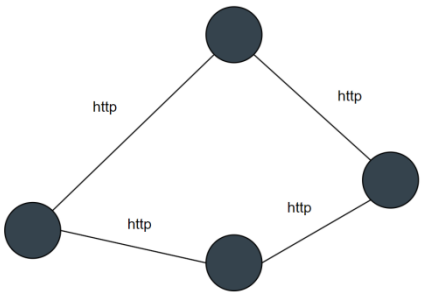
У овом изразу $F_k(\omega)$ представља локалну функцију циља клијента *k*. Ова функција обухвата рачунање функције циља за сваки појединачни податак у скупу података клијента *k*. Вредност $n_k$ представља величину скупа података $P_k$ . Тежински фактор $n_k$ / *n* одређује допринос клијента *k* глобалној функцији циља у складу са величином његовог скупа података.

## 3. ИМПЛЕМЕНТАЦИОНЕ ТЕХНОЛОГИЈЕ

У процесу имплементације коришћени су програмски језици R, Python и Go, сваки са специфичним разлозима. Програмски језик R је коришћен за иницијалну визуализацију и припрему података због своје изврсне подршке за статистичко рачунање и напредну манипулацију подацима, што је омогућило ефикасну обраду и анализу података. Python је изабран за развој и тестирање алгоритама машинског учења захваљујући својој флексибилности и богатом екосистему библиотека као што су NumPy, Pandas и TensorFlow, који омогућавају лако и брзо креирање и експериментисање са моделима. Програмски језик Go је коришћен за имплементацију серверских компоненти и комуникацију између чворова захваљујући својој способности за паралелну обраду великих количина података, што је обезбедило високе перформансе и поузданост система. Ове технологије су заједно омогућиле ефикасну и прецизну обраду података, испуњавајући комплексне захтеве система федеративног учења.

## 4. ИМПЛЕМЕНТАЦИЈА ФЕДЕРАТИВНОГ СИСТЕМА

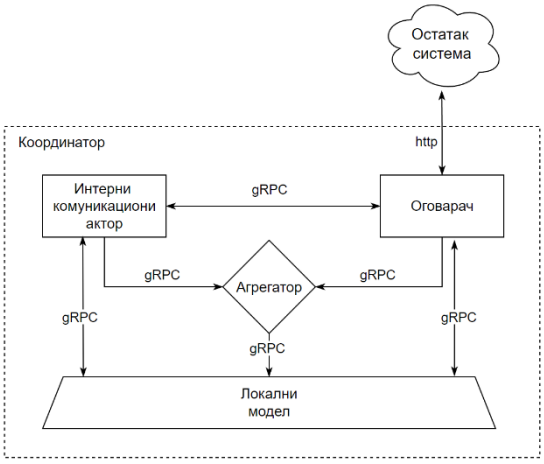
Имплементирани систем је заснован на федеративном учењу. Централна идеја система је извршавање бинарне класификације над улазним подацима. Конкретно, систем на основу обележја, која су у овом случају симптоми пацијиената, даје предикцију о томе да ли је пацијент заражен вирусом или не. Систем је имплементиран као скуп чворова који међусобно комуницирају слањем HTTP захтева. Архитектура система, приказана на слици 1, представља класичну P2P мрежу.




Слика 1 – Архитектура система

Сваки чвор је имплементиран као колекција актора који међусобно комуницирају путем gRPC комуникације. Актори може бити комуникациони, агрегатор или координатор. Додатно, у оквиру сваког чвора налази се имплементација Flask сервера који покреће неуронску мрежу и нуди интерфејс за добављање кључних података везаних за стање модела. Имплементирана неуронска мрежа представља облик вишеслојног перцептрона (енгл. *Multilayer Perceptron - MLP*). Поред пропагације унапред, имплементирана је и пропагација уназад (енгл. *backpropagation*), ради постизања бољих

перформанси . Интерна структура чвора приказана је на слици 2, док се у наставку налази опис најзначајнијих каракеристика ове структуре.




