Universitatea POLITEHNICA din București
Facultatea de Electronică, Telecomunicații și Tehnologia Informației

Reproducerea automată a mișcării folosind tehnici de învățare profundă

Proiect de diplomă

prezentat ca cerință parțială pentru obținerea titlului de *Inginer*
în domeniul *Calculatoare și tehnologia informației*
programul de studii de licență
Ingineria informației

Conducători științifici
Prof. dr. ing. Corneliu Burileanu
As. drd. ing. Ana-Antonia Neacșu

Absolvent
Cătălina Guță

2023

Universitatea “Politehnica” din Bucureşti
Facultatea de Electronică, Telecomunicaţii şi Tehnologia Informaţiei
Program de studiu **INF**

Anexa 1

TEMA PROIECTULUI DE DIPLOMĂ

a studentului **GUŢĂ N. Cătălina, 442A**

1. Titlul temei: Reproducerea automată a mișcării folosind tehnici de învățare profundă

2. Descrierea temei și a contribuției personale a studentului (în afara părții de documentare):
Lucrarea de față va avea în vedere reproducerea mișcării membrelor inferioare de către un exoschelet folosind date biometrice. Se va colecta o bază de date cu ajutorul unui senzor de accelerație, ADXL345, iar aceasta va fi folosită pentru antrenarea diferitelor modele de rețele neuronale dezvoltate folosind biblioteci specializate (Keras și TensorFlow din Python). Se urmărește optimizarea modelului în scopul reproducerii fidele a mișcărilor. Exoscheletul este format din patru motoare RMD X8 PRO comandate fiecare de către un microcontroler ESP32 conectate prin protocol CAN. Cu ajutorul aplicației MQTT ce comunică prin WI-FI cu un microcontroler ESP32 intermerdiar, se va realiza conectarea prin protocol UART cu un alt ESP32 master. Cel din urmă va comanda prin protocolul ESP-NOW patru microcontrolere ESP32 slave care vor avea asociate câte un motor RMD X8 PRO.

3. Discipline necesare pt. proiect:
Microcontrolere(MC), recunoașterea formelor si inteligență artificială (RFIA), tehnici de optimizare (TOP)

4. Data înregistrării temei: 2022-12-05 09:03:11

Conducător(i) lucrare,
As. drd. Ing. Ana-Antonia NEACȘU

prof. Corneliu Burileanu

Student,
GUŢĂ N. Cătălina

Director departament,
Ș.L. dr. ing Bogdan FLOREA

Decan,
Prof. dr. ing. Mihnea UDREA

Cod Validare: **168ac2f0b8**

Declaraţie de onestitate academică

Prin prezenta declar că lucrarea cu titlul *Reproducerea automată a mișcării folosind tehnici de învățare profundă*, prezentată în cadrul Facultăţii de Electronică, Telecomunicaţii şi Tehnologia Informaţiei a Universităţii "Politehnica" din Bucureşti ca cerinţă parţială pentru obţinerea titlului de *Inginer* în domeniul Inginerie Electronică şi Telecomunicaţii/ Calculatoare şi Tehnologia Informaţiei, programul de studii *Ingineria Informației* este scrisă de mine şi nu a mai fost prezentată niciodată la o facultate sau instituţie de învăţământ superior din ţară sau străinătate.

Declar că toate sursele utilizate, inclusiv cele de pe Internet, sunt indicate în lucrare, ca referinţe bibliografice. Fragmentele de text din alte surse, reproduse exact, chiar şi în traducere proprie din altă limbă, sunt scrise între ghilimele şi fac referinţă la sursă. Reformularea în cuvinte proprii a textelor scrise de către alţi autori face referinţă la sursă. Înţeleg că plagiatul constituie infracţiune şi se sancţionează conform legilor în vigoare.

Declar că toate rezultatele simulărilor, experimentelor şi măsurătorilor pe care le prezint ca fiind făcute de mine, precum şi metodele prin care au fost obţinute, sunt reale şi provin din respectivele simulări, experimente şi măsurători. Înţeleg că falsificarea datelor şi rezultatelor constituie fraudă şi se sancţionează conform regulamentelor în vigoare.

Bucureşti, Iulie 2023.
Absolvent: Cătălina Guță

.....

Cuprins

Listă de figuri

Listă de tabele

Listă de acronime

ACC *Accuracy* - Acuratețe

Adam *Adaptive Moment Estimation* - Algoritm cu momente statistice adaptive

CSV *Comma Separated Values* - Valori separate prin virgulă

FN *False Negative* - Fals negative

FP *False Positive* - Fals pozitive

I2C *Inter-Integrated Circuit* - Între circuite integrate

ISM *Industrial, Scientific and Medical* - Industrial, științific și medical

LED *Light Emitting Diode* - Diodă emițătoare de lumină

Li-Po *Lithium Polymer* - Litiu polimer

mAh *miliAmpere hour* - miliAmper oră

MAV *Mean Absolute Value* - Valoarea medie absolută

MLP *Multilayer Perceptron* - Perceptronul multinivel

NumPy *Numerical Programming Package* - Pachet pentru programare numerică

ReLU *Rectified Linear Unit* - Funcție identitate redresată

SCL *Serial Clock* - Linia de ceas

SDA *Serial Data* - Linia de date

SELU *Scaled Exponential Linear Unit* - Funcție identitate exponențială scalată

TN *True Positive* - Adevărat pozitive

TP *True Negative* - Adevărat negative

USB *Universal Serial Bus* - Magistrală serială Universală

WL *Waveform Length* - Lungimea semnalului

ZC *Zero Crossing Rate* - Rata de trecere prin zero

Introducere

Motivație

În lucrarea de față va fi analizată motricitatea membrelor inferioare, cu accent pe coordonare, cu scopul de a realiza un model de învățare profundă pentru diverse tipuri de mișcări.

O motricitate normală a membrelor inferioare este extrem de importantă pentru o persoană în ceea ce privește capacitatea sa de a-și menține independența. Aceasta permite unui om să meargă, să alerge, să urce și să coboare scări, să se așeze și să se ridice de pe un scaun, să își mențină echilibrul în timpul activităților zilnice.

Tehnologia a dus la implementarea unor proiecte creative în ceea ce privește diverse domenii, cum ar fi sportul sau domeniul medical.

Aplicabilitate

Mișcările membrelor inferioare pot fi reproduse cel mai simplist în format digital. O animație sincronizată cu activitățile unei persoane poate fi folosită în scop recreativ, dar și pentru monitorizare.

În schimb, reproducerea automată a mișcării de către un exoschelet duce la soluționarea mai multor probleme cum ar fi intervenții în spații închise contaminate, atingerea unor forțe neobișnuite în scopul de a ridica obiecte grele. Motricitatea membrelor inferioare poate fi afectată de diferite afecțiuni patologice, care ar duce la nevoia de tratament și de asistență. Un exoschelet poate ajuta la realizarea funcției de locomoție, acționând complementar cu corpul uman.

Totodată, monitorizarea coordonării membrelor inferioare este esențială pentru observarea schimbărilor de performanță în situații precum urmarea unui tratament sau a unui antrenament sportiv.

Obiective

Scopul proiectului se împarte în următoarele etape:

- Implementarea unui sistem de achiziție
- Crearea bazei de date
- Preprocesarea datelor
- Dezvoltarea unui model de clasificare

1 Noțiuni teoretice

Pentru început este nevoie de clarificarea unor noțiuni teoretice care stau la baza realizării acestui proiect.

1.1 Motricitatea membrelor inferioare

Motricitatea membrelor inferioare se referă la abilitatea și controlul mișcărilor realizate de acestea. Membrele inferioare includ picioarele, coapsele și gambele. Acestea joacă un rol crucial în deplasarea, echilibrul și susținerea greutății corpului.

Motricitatea este coordonată de sistemul nervos central, care include creierul și măduva spinării. Aceste componente controlează și coordonează activitatea musculară și transmit semnale motorii către mușchii membrelor inferioare.

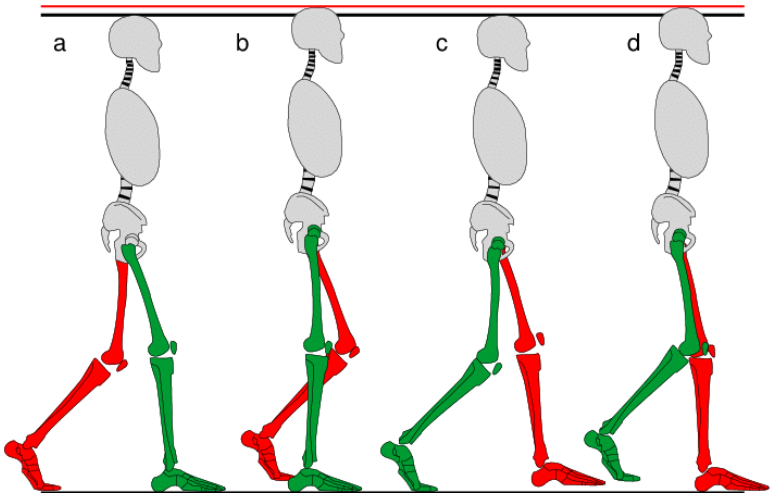

Figura 1.1: Etapele mișcării picioarelor în timpul mersului

Există mai multe aspecte ale motricității membrelor inferioare.

- Forța musculară se referă la capacitatea mușchilor de a genera tensiune pentru realizarea mișcărilor și susținerea greutății corporale.
- Coordonarea reprezintă capacitatea de a controla mișcările într-un mod precis și eficient.
- Echilibrul este esențial pentru menținerea poziției verticale și pentru a prevenirea căderilor.
- Flexibilitatea implică capacitatea de a întinde și a îndoi mușchii împreună cu articulațiile într-un mod controlat și fără a provoca disconfort sau leziuni.

Motricitatea membrelor inferioare poate fi afectată de diverse afecțiuni patologice, cum ar fi leziuni ale măduvei spinării sau paralizie cerebrală. În astfel de cazuri, recuperarea este esențială pentru a îmbunătăți motricitatea și funcționalitatea membrelor inferioare.

Motricitatea membrelor inferioare este esențială pentru mobilitatea și independența noastră de zi cu zi. [1]

1.2 Procesarea semnalelor

Majoritatea informației din acest capitol se bazează pe referința [2].

1.3 Conversia analog - digital

Mișcările membrelor inferioare pot fi considerate semnale analogice. Pentru a le putea aduce pe acestea într-un format digital, înțeles de calculator, este necesar un proces de conversie numit și digitalizare.

Digitalizarea se rezumă la convertirea valorilor continue în valori discrete care pot fi mai apoi reprezentate în sistem binar. Poate fi împărțită în două etape.

1. Eșantionare
   În etapa de eșantionare se dorește prelevarea valorilor atinse de semnalul analogic la momente egale de timp, astfel ajungându-se la noțiunea de *frecvență de eșantionare*. Frecvența de eșantionare este esențială, aceasta având un rol cheie în reproducerea optimă a informației adusă de semnal. Principiul lui Nyquist este un concept care afirmă că valoarea minimă a frecvenței de eșantionare trebuie să fie cel puțin egală cu dublul frecvenței maxime din spectru.
   $$f_s \geq 2 \cdot f_{max} \tag{1.1}$$
   În formulă a fost notată valoarea frecvenței de eșantionare cu f_s, iar frecvența maximă atinsă de semnalul analogic cu f_{max}.

   Totodată, există și o limită superioară a frecvenței de eșantionare care în cazul în care este depășită datele vor avea o redundanță ridicată, neacceptată. Deși sunt extrase mai multe eșantioane din semnal, acestea nu ne aduc informație suplimentară și ne îngreunează procesul de prelucrare, fiind nevoie de mai multe resurse.

   O eșantionare optimă oferă un echilibru între redarea fidelă a semnalului și captarea unui volum de date minim.

2. Cuantizare
   Cea de-a doua etapă a conversiei unui semnal analogic într-un semnal digital constă în atribuirea unei valori discrete fiecărui eșantion captat în etapa precedentă. Se împarte intervalul de valori în care se poate afla amplitudinea semnalului analogic într-un număr finit de trepte, de niveluri discrete.

   Similar cu aspectele prezentate în etapa de eșantionare, este nevoie de un compromis între mărimea nivelurilor în care este împărțit intervalul și resursele necesare. Mărimea nivelurilor definesc rezoluția procesului de cuantizare. Cu cât acestea sunt mai mici, rezoluția este mai scăzută și resursele necesare sunt mai mari. În schimb, dacă se dorește o rezoluție mare, treptele în care este împărțit intervalul sunt mai mari, dar volumul de calcul scade.

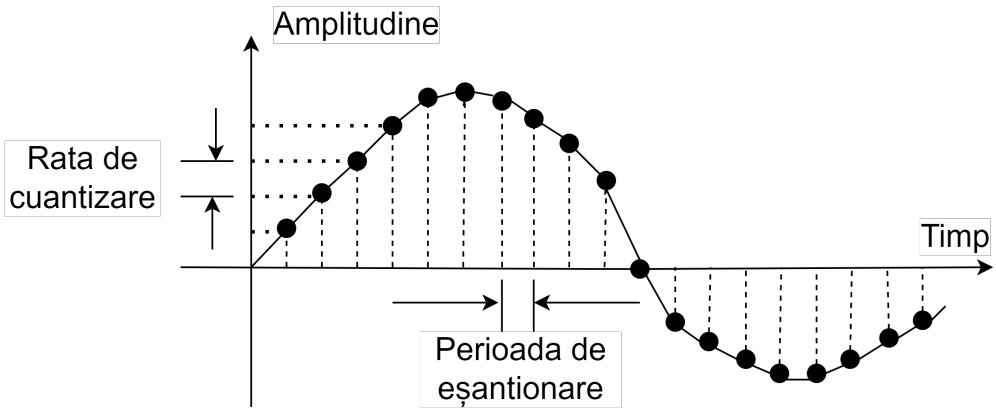

Figura 1.2: Eșantionare și cunatizare

În cele ce urmează sunt explicate procesele de bază aplicate semnalelor digitale pentru a ajunge într-un format avantajos din punct de vedere al extragerii caracteristicilor, caracteristici ce vor contribui la antrenarea rețelei neurale.

1.4 Ferestruire

Ferestruirea cu ajutorul unei ferestre glisante rectangulare este o tehnică utilizată în prelucrarea semnalelor pentru a segmenta o înșiruire mai lungă de date în porțiuni mai mici, fără a aduce modificări semnalului în sine. Această tehnică este utilizată în analiza semnalelor și în prelucrarea datelor secvențiale.

Principiul de bază al acestui tip de ferestruire constă în deplasarea unei fereastre cu dimensiune fixă de-a lungul semnalului. Aceasta selectează o porțiune de semnal egală cu ea fără să aducă modificări. Fereastra se deplasează într-o manieră glisantă, astfel încât să acopere întreaga secvență de date. Porțiunea de suprapunere a două ferestre consecutive se numește *overlap* și determină gradul de detaliere cu care este analizat semnalul. Cu cât suprapunerea este mai mare, cu atât se formează mai multe ferestre, iar pasul de glisare a acesteia este mai mic. Acest scenariu poate fi avantajos, dar poate fi un dezavantaj din punct de vedere al resurselor implicate pentru prelucrarea unui volum mare de date.

Această tehnică poate fi folosită în diferite aplicații, cum ar fi analiza unor semnale și evoluția lor în timp, situația acestei lucrări. Ajută la extragerea de caracteristici temporale ale semnalului. Prin segmentarea semnalului în secțiuni mai mici cu ajutorul ferestrei glisante, se poate realiza o analiză detaliată a proprietăților temporale ale acestuia.

Numărul de ferestre, N_w, poate fi calculat cu formula de mai jos. S este numărul de valori totale ale semnalului, W este numărul de eșantioane cuprinse într-o fereastră, iar O numărul de eșantioane cuprinse în suprapunere.

$$N_w = \left[\frac{S-W}{W-O}\right] + 1 \tag{1.2}$$

1.5 Padding

Padding-ul înseamnă adăugarea de valori suplimentare la finalul semnalului pe care se dorește realizată ferestruirea. Reprezintă procesul de ajustare a dimensiunii setului de date astfel încât să cuprindă un număr întreg de ferestre. Este o etapă opțională, care poate lipsi, alegându-se ca soluție alternativă renunțarea la valorile finale care nu ocupă integral o fereastă. Există două moduri de realizare a paddingului.

- Zero-padding
  Aceasta implică adăugarea de valori de zero la marginea semnalului. Acesta este cel mai comun tip de padding utilizat. Prin adăugarea de zero-uri, dimensiunea semnalului poate fi ajustat la dimensiunea dorită. De exemplu, în cazul prelucrării semnalelor analizate în acest proiect, zero-padding-ul poate fi folosit pentru a adăuga zero-uri la sfârșit pentru a-l extinde la o lungime specifică. Totuși, nu este o soluție preferată deoarece sunt adăugate valori eronate care ar influența rezultatele caracteristicilor.

- Reflective padding
  Acest tip de padding implică copierea simetrică a valorilor semnalului. Aceasta înseamnă că valorile semnalului sunt reflectate și adăugate pe margine, în ordine inversă. Reflective padding este adesea folosit pentru a evita adăugarea de valori eronate.

Numărul de ferestre formate când se utilizează padding, N_wp, este mai mare cu o unitate față de situația în care se recurge la renunțarea valorilor de la final.

$$N_{wp} = \left[\frac{S-W}{W-O}\right] + 2 \tag{1.3}$$

Atât ferestruirea cât și modul în care poate fi realizat padding-ul pot fi vizualizate în figura 1.3.

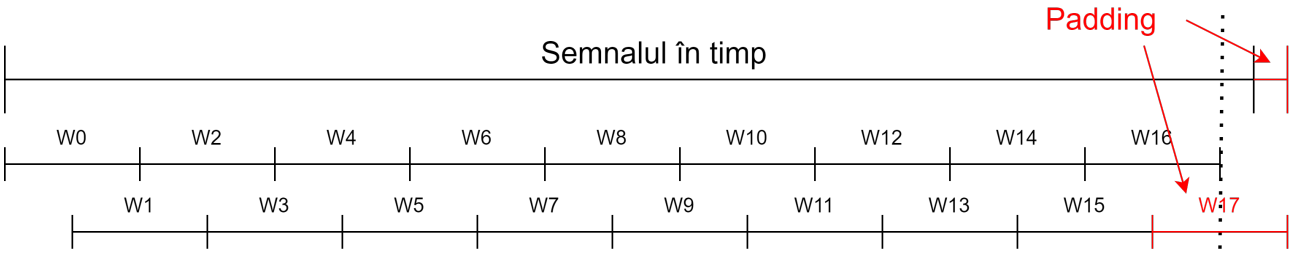

Figura 1.3: Ferestruirea și padding-ul

1.6 Caracteristici

Din fiecare fereastră selectată sunt extrase anumite caracteristici care se doresc a fi relevante pentru semnalul analizat. [3]

- Media absolută - MAV
  Valoarea medie absolută, cunoscută și sub denumirea de media absolută, este valoarea centrala a modulului unui set de valori. Se calculează prin determinarea modulului fiecărui

element din setul de date și calcularea mediei aritmetice. Procesul e împărțit în următorii pași:

1. Fiecărei valori din setul de date i calculează valoarea absolută. Valoarea absolută a unei valori reprezintă modulul acesteia.
2. Calculează media valorilor absolute calculate în pasul anterior. Aceasta se face prin adunarea tuturor modulelor și împărțirea sumei rezultate la numărul de valori.

Media absolută a valorilor este utilă pentru a determina valoarea reprezentativă a setului de date, fără a ține cont de semnul valorilor, pozitiv sau negativ. Astfel, ea reflectă o măsură a valorii medii a amplitudinilor din setul de date, ignorând semnul acestora.

$$MAV(x) = \frac{1}{N}\sum_{i=0}^{N-1}|x_i| \tag{1.4}$$

- Rata de trecere prin zero - ZC
  Este o caracteristică extrasă pentru prelucrarea semnalelor digitale. Aceasta evaluează de câte ori semnalul analizat a trecut de la o valoare pozitivă la o valoare negativă sau invers. Procesul se rezumă la o singură etapă care constă în numărarea momentelor în care valoarea datele au trecut prin valoarea zero. α este o constantă care asigură un rezultat corect chiar și în cazul unui semnal zgomotos pe porțiuni.

$$ZC(x) = |\{i : (|x_i - x_{i-1}| \geq \alpha) \wedge (sgn(x_i) \neq sgn(x_{i-1}))\}| \tag{1.5}$$

- Skewness
  Este o măsură statistică utilizată pentru a evalua asimetria unor valori numerice. Aceasta furnizează informații despre modul în care partea aflată în afara mediei distribuției este distribuită în raport cu aceasta. Această caracteristică cuantifică dacă datele au o asimetrie spre stânga, skewness negativ, spre dreapta, skewness pozitiv, sau sunt simetrice, skewness nul.

  Este important de menționat că skewness-ul este doar o caracteristică care se referă la asimetrie și nu oferă informații complete despre distribuția valorilor. Prin urmare, este recomandat să fie utilizată împreună cu alte măsuri statistice și metode de analiză pentru o evaluare mai detaliată a datelor.

$$Skewness(x) = \frac{1}{N}\sum_{i=0}^{N-1}\left(\frac{x_i - \overline{x}}{\sigma}\right)^3 \tag{1.6}$$

- Lungimea semnalului - WL
  Este măsura care arată cât de mult a variat magnitudinea valorilor analizate. Se calculează diferența între două eșantioane consecutive, acesteia i se aplică modulul, iar în final toate numerele rezultate se însumează.

  În cazul unui semnal cu variații mari și dese ale amplitudinii se obține o valoare pentru

WL mare, iar în cazul în care semnalul este aproape constant, aceasta va fi aproape zero.

$$WL(x) = \sum_{i=1}^{N} |x_i - x_{i-1}| \tag{1.7}$$

- Deviația standard - σ
  Este o măsură statistică care cuantifică dispersia sau variabilitatea datelor într-o distribuție. Reprezintă o măsură a diferenței între fiecare punct de date și media acestora. Cu cât deviația standard este mai mare, cu atât valorile din distribuție sunt mai variabile sau mai răspândite în jurul valorii medii.

  Procesul de calcul al deviației standard constă în următoarele etape:

  1. Se calculează media distribuției, $\overline{x}$, prin însumarea tuturor punctelor de date și împărțirea la numărul total de valori.
  2. Se calculează diferența între fiecare punct de date și media distribuției
  3. Se ridică la pătrat fiecare diferență.
  4. Se calculează media valorilor ridicate la pătrat.
  5. În final, se aplică radical pe media valorilor ridicate la pătrat pentru a obține rezultatul dorit.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1}(x_i - \overline{x})^2} \tag{1.8}$$

1.7 Normalizarea datelor

Normalizarea datelor este un proces utilizat în analiza și prelucrarea datelor pentru a aduce valorile acestora la o scară comună sau pentru a le transforma într-un interval specific. Există mai multe moduri diferite de normalizare a datelor, dintre care pot fi amintite:

- Normalizarea min max
  Această metodă aduce datele într-un interval specific, uzual între 0 și 1. Datele inițiale sunt redimensionate astfel încât valoarea minimă devine 0, iar valoarea maximă devine 1.

  În formula acestui tip de normalizare $min(x)$ este valoarea minimă a datelor analizate, $max(x)$ reprezintă valoarea maximă a acestora, x_i valoarea inițială, iar x_i' este valoarea normalizată.

$$x_i' = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{1.9}$$

- Standardizarea
  Această metodă transformă datele astfel încât acestea să aibă o media 0 și dispersia 1. Valorile inițiale sunt ajustate în funcție de media și deviația standard a valorilor analizate.

  Ca în formula anterioară, x_i este valoarea inițială, x_i' este valoarea normalizată, cu $\overline{x}$ a

fost notată media datelor, iar cu σ dispersia.

$$x_i' = \frac{x_i - \overline{x}}{\sigma} \tag{1.10}$$

- Normalizarea sumei absolute
  Această metodă normalizează valorile în funcție de suma valorilor absolute ale datelor. Rezultatul este o distribuție în care suma valorilor absolute este 1.

  Notațiile din formula normalizării prezentate în acest subpunct au fost explicate anterior.

$$x_i' = \frac{x_i}{\sum_{i=0}^{N-1} |x_i|} \tag{1.11}$$

1.8 Distribuția gaussiană

Distribuția gaussiană cunoscută și sub numele de distribuție normală, este una dintre cele mai importante și des întâlnite distribuții în statistica și în teoria probabilităților. Caracteristicile care merită amintite sunt expuse în lista următoare:

- Distribuția gaussiană are o formă de clopot complet determinată de media μ și deviația standard σ.
- Media reprezintă centrul distribuției, iar deviația standard indică măsura dispersiei sau variabilității datelor în jurul acesteia.

- Distribuția normală respectă regula celor trei sigma. Conform acestei reguli, aproximativ 68.27% dintre valorile se află la o deviație standard față de medie, 95.45% se află la două deviații standard față de medie, iar 99.73% se află la trei deviații standard față de medie.

O observație importantă este legătura între standardizarea datelor și distribuția gaussiană. Prin standardizare datele pot fi caracterizate de o distribuție normală de medie 0 și dispersie 1. Indiferent de intervalul în care erau cuprinse datele inițial, acestea vor fi cuprinse în proporție de 99.73% în intervalul $[-3, 3]$.
Distribuția normală se notează cu $\mathcal{N}$ și este descrisă de medie μ și dispersie σ^2.

$$\mathcal{N}(\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{1.12}$$

Suportul teoretic ce stă la baza ultimelor 3 secțiuni face parte din referința [4].

subsubsection

1.9 Rețele neurale. Perceptronul multinivel

Modelele de învățare profundă reprezintă un subdomeniu important al inteligenței artificiale care a avut un impact semnificativ asupra domeniilor tehnologiei și informaticii din ultimii ani. Domeniul a fost inspirat de modul în care funcționează creierul uman, cu neuroni

interconectați în straturi multiple pentru a procesa informațiile. Aceste modele au revoluționat modul în care calculatoarele înțeleg și procesează informațiile, permițându-le să învețe și să facă predicții în mod autonom.

Majoritatea noțiunilor din acest capitol se bazează pe referința [4].

Învățarea automată constă în dezvoltarea de algoritmi și modele matematice care permit calculatoarelor să învețe din datele furnizate și experiență. Pot fi capabile să identifice modele și să facă predicții sau să ia decizii bazate pe acestea.

Învățarea profundă este o ramură a învățării automate care se concentrează pe utilizarea rețelelor neuronale artificiale cu mai multe straturi, numite rețele neurale profunde, pentru a extrage caracteristici complexe și a înțelege datele oferite.

Prin folosirea unor arhitecturi complexe de rețele neuronale, învățarea profundă a avut succes în domenii precum recunoașterea de imagini, recunoașterea vocală sau traducerea automată. Aceasta a dus la progrese semnificative în domenii precum asistență virtuală, vehicule autonome, finanțe sau medicină. Totodată, rețelele neuronale aduc preocupări ce țin de etica datelor implicate și utilizărarea modelelor pentru interpretarea acestora.

Perceptronul multinivel se numără printre cele mai populare și des utilizate arhitecturi. Cuvântul *perceptron* este un cuvânt derivat din cuvântul *percepție* care reprezintă procesul psihic de cunoaștere senzorială.

1.9.1 Structura

MLP este o rețea fără reacție, feedforward, care permite informației să se propage într-o singură direcție, de la stratul de intrare spre stratul de ieșire. Fiecare neuron este conectat cu toți neuronii din stratul următor, fără a fi prezente conexiuni care să formeze cicluri, care ar permite informației să se transmită între neuroni din același start sau revenirea acesteia la stratul anterior. Fluxul de informație unidirecțional face perceptronul multinivel să fie mai ușor de antrenat și mai intuitiv.

Straturile de neuroni sunt împărțite în 3 categorii.

- Stratul de intrare
  Neuronii din primul strat, stratul de intrare, sunt caracterizați ca fiind virtuali, transparenți. Aceștia nu prelucrează semnalul, iar de aceea pot fi denumiți și neuroni pasivi. Numărul lor este egal cu numărul de valori care se află într-un vector exemplu, în cazul nostru 15.

- Straturile intermediare
  Neuronii intermediari pot fi împărțiți pe mai multe straturi fără a fi impus un număr fix pe fiecare. Acestea se numesc straturi ascunse. Se comportă ca detectori de caracteristici, participând la prelucrarea datelor.

  Un neuron dintr-un strat intermediar realizează prelucrarea datelor prin două etape.

  1. Etapa liniară

$$\mathbf{z} = \mathbf{W} \cdot \mathbf{x} + \mathbf{b} \tag{1.13}$$

W este matricea ponderilor asociate sinapselor cu neuronul de interes, x este exemplul, vector unidimensional, iar b este termenul de deplasare, bias, tot un vector unidimensional care are rolul de a ajusta pragul de activare din etapa neliniară.

2. Etapa neliniară

$$\mathbf{a} = \sigma(\mathbf{z}) \tag{1.14}$$

Rezultatului din etapa liniară z i se aplică o funcție de activare notată cu σ care dă un rezultat intermediar a ce poate fi considerat exemplul stratului intermediar următor. Funcția de activare a unui neuron este o componentă esențială într-o rețea neuronală și joacă un rol crucial în procesarea informațiilor. Determină modul în care un neuron răspunde la stimulii pe care îi primește și generează o ieșire în funcție de aceștia.

- Stratul de ieșire
  Pe stratul de ieșire se regăsesc neuronii *perceptroni* sau *clasificatori* care se folosesc de caracteristicile oferite de ultimul strat intermediar pentru a lua decizii de recunoaștere. Aceștia prelucrează datele similar cu neuronii aflați pe straturile intermediare, diferența fiind funcția de activare utilizată. Pe stratul de ieșire se obțin rezultate optime pentru această aplicație dacă este utilizată funcția *softmax*. Rezultatele acesteia reprezintă probabilități de apartenență a exemplului dat la intrare la o anumită clasă. [5]

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{m} e^{z_j}} \tag{1.15}$$

Numărul de neuroni de pe stratul de ieșire este egal cu numărul de clase, de categorii, în care pot fi încadrate exemplele.

1.9.2 Învățarea supervizată

În această secțiune va fi analizat algoritmul de antrenare a rețelei neurale, atât pașii, vezi figura 1.4, cât și funcțiile implicate.

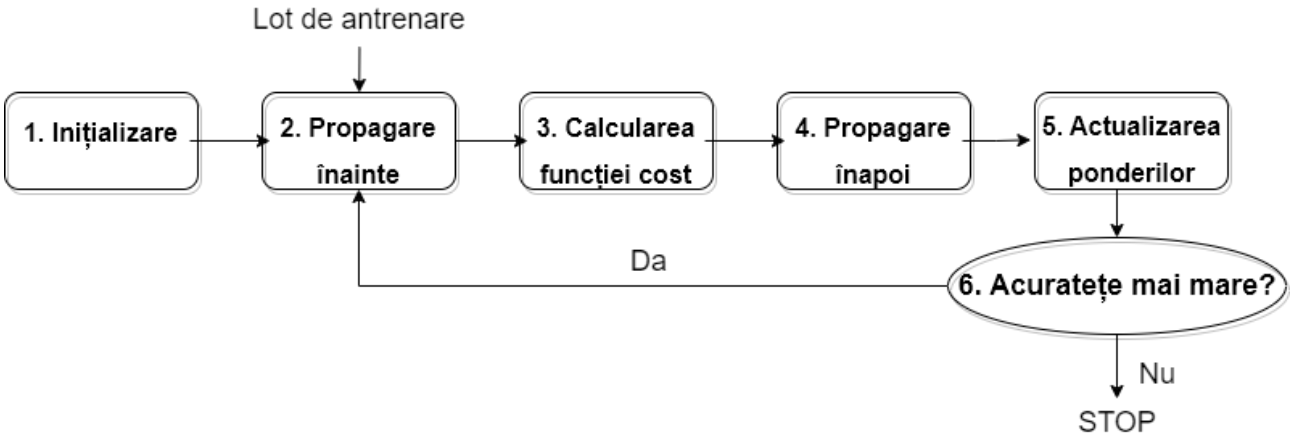

Figura 1.4: Algoritmul învățării supervizate

1. Inițializare
   Matricea de ponderi **W** și vectorul de bias ale rețelei neurale sunt inițializate cu valori aleatoare.

$$\mathbf{W} = \begin{pmatrix} w_{11} & w_{12} & \dots & w_{1m} \\ w_{21} & w_{22} & \dots & w_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ w_{n1} & w_{n2} & \dots & w_{nm} \end{pmatrix} \tag{1.16}$$

$$\mathbf{b} = [b_1, b2, \dots, b_m] \tag{1.17}$$

2. Propagarea înainte
   Primul exemplu este introdus în rețeaua neurală, iar acesta se propagă către ieșire. Calcularea valorii de ieșire pentru fiecare neuron se face prin înmulțirea ponderilor cu valorile de la intrare și aplicarea funcției de activare asociată.

$$\mathbf{y} = \sigma(\mathbf{W} \cdot \mathbf{x} + \mathbf{b}) \tag{1.18}$$

3. Calcularea funcției cost
   Cum am menționat anterior, MLP este o rețea neurală supervizată ce are ca scop minimizarea funcției cost. Funcția cost folosită este *entropia încrucișată multi-clasă* și este strâns legată de funcția de activare a neuronilor de pe stratul de ieșire. În formulă, N este numărul de clase și de neuroni de pe ultimul strat, **y** este ieșirea ideală, iar **p** este vectorul de probabilități rezultat în urma aplicării funcției de activare *softmax*. [6]

$$H(\mathbf{y}, \mathbf{p}) = -\sum_{i=1}^{N} y_i \log_2 p_i \tag{1.19}$$

4. Propagarea înapoi
   Valoarea funcției cost calculată la pasul anterior parcurge în sens invers rețeaua acesta fiind modul în care parametrii rețelei neurale se modifică pentru a putea minimiza funcția cost. Acest proces este cunoscut și sub denumirea de *Backpropagation*. [7]

5. Actualizarea ponderilor
   Ajustarea ponderilor în urma calculării funcției cost reprezintă procesul de rafinare a acestora, proces ce are ca scop obținerea unei valori mai mici a funcției cost pentru următorul exemplu.[8]

   Adaptive Moment Algorithm este cel mai des utilizat optimizator, având performanțe ridicate. Acest algoritm se bazează pe rate de învățare adaptive. Adam ia în considerare atât cumularea gradienților a iterațiilor precedente ale algoritmului, cât și variațiile acestora. [9]

6. Repetarea și condiția de oprire
   Pașii 2, 3, 4 și 5 se repetă pentru fiecare exemplu din lotul de antrenare. Întreg procesul poate fi reluat de mai multe ori în funcție de câte epoci au fost specificate.

Condiția de oprire o reprezintă evoluția metricii de evaluare. Dacă un anumit număr de epoci metrica nu se îmbunătățește sau chiar scade, antrenarea este oprită mai devreme, chiar dacă a fost menționat un număr mai mare de epoci. [10]

În continuare, este dezvoltat subiectul metricilor de performanță.

Cum am menționat, pentru a măsurara calitatea modelului de clasificare multi-clasă se pot folosi diferite metrici de performanță.

Pentru simplitatea înțelegerii conceptelor, metricile de performanță prezentate sunt particularizate pentru clasificarea binară. Pot fi ușor generalizate pentru o clasificare cu N clase.

Pentru înțelegerea acestora sunt expuse clasificările posibile ale datelor: FN, FP, TN, TP, noțiuni extrase din referința [11].

subsubsection

	clasa X reală	clasa $\overline{X}$ reală
clasa X clasificată	TP	FP
clasa $\overline{X}$ clasificată	FN	TN

Tabela 1.1: Modurile de clasificare

Tabelul 1.1 ajută la interpretarea categoriilor în care se pot încadra datele.

1. Categoriile pe care le dorim maximizate

   - True Positive - TP
     În această categorie se regăsesc datele care aparțin clasei X și au fost clasificate ca fiind în clasa X.
   - True Negative - TN
     În această categorie se regăsesc datele care nu aparțin clasei X și au fost clasificate ca nefăcând parte din clasa X.

2. Categoriile pe care le dorim minimizate

   - False Positive - FP
     În această categorie sunt datele care deși nu aparțin clasei de interes, au fost clasificate ca făcând parte din aceasta.
   - False Negative - FN
     În această categorie sunt datele care deși aparțin clasei de interes, au fost clasificate ca nefăcând parte din aceasta.

În secțiunea curentă sunt prezentate 4 dintre cele mai utilizate astfel de metrici care au la bază tipurile de clasificare.

- Precizie
  Precizia are ca rezultat procentajul în care datele clasificate ca făcând parte din clasa de

interes aparțin cu adevărat acesteia.

$$Precizie = \frac{TP}{TP + FP} \tag{1.20}$$

- Reamintire
  Reamintirea este procentajul în care datele care aparțin clasei de interes au fost clasificate ca făcând parte din aceasta.

$$Reamintire = \frac{TP}{TP + FN} \tag{1.21}$$

- Scor F1
  Scorul F1 este o metrică de performanță care îmbină *precizia* și *reamintirea*, reprezentând media armonică a acestora.

$$ScorF1 = 2 \cdot \frac{Precizie \cdot Reamintire}{Precizie + Reamintire} \tag{1.22}$$

- Acuratețe
  Cea mai des folosită metrică de performanță este acuratețea care poate fi explicată ca fiind procentajul de date clasificate corect. Este utilă pentru a evalua performanța generală deoarece îmbină și ea *precizia* și *reamintirea*.

$$Acuratețe = \frac{TP + TN}{FN + TP + FP + TN} \tag{1.23}$$

O matrice care evidențiază performanța unui model de clasificare multi-clasă este *matricea de confuzie*. În tabelul 1.2 poate fi vizualizată structura unei astfel de matrici.

Pe diagonala principală se regăsesc exemplele clasificate corect, care fac parte din clasa i și au fost clasificate ca făcând parte din clasa i. Valoarea de pe linia i și coloana j, când $i \neq j$, reprezintă numărul de exemple care deși aparțineau clasei j, au fost clasificate ca făcând parte din clasa i.

-	Clasa 0 reală	Clasa 1 reală	...	Clasa N reală
Clasa 0 clasificată	TP	FN	...	FN
Clasa 1 clasificată	FN	TP	...	FN
...	...	...	...	...
Clasa N clasificată	FN	FN	...	TP

Tabela 1.2: Matrice de confuzie

1.9.3 Exemple de funcții de activare

Cum a fost menționat anterior, funcția de activare se notează cu σ și reprezintă etapa neliniară din prelucrarea datelor de către un neuron. În lista de mai jos sunt prezentate 5 astfel

de funcții. [12]

- Liniară
  Funcția liniară de activare este simplă, aceasta având rezultatul egal cu valoarea de la intrare. Graficul poate fi vizualizat în figura 1.5.

$$\sigma(x) = x \tag{1.24}$$

- ReLU
  Această funcție de activare are o caracteristică importantă: nu activează toți neuronii în același timp. Neuronii activi sunt doar cei care în etapa liniară au avut un rezultat pozitiv. Această caracteristică poate fi un avantaj privind din punct de vedere computațional, dar se poate transforma într-o problemă atunci când ponderile anumitor neuroni nu sunt actualizate din cauza gradientului nul, ducând la *moartea* acestora. Graficul poate fi vizualizat în figura 1.6.

$$\sigma(x) = \max(0, x) \tag{1.25}$$

- SELU
  Folosirea funcției SELU duce la păstrarea datelor normalizate, așa cum au fost introduse în rețea. Spre deosebire de ReLU, această funcție de activare poate avea și rezultate negative ceea ce reprezintă un avantaj ținând cont de valorile carcateristicilor din baza de date. α și λ au valori predefinite. Graficul poate fi vizualizat în figura 1.7.

$$\sigma(x) = \lambda \cdot \begin{cases} \alpha(e^x - 1) & \text{dacă } x < 0 \\ x & \text{dacă } x \geq 0 \end{cases} \tag{1.26}$$

- Sigmoid
  Funcția sigmoid are ca domeniu de definiție întreg spațiul numerelor reale, iar codomeniul va fi intervalul $(0; 1)$. Cu cât valoarea de la intrare este mai mare, cu atât rezultatul funcției sigmoid va fi mai apropiat de 1. Cu cât valoarea de la intrare este mai mică, cu atât rezultatul funcției sigmoid va fi mai apropiat de 0. Graficul poate fi vizualizat în figura 1.8.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{1.27}$$

- Tangentă hiperbolică
  Este o funcție care are codomeniul intervalul $(-1; 1)$ și are o formă similară cu cea sigmoid. Faptul că valorile de ieșire ale funcției sunt simetrice față de origine devine un avantaj atunci când datele de intrare au și ele o scală simetrică în jurul valorii 0. Graficul poate fi vizualizat în figura 1.9.

$$\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{1.28}$$

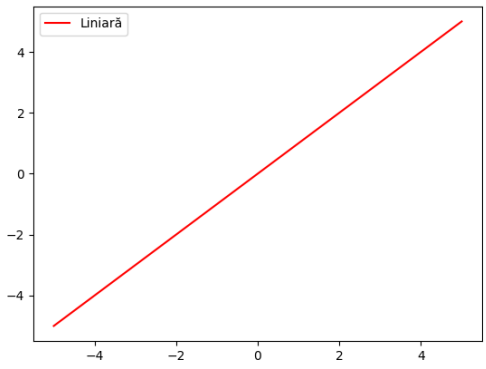

Figura 1.5: Funcția de activare liniară

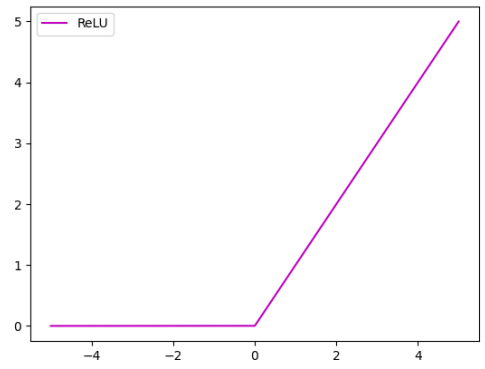

Figura 1.6: Funcția de activare ReLU

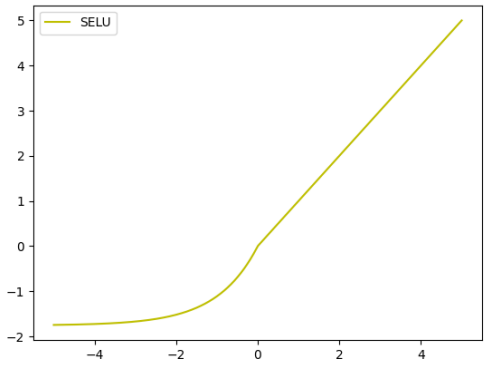

Figura 1.7: Funcția de activare SELU

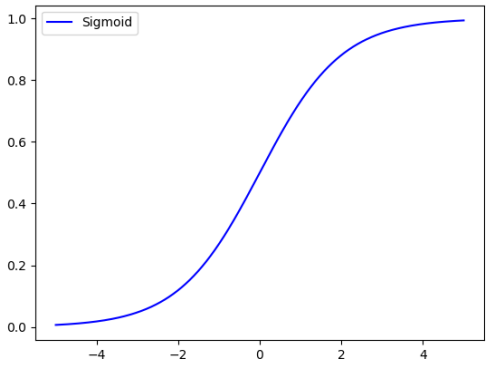

Figura 1.8: Funcția de activare sigmoid

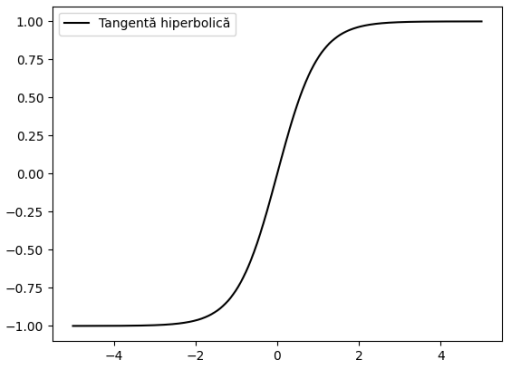

Figura 1.9: Funcția de activare tangentă hiperbolică

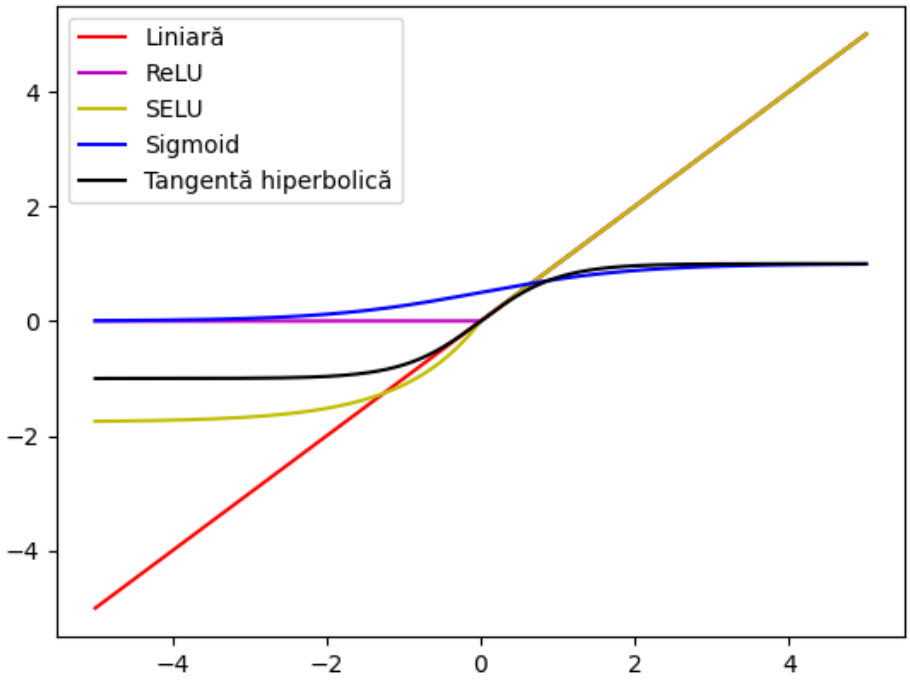

Figura 1.10: Toate funcțiile de activare

În figura 1.10 se regăsesc toate cele 5 funcții de activare folosite în procesul de optimizare pentru vizualizarea comparativă.

2 Sistemul de achiziție de date

Pentru a prelua date despre mișcarea membrelor inferioare, în cazul nostru despre accelerație, se dorește realizarea unui sistem ce poate atinge performanțele dorite. Sunt urmărite îndeplinirea următoarelor cerințe:

- Frecvență de eșantionare de cel puