

EDGE IMPULSE

UPUTSVO ZA KORIŠĆENJE

PRIPREMILE:
DUBRAVKA ŠČEKIĆ
STANKA MIŠELJIĆ
AZRA RASTODER

SADRŽAJ

1. **Uvod**
2. **ŠTA JE EDGE IMPULSE**
3. **PRIKUPLJANJE PODATAKA**
4. **KREIRANJE IMPULSA**
5. **FREKVENCIJSKE KARAKTERISTIKE**
6. **KLASIFIKACIJA**
7. **UNAPRIJEĐENJE MODELA**
8. **IMPLEMENTACIJA NA UREĐAJU**
9. **ZAKLJUČAK**

UVOD

1. Svrha ovog uputstva

Ovo uputstvo ima za cilj da pruži praktičan i sistematičan pregled rada sa platformom Edge Impulse za razvoj modela vještačke inteligencije koji se izvršavaju direktno na ugrađenim (embedded) sistemima.

Na kraju ovog procesa korisnik treba da razumije kako se realni fizički signal pretvara u funkcionalni AI sistem na hardveru.

2. Uputstvo je namijenjeno:

- polaznicima AIoT Edu
- početnicima u oblasti TinyML i Edge AI
- inženjerima koji žele da implementiraju ML modele na mikrokontrolere
- studentima elektrotehnike, računarstva i biomedicinskog inženjeringa

Ne zahtijeva prethodno iskustvo sa Edge Impulse-om, ali se podrazumijeva osnovno znanje o radu sa mikrokontrolerima, senzorima i osnovama programiranja (Arduino / C++).

3. Šta ćete naučiti

Nakon prolaska kroz ovo uputstvo, korisnik će biti sposoban da:

pravilno organizuje i pripremi podatke

dizajnira impulse pipeline (processing + learning blokove)

generiše i analizira features

trenira i evaluira klasifikacioni model

deploy-uje model na Arduino Nano 33 BLE Sense

testira sistem u realnom vremenu

Drugim riječima, korisnik će savladati kompletan ML pipeline na edge uređaju.

4. Zašto Edge Impulse

Razvoj AI sistema za medicinske i nosive uređaje zahtijeva:

malu potrošnju energije

real-time obradu

rad bez cloud zavisnosti

visok nivo pouzdanosti

Edge Impulse omogućava:

automatizovanu obradu signala

optimizaciju modela za embedded sisteme

jednostavan deploy na mikrokontrolere

integraciju sa realnim senzorima

ŠTA JE EDGE IMPULSE?

Edge Impulse je platforma za razvoj mašinskog učenja na ugrađenim sistemima (mikrokontrolerima i edge uređajima). Omogućava kreiranje, treniranje i implementaciju AI modela koji rade direktno na hardveru, bez potrebe za cloud obradom.

Platforma je optimizovana za uređaje sa:

- ograničenom memorijom
- malom procesorskom snagom
- niskom potrošnjom energije

Edge AI koncept

Edge AI znači da se obrada podataka i inferencija izvršavaju direktno na uređaju (npr. Arduino), a ne na serveru. Prednosti ovog pristupa:

- manja latencija
- veća privatnost podataka
- pouzdan rad bez interneta
- pogodnost za medicinske i nosive uređaje

Zašto se koristi u ovom kursu

Edge Impulse je izabran jer:

- podržava Arduino Nano 33 BLE Sense
- omogućava rad sa realnim senzorima
- pojednostavljuje embedded AI razvoj
- koristi se u industriji



PRIKUPLJANJE PODATAKA

GOTOVI DATASET

Skup podataka koji već imamo pripremljen u originalnom formatu (npr. SisFall)

1

PRIPREMA FORMATA ZA EDGE IMPULSE

Python skriptom prilagođavamo fajlove u .csv format koji Edge Impulse zahtijeva, sređujemo header (ima nazive osa kao accX,accY,accZ,gyroX,gyroY,gyroZ) i sortiramo fajlove u odgovarajuće foldere za lakši upload i labeliranje.

2

DATA ACQUISITION U EDGE IMPULSE

Otvaramo sekciju Data Acquisition u Edge Impulse Studio i pripremamo interfejs za upload – ovdje uvodimo sve pripremljene .csv fajlove.

3

UPLOAD PODATAKA

Selektujemo folder(e) sa pripremljenim fajlovima i učitavamo ih u projekat detekcije.

4

LABELIRANJE I SPLITOVANJE

Ako su fajlovi već sortirani u foldere, najbolje je ručno imenovati foldere/labelu (npr. fall, non_fall). Platforma automatski dijeli podatke na trening i test set (npr. 80/20) koristeći hash fajla.

5

Šta je Data Acquisition?

Data Acquisition je faza u kojoj se podaci unose u Edge Impulse projekat i pripremaju za obradu i treniranje modela. U ovom slučaju ne snimamo podatke uživo, već koristimo već postojeći dataset koji je prethodno prilagođen pomoću Python skripte.

Data Acquisition sekcija

U Edge Impulse Studio interfejsu:

Otvoriti Data Acquisition panel

Izabrati opciju Upload data

Ovdje se vrši prenos lokalnih CSV fajlova u cloud projekat.

Način uvoza podataka

Pošto su fajlovi već organizovani po folderima (npr. fall i non_fall), preporučeni način je:

Select folder

Ova opcija omogućava:

upload više fajlova odjednom

automatsko prepoznavanje strukture foldera

brži rad sa velikim datasetima

Podešavanja tokom upload-a

1. Upload into category

Izabrati opciju:

Automatically split between training and testing

Ova opcija:

automatski dijeli podatke u odnosu 80% trening / 20% test

koristi hash fajla za konzistentnu podjelu

osigurava da isti fajl uvijek pripada istoj grupi

Ovo je standardna praksa u mašinskom učenju radi objektivne evaluacije modela.

2. Label

Pošto su fajlovi već sortirani u foldere:

label se može dodijeliti ručno

npr. fall i non_fall

Na ovaj način se osigurava jasna klasna struktura i izbjegavaju greške u označavanju.

Provjera nakon upload-a

Nakon uspješnog uvoza podataka, obavezno provjeriti:

da li su svi fajlovi pravilno učitani

da li su labele korektno dodijeljene

da li se signali ispravno prikazuju u Data Explorer sekciji

Ova provjera je kritična jer greške u ovoj fazi direktno utiču na kvalitet modela.

Zašto je Data Acquisition kritična faza?

model uči isključivo iz unesenih podataka

loše označeni podaci dovode do pogrešnih predikcija

pogrešan format onemogućava dalju obradu

Drugim riječima:

Kvalitet podataka direktno određuje kvalitet modela.

KREIRANJE IMPULSA

Šta je Impulse?

Impulse predstavlja ML pipeline koji definiše kako se sirovi podaci transformišu i kako se model uči.

Sastoji se od:

- Input block
- Processing block
- Learning block

Zajedno oni određuju način obrade signala i oblik modela.

U ovoj fazi ne treniramo model. Ovdje zapravo dizajniramo signalni i mašinski pipeline koji određuje:

kako se sirovi senzorski podaci dijele u uzorke,
kako se iz signala izvlače korisne karakteristike,
i koji tip modela će učiti te obrasce.

Drugim riječima, Create Impulse je mjesto gdje spajamo signal processing i machine learning u jedinstven sistem.

Kvalitet ovog koraka direktno određuje koliko će model biti stabilan, brz i tačan na stvarnom uređaju.

Kreiranje impulsa



KREIRANJE IMPULSA

Time Series Data

Šta je time series podataka?

Time series podaci su nizovi uzoraka koji su mjerene vrijednosti senzora u rastućim vremenskim tačkama — npr. akcelerometar koji uzima vrijednost 200 puta u sekundi. Edge Impulse koristi ovaj niz da nauči obrasce kao što je pad ili ne-pad.

Window size (veličina prozora)

Određuje koliko trajanja signala (u milisekundama) se koristi kao jedan primjer za učenje modela. Prekratak prozor neće uhvatiti dovoljno informacija da se razlikuje pad od normalnih pokreta. Predug prozor može uključiti nepotrebne podatke i otežati modelu da uoči relevantne osobine. Optimalna veličina zavisi od trajanja događaja koji model treba da prepozna — npr. pad se može desiti unutar jedne sekunde, pa se često bira prozor od ~1000 ms.

Window increase (pomeranje prozora)

Određuje za koliko se prozor pomjera unaprijed pri generisanju novih uzoraka iz dužeg signala. Ako su podaci duži od prozora, sistem napravi više uzoraka iz iste sekvence.

Manje pomjeranje (npr. 50 %) daje više preklapanja, što znači više uzoraka za učenje, ali i veći broj ulaza za model.

To pomaže da se bolje uhvate obrasci koji se javljaju u različitim dijelovima signala.

Sampling frequency (učestalost uzorkovanja)

Broj mjerenja u sekundi koje senzor snima (npr. 200 Hz znači 200 uzoraka/s). Viša frekvencija daje finergrained (detaljniju) sliku signala, ali isto tako povećava količinu podataka koje model mora obraditi.

Niža frekvencija može smanjiti zahtjeve za memorijom i energijom uređaja bez značajnog gubitka kvaliteta modela ako signali nisu previše brzi.

WINDOW SIZE

1
Određuje dužinu segmenta signala koji se koristi kao jedan primjer za model. Biramo ga da obuhvati cjelokupni događaj koji želimo detektovati.

WINDOW INCREASE

2
Određuje koliko se segment pomjera naprijed da bi se napravio novi primjer. Veći overlap daje više uzoraka iz istog signala.

SAMPLING FREQUENCY

3
To je učestalost uzorkovanja senzora (npr. 200 Hz). Viša frekvencija daje detaljnije podatke, ali više zahtjeva za memoriju.

KREIRANJE IMPULSA

Processing Block – konfiguracija obrade signala

Ovaj korak definiše kako će se sirovi senzorski podaci pretvoriti u numeričke karakteristike koje koristi model.

Izbor tipa obrade (Spectral Features)

Ako se radi sa podacima pokreta (akcelerometar, žiroskop):

- izaberi Spectral Features

Ovaj blok vrši frekvencijsku analizu signala i generiše karakteristike koje opisuju energiju i strukturu pokreta.

Provjeri da su označene odgovarajuće ose senzora.

Ose moraju odgovarati formatima podataka koji su uploadovani u dataset.

Learning Block – treniranje klasifikacionog modela

Learning block je dio gdje sistem prima generisane karakteristike, trenira model i uči da mapira obrasce u klase.

Ovo je faza gdje se podaci pretvaraju u statističko znanje.

Zašto se koristi klasifikacioni model?

Događaj se mora svrstati u jednu od diskretnih klasa.

On je optimizovan za mali footprint modela, brzo izvođenje inferencije i stabilan rad na MCU uređajima

Impulse #1

An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data.

The screenshot displays the 'Impulse #1' configuration interface. It consists of four main blocks arranged horizontally:

- Time series data (Red block):** Contains settings for 'Input axes (6)' (accX, accY, accZ, gyrX, gyrY, gyrZ), 'Window size' (2,000 ms), 'Window increase (stride)' (1,000 ms), 'Frequency (Hz)' (200), and a checked 'Zero-pad data' option.
- Spectral Analysis (White block):** Has a 'Name' field set to 'Spectral features' and 'Input axes (6)' with all six axes (accX, accY, accZ, gyrX, gyrY, gyrZ) selected via checkboxes.
- Classification (Purple block):** Has a 'Name' field set to 'Classifier', 'Input features' with 'Spectral features' checked, and 'Output features' listed as '4 (ADL, BACKWARD_FALL, FORWARD_FALL, LATERAL_FALL)'. Below this block is a dashed box labeled 'Add a learning block'.
- Output features (Green block):** Lists the same four classes: '4 (ADL, BACKWARD_FALL, FORWARD_FALL, LATERAL_FALL)'. A 'Save Impulse' button is located to the right of this block.

At the bottom of the interface, there are two dashed boxes: 'Add a processing block' (with a lightning bolt icon) and 'Add a learning block' (with a flask icon).

FREKVENCIJSKE KARAKTERISTIKE

Ova sekcija definiše način obrade signala prije ulaska u neuronsku mrežu. Cilj je transformacija vremenskih senzorskih podataka u frekvencijski domen radi izdvajanja stabilnih karakteristika za klasifikaciju.

1 Filter parametri

Scale axes - Vrijednost: 1

Ulazni podaci se ne skaliraju dodatno. Koristi se originalni opseg senzora definisan tokom akvizicije podataka i kada su podaci već normalizovani ili kada je fizička skala signala važna za model.

Input decimation ratio - Vrijednost: 1

Ne vrši se redukcija uzoraka. Svi vremenski uzorci ostaju u obradi. Ova postavka je preporučena kod aplikacija sa brzim tranzijentima, kao što su padovi i nagli pokreti.

Filter type - Vrijednost: none

Ne primjenjuje se dodatna filtracija (low-pass, high-pass, band-pass). FFT analiza se primjenjuje direktno na ulazni signal.

2 Analysis parametri

Analysis type - Vrijednost: FFT

Primjenjuje se Fast Fourier Transform radi transformacije signala iz vremenskog domena u frekvencijski domen. Ova transformacija omogućava izdvajanje spektralnih komponenti povezanih sa obrascima kretanja.

FFT length - Vrijednost: 16

Definiše veličinu FFT prozora.

Manje vrijednosti smanjuju potrošnju memorije, ubrzavaju obradu i pogodne su za mikrokontrolere.

Veće vrijednosti poboljšavaju frekvencijsku rezoluciju i povećavaju računarsko opterećenje.

Izabrana vrijednost predstavlja kompromis između brzine izvršavanja i informativnosti spektra.

Take log of spectrum - Vrijednost: enabled

Primjenjuje logaritamsku transformaciju amplituda spektra.

Efekt su smanjuje uticaj ekstremnih vrijednosti, stabilizuje ulazne podatke za neuronsku mrežu i poboljšava konvergenciju tokom treniranja.

Overlap FFT frames - Vrijednost: enabled

FFT prozori se djelimično preklapaju u vremenu.

Prednosti su bolja kontinualna analiza signala, smanjuje gubitak informacija na granicama prozora i poboljšava detekciju kratkotrajnih događaja.

Improve low frequency resolution - Vrijednost: disabled

Ova opcija poboljšava rezoluciju niskih frekvencija, ali povećava računarsko opterećenje.

Kod aplikacija sa brzim dinamičkim promjenama (npr. padovi) prioritet su srednje i visoke frekvencije, pa opcija ostaje isključena.

Raw features

Label

LATERAL_FALL

Parameters

Filter

Scale axes 1

Input decimation ratio 1

Type none

Analysis

Type FFT

FFT length 16

Take log of spectrum? ☒

Overlap FFT frames? ☒

Improve low frequency resolution? ☐

FREKVENCIJSKE KARAKTERISTIKE

Čuvanje podešavanja

Nakon što su parametri podešeni:

1. Klikni Save parameters
2. Klikni Generate features
3. Sačekaj završetak obrade

Provjera generisanih karakteristika

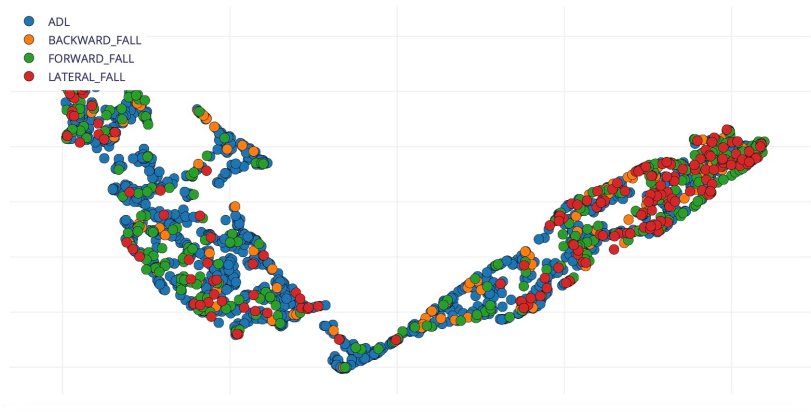
Nakon obrade:

- provjeri Data Explorer grafikon
- klase treba da budu djelimično razdvojene
- ako su potpuno preklapljene, potrebno je prilagoditi parametre ili dataset

Napomena za korisnike:

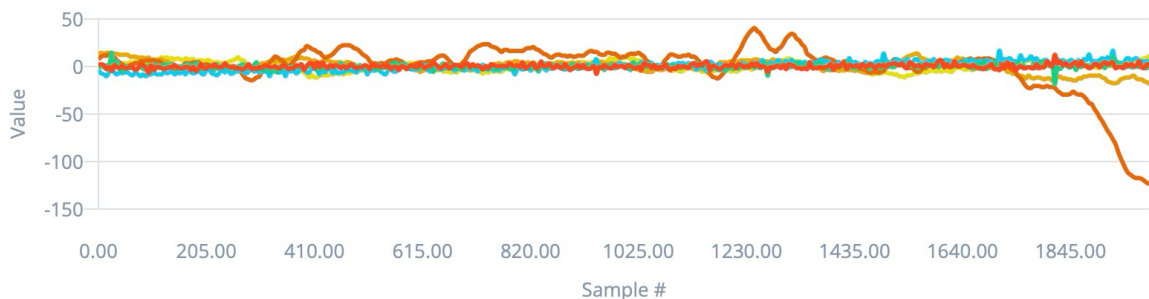
Spectral Features blok direktno utiče na kvalitet modela. Loša obrada signala ne može biti kompenzovana složenim modelom.

Feature explorer ①

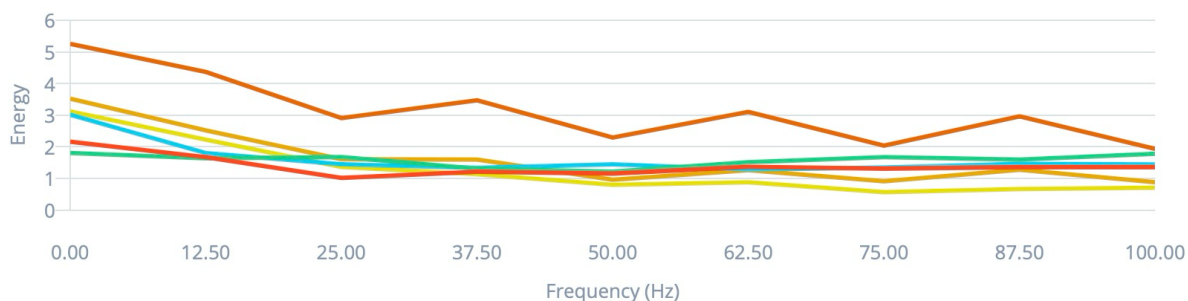


DSP result

After filter



Spectral power (log)



Processed features

2.1807, 0.5695, 2.4457, 2.2176, 3.3269, 1.6697, 1.0150, 1.2105, 1.1500, 1.3652, 1.3078, 1.3530, 1.3534, 2.342...

KLASIFIKACIJA

1. Training Settings

1.1 Number of training cycles (Epochs) - Vrijednost: 30

Ovaj parametar određuje koliko puta neuronska mreža

prolazi kroz kompletan trening skup podataka.

Omogućava stabilnu konvergenciju modela,

izbjegava pretreniranost (overfitting) i daje dobar

balans između tačnosti i vremena treniranja.

Napomena: povećanje broja epoha ne znači

automatski bolji model. Prevelike vrijednosti

mogu dovesti do lošije generalizacije.

1.2 Learning rate - Vrijednost: 0.0005

Learning rate kontroliše brzinu prilagođavanja težina

neuronske mreže. Izabrana vrijednost omogućava

stabilno učenje bez oscilacija, smanjuje rizik od

divergencije modela i pogodna je za signale

dobijene iz spektralnih karakteristika. Prevelika

vrijednost može izazvati nestabilnost,

dok premala produžava trening i usporava konvergenciju.

1.3 Training processor - Izabrana opcija: CPU

CPU trening se koristi jer ne zahtijeva dodatnu konfiguraciju, pogodan je za male i srednje modele i daje pouzdane rezultate u edukativnim okruženjima

Napomena: GPU se koristi kod velikih dataset-a i kompleksnijih arhitektura, ali nije neophodan za embedded ML primjene.

Neural Network settings



Training settings

Number of training cycles ?

30

Use learned optimizer ?

☐

Learning rate ?

0.0005

Training processor ?

CPU



KLASIFIKACIJA

2. Neural Network Architecture

U ovom eksperimentu koristi se klasična potpuno povezana neuronska mreža (Fully Connected Neural Network).

2.1 Input layer

Broj ulaznih feature-a: 78

Ovaj broj je automatski generisan iz Spectral Features bloka i predstavlja energetske karakteristike signala, frekvencijske komponente i statističke osobine senzorskih podataka

Input sloj ne trenira parametre već služi kao interfejs između DSP bloka i neuronske mreže.

2.2 Hidden layers (Dense layers):

Korišćena su dva skrivena sloja:

Prvi sloj: 20 neurona

Drugi sloj: 10 neurona

Ovakva struktura omogućava ekstrakciju nelinearnih obrazaca iz feature-a, smanjenje dimenzionalnosti i bolju generalizaciju modela.

Veći broj neurona povećava kapacitet mreže ali i rizik od overfitting-a i potrošnje memorije na embedded uređaju.

2.3 Output layer

Broj klasa: 4

Izlazni sloj odgovara broju klasa u dataset-u: ADL (normalne aktivnosti), Forward fall, Backward fall, Lateral fall

Svaki neuron izlaznog sloja predstavlja vjerovatnoću pripadnosti ulaznog uzorka jednoj klasi.

3. Pokretanje treninga

Nakon provjere parametara potrebno je:

Kliknuti na dugme Save & Train

Neural network architecture

Neural network

Complete architecture / classic ML

Input layer (78 features)

Dense layer (20 neurons)

Dense layer (10 neurons)

Add an extra layer

Output layer (4 classes)

Save & train



KLASIFIKACIJA

1. **Accuracy** znači procenat tačnih predikcija na validacionom skupu. Ovdje je tačnost ispod 50%, što je blizu nasumičnog pogađanja za 4 klase. To je jasan signal da model trenutno ne generalizuje dobro.
 - Ovo nije prihvatljivo za medicinske aplikacije
 - Pad detekcija ima sigurnosne implikacije
 - Cilj: minimum 80–90% za demo, više za realne sisteme
2. **Loss** mjeri koliko se izlazi modela razlikuju od stvarnih oznaka. Manji loss znači stabilniji i sigurniji model. Vrijednost 1.96 je visoka i pokazuje da mreža još nije naučila jasne granice između klasa.
 - Visok loss = loša separacija klasa
 - Ukazuje na problem u feature extraction ili dataset balansu
3. **Confusion Matrix** (najvažniji dio za medicinu)
ADL (normalne aktivnosti) Model relativno dobro prepoznaje normalne pokrete
BACKWARD FALL Model skoro uopšte ne prepoznaje pad unazad. Većinu ovih uzoraka pogrešno klasifikuje kao ADL
FORWARD FALL
LATERAL FALL - Najproblematičnija klasa.
4. F1 score kombinuje preciznost i odziv.
Vrijednosti ispod 0.5 znače da klasa praktično nije pouzdana.
 - ADL: 0.63 → jedino donekle stabilno
 - Ostalo: katastrofalno nisko
5. **Metrics** sekcija
AUC od 0.5 znači nasumično pogađanje. Mi smo samo malo iznad toga, što znači da model ne razdvaja klase dobro.
Precision 0.54 - Kada model kaže “ovo je pad”, tačno je samo u 54% slučajeva.
Recall 0.47 - Model propušta više od pola stvarnih padova. Ovo je kritično za medicinske sisteme.
Weighted F1 = 0.47
-Ukupna robusnost modela je trenutno slaba.

Model

Model version: ②

Quantized (int8) ▾

Last training performance (validation set)

📄

 **ACCURACY**
46.8%

 **LOSS**
1.96

Confusion matrix (validation set)

	ADL	BACKWARD_FALL	FORWARD_FALL	LATERAL_FALL
ADL	61.3%	0%	26.0%	12.7%
BACKWARD_FALL	70.0%	0.5%	21.1%	8.4%
FORWARD_FALL	56.6%	0%	26.3%	17.1%
LATERAL_FALL	68.3%	0%	12.4%	19.3%
F1 SCORE	0.63	0.01	0.23	0.16

Metrics (validation set)

📄

METRIC	VALUE
Area under ROC Curve ②	0.55
Weighted average Precision ②	0.54
Weighted average Recall ②	0.47
Weighted average F1 score ②	0.47

KLASIFIKACIJA

4. Data Explorer vizualizacija (donji scatter plot)

Zeleni = tačno

Crveni/roze = pogrešno

- Klasteri su izmiješani
- Nema jasnih granica između padova i ADL

Dobro istreniran model ima jasno razdvojene klasterne. Kod nas se klase preklapaju, što znači da feature extraction ne izdvaja dovoljno informacija.

5. On-device performance

Inference time: 1 ms

Model je ekstremno brz i pogodan za real-time obradu.

RAM: 1.4 KB

Flash: 16 KB

_Ovo je idealno za embedded uređaje i baterijske medicinske senzore.

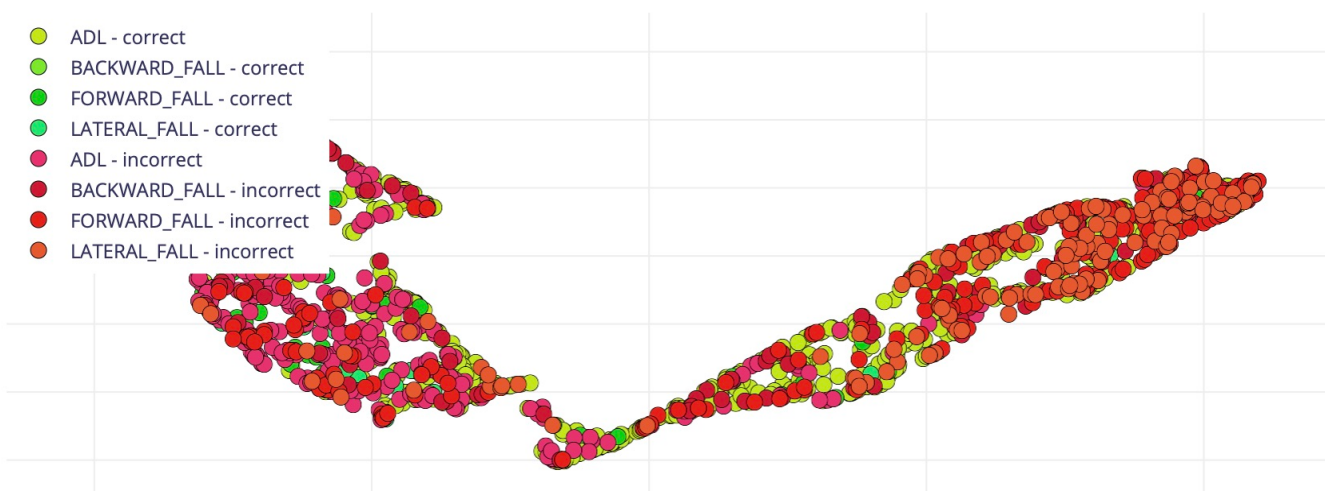
Znači:

- Model je hardverski efikasan
- Ali algoritamski slab

Ovaj primjer savršeno pokazuje realnost embedded AI-a: nije dovoljno samo trenirati model. Potrebno je balansirati tri stvari: kvalitet podataka, kvalitet feature extractiona i ograničenja hardvera.

Brz model nije nužno dobar model!

Data explorer (full training set) ?



On-device performance ?

Engine: ? **EON™ Compiler**



INFERRING TIME

1 ms.



PEAK RAM USAGE

1.4K



FLASH USAGE

16.0K

UNAPRIJEĐENJE

Kako unaprijediti model za detekciju pada (Preporuke za optimizaciju)

Na osnovu dobijenih rezultata (nizak accuracy, loša separacija klasa i visoka greška), predlažu se sledeći koraci za poboljšanje performansi modela:

1. Poboljšanje kvaliteta i balansa podataka

Problem: Neke klase (npr. BACKWARD_FALL i LATERAL_FALL) imaju veoma nisku tačnost.

Rješenje: Potrebno je povećati broj uzoraka za slabo prepoznate klase i izjednačiti broj primjera po klasi. Konkretno, dodati više realnih simulacija padova, izbalansirati dataset (isti broj uzoraka po klasi), ukloniti šumovite ili nekonzistentne zapise.

Neuralna mreža uči obrasce iz podataka. Ako su podaci loši, model ne može biti dobar.

2. Optimizacija Time Series parametara

Window size Testirati veće prozore (npr. 2500 ms ili 3000 ms) da bi cijeli pad bio obuhvaćen u jednom uzorku.

Pad traje više od 1 sekunde. Prekratak prozor gubi dinamiku pokreta

Window increase (stride) Smanjiti stride (npr. 500 ms) za veću preklapljenost podataka.

Sampling frequency Testirati 100 Hz umjesto 200 Hz radi smanjenja šuma i memorijskih zahtjeva.

3. Poboljšanje Spectral Features parametara

FFT lenght - Povećati FFT length (npr. sa 16 na 64 ili 128).

Improve low frequency resolution - Aktivirati opciju za bolju niskofrekventnu rezoluciju.

4. Promjena arhitekture neuronske mreže

Trenutno stanje: Mala mreža (20 + 10 neurona). Povećati broj neurona ili dodati dodatni sloj.

Veća mreža = veća potrošnja memorije. Potrebno je naći balans.

5. Povećanje broja trening ciklusa (epochs)

Trenutno: 30

Predlog: Testirati 50–100 trening ciklusa

Mreža dobija više vremena da nauči strukturu podataka. Smanjuje underfitting

6. Evaluacija pomoću Feature Explorer-a

Nakon svake izmjene, analizirati Feature Explorer da bi se vizuelno provjerilo da li se klase bolje razdvajaju.

Cilj: Jasni klasteri i manje preklapanje boja

Model se ne popravlja jednim klikom,
već sistemskim inženjerskim
pristupom.

IMPLEMENTACIJA NA UREĐAJU

Odabir “Deployment target”

Deployment target: Arduino library

Ova opcija generiše Arduino biblioteku (ZIP) koja sadrži:

- Edge Impulse inferencing SDK
- DSP blok (npr. Spectral features)
- trenirani model (TFLite)
- primjere (Examples) za podržane ploče

Zašto Arduino library: omogućava najbrži put od modela do rada na uređaju bez ručnog portovanja C++ koda.

Odabir “Inference engine”

Inference engine: EON™ Compiler (preporučeno)

EON Compiler optimizuje model za embedded uređaje i tipično:

- smanjuje potrošnju RAM-a i Flash-a
- zadržava približno istu tačnost (u većini slučajeva)
- ubrzava inferencu

Ukoliko postoji rizik da optimizacija utiče na tačnost, moguće je testirati obje varijante preko Run model testing.

Model optimizations and performance (tumačenje tabele)

U tabeli su prikazane procjene za ciljnu platformu (npr. Cortex-M4F 80 MHz). Razdvojeno je vrijeme i memorija za:

- Spectral features (DSP dio)
- Classifier (neuronska mreža)
- Total (ukupno)

a) Quantized (int8) – izabrano

- Latency (Total): ~76 ms
- RAM (Total): ~10.3 KB
- Classifier: ~1 ms (inferenca mreže)
- Spectral features: ~75 ms

Najveći dio vremena odlazi na DSP (ekstrakciju karakteristika), a sama neuronska mreža je veoma brza.

b) Unoptimized (float32)

- Latency je slična (~77 ms), ali:
- memorija i Flash su obično veći

Zaključak: int8 je pogodniji za mikrokontrolere jer smanjuje memoriju uz minimalan uticaj na brzinu.

Build i preuzimanje biblioteke

Kliknuti **Build**

Preuzeti generisani .zip fajl
(Arduino biblioteka)

Deployment target



Arduino library

An Arduino library with examples that runs on most Arm-based Arduino development boards.

Inference engine



EON™ Compiler

Same accuracy, 54% less RAM, 56% less ROM.

Model optimizations and performance

Model optimizations can increase on-device performance but may reduce accuracy. Performance estimate for Cortex-M4F 80MHz - [Change target](#)

Quantized (int8)

Selected ✓

	SPECTRAL FEATURES	CLASSIFIER	TOTAL
LATENCY	75 ms.	1 ms.	76 ms.
RAM	10.3K	1.4K	10.3K
FLASH	-	16.0K	-
ACCURACY	-	-	-

Unoptimized (float32)

Select

	SPECTRAL FEATURES	CLASSIFIER	TOTAL
LATENCY	75 ms.	2 ms.	77 ms.
RAM	10.3K	1.6K	10.3K
FLASH	-	17.9K	-
ACCURACY	-	-	-

To compare model accuracy, run model testing for all available optimizations.

Run model testing

Build

IMPLEMENTACIJA NA UREĐAJU

Dodavanje biblioteke u Arduino IDE

1. Otvoriti Arduino IDE
2. Izabrati: Sketch → Include Library → Add .ZIP Library...
3. Odabrati preuzeti .zip fajl biblioteke
4. Nakon instalacije, primjeri se pojavljuju pod:
 - File → Examples → (naziv projekta) →

Pokretanje inferencije na uređaju

1. Povezati Arduino Nano 33 BLE Sense USB kablom
2. U Arduino IDE odabrati:
 - Tools → Board → Arduino Nano 33 BLE (tačna verzija: rev1 ili rev2)
 - Tools → Port → /dev/cu...
3. Otvoriti primjer koji odgovara korišćenim senzorima/modelu (npr. IMU/accelerometer/fusion)
4. Kliknuti Upload
5. Otvoriti Serial Monitor (115200 baud) i posmatrati izlaz:
 - vjerovatnoće klasa (npr. fall vs non-fall)
 - vrijeme DSP i klasifikacije

```
Predictions (DSP: 330 ms., Classification: 0 ms., Anomaly: 0 ms.):  
ADL: 0.21875  
BACKWARD_FALL: 0.10547  
FORWARD_FALL: 0.45703  
LATERAL_FALL: 0.21875
```

```
Predictions (DSP: 330 ms., Classification: 0 ms., Anomaly: 0 ms.):  
ADL: 0.45703  
BACKWARD_FALL: 0.10547  
FORWARD_FALL: 0.21875  
LATERAL_FALL: 0.21875
```

Napomena o kompatibilnosti senzora

Model zahtijeva tačno definisane ulazne ose (npr. accX, accY, accZ, gyroX, gyroY, gyroZ).

Ako primjer na ploči koristi drugačiji senzor ili ne šalje sve ose, pojaviće se greška tipa:

“Sensors don't match the sensors required in the model”

U tom slučaju potrebno je:

- koristiti primjer koji odgovara modelu (accelerometer ili fusion)
- ili ponovo trenirati impulse sa osi koje su dostupne na uređaju

```
83  
84     "accX", &data[0], &poll_acc, &init_IMU, NOT_USED,  
85     "accY", &data[1], &poll_acc, &init_IMU, NOT_USED,  
86     "accZ", &data[2], &poll_acc, &init_IMU, NOT_USED,  
87     "gyrX", &data[3], &poll_gyr, &init_IMU, NOT_USED,  
88     "gyrY", &data[4], &poll_gyr, &init_IMU, NOT_USED,  
89     "gyrZ", &data[5], &poll_gyr, &init_IMU, NOT_USED,  
90     "magX", &data[6], &poll_mag, &init_IMU, NOT_USED,  
91     "magY", &data[7], &poll_mag, &init_IMU, NOT_USED,  
92     "magZ", &data[8], &poll_mag, &init_IMU, NOT_USED,  
93  
94     "temperature", &data[9], &poll_HTS, &init_HTS, NOT_USED,  
95     "humidity", &data[10], &poll_HTS, &init_HTS, NOT_USED.
```

Output Serial Monitor ×

Message (Enter to send message to 'Arduino Nano 33 BLE' on '/dev/cu.usbmodem101')

Starting inferencing in 2 seconds...

ERR: Sensors don't match the sensors required in the model

Following sensors are required: accX + accY + accZ + gyroX + gyroY + gyroZ

ZAKLJUČAK

U ovom uputstvu prikazan je kompletan proces razvoja Edge AI sistema, od pripreme podataka do implementacije modela na ugrađeni uređaj. Studenti su stekli praktično razumijevanje rada sa vremenskim signalima, ekstrakcije karakteristika, treniranja modela i optimizacije za embedded okruženje.

Stecheno znanje predstavlja osnovu za razvoj naprednih inteligentnih sistema u oblastima medicinskih uređaja, nosive elektronike i pametnih senzora. Dalji rad treba da bude usmjeren na proširenje dataset-a, optimizaciju parametara i testiranje modela u realnim uslovima.