Слика 2 – Интерна структура чвора

**1. Координатор** је задужен за дефинисање тока реализације федеративне обуке. Он посредује у интерној комуникацији између актора, управља креирањем и уништавањем актора у зависности од параметара као што су обим посла и оптерећеност мреже. У контексту система, координатор осигурава синхронизовани тренинг модела, складишти међурезултате обуке и иницијализује тежине модела насумичним вредностима које дистрибуира свим чворовима .

**2. Агрегатор** је одговоран за прикупљање и комбиновање тежина модела са различитих чворова у систему. Он прима локалне моделе са клијентских уређаја, агрегира их применом агрегационе функције и ажурира глобални модел. Ажуриране тежине модела се затим пропагирају назад чворовима за наредну епоху обуке. Агрегатор осигурава да модели буду правилно синхронизовани и ажурирани током целокупног процеса обуке.

**3. Комуникациони актор** је кључна компонента федеративног учења, одговоран за координацију и комуникацију током процеса обуке модела. Постоје два типа актора: оговарач и интерни актор. Оговарач омогућава размену информација између чворова система. Шаље и прима информације о тежинама модела и другим подацима, омогућавајући синхронизовану обуку модела. Интерни актор комуницира са сервером који обрађује податке и врши обуку модела. Функционише као интерфејс између чворова и сервера модела, прикупља тежине и бајасе из неуронске мреже и шаље их агенту агрегатору.

## 5. ЕВАЛУАЦИЈА МОДЕЛА

Добијене тежине из процеса федеративног обучавања се користе за тренирање модела над тестним скупом података. За евалуацију модела одабрана је F1 метрика, која представља меру перформанси бинарног класификатора и комбинује прецизност (енгл. *precision* - *p*) и одзив (енгл. *recall* - *r*). Прецизност дефинише тачност позитивних предикција, док одзив мери ефикасност у утврђивању позитивних инстанци. Висока вредност F1 метрике означава да је модел ефикасан приликом правилне идентификације свих позитивних инстанци и минимизирању броја лажно позитивних и лажно негативних предвиђања. Обука модела над тестним скупом података, као и евалуација, реализоване су покретањем R скрипте. Модел је показао задовољавајуће перформансе са F1 метриком од 0.89, што указује на добру равнотежу између прецизности и одзива.

## 6. ЗАКЉУЧАК

У овом мастер раду истраживана је примена федеративног учења у решавању проблема бинарне класификације. Објашњени су основни принципи дистрибуираних система и њихова примена у машинском учењу, као и проблеми са којима се суочавају приликом рада са великим скуповима података. Федеративно учење је уведено као нови приступ обуци модела над децентрализованим скуповима података. Имплементирани систем успешно је применио федеративно учење за бинарну класификацију података. Истраживања су показала могућности за даља унапређења кроз обуку на релевантнијим скуповима података, оптимизацију алгоритама и унапређење алгоритама агрегације.

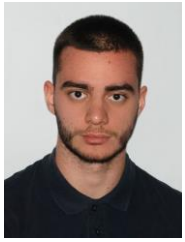
## 7. ЛИТЕРАТУРА

[1] George F Coulouris, Jean Dollimore, and Tim Kindberg. Distributed systems: concepts and design. pearson education, 2005.

[2] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguery Arcas. Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics, pages 1273--1282. PMLR, 2017.

[3] Maarten Van Steen and Andrew S Tanenbaum. Distributed systems. Maarten van Steen Leiden, The Netherlands, 2017.

**Кратка биографија:**



**Андрија Станишић** рођен је 25. јула 2000. године у Београду. Завршио је Гимназију „Исидора Секулић“ 2019. године након чега уписује Факултет техничких наука, смер Рачунарство и аутоматика. Дипломски рад је одбранио 2023. године када уписује мастер студије на студијском програму Рачунарство и аутоматика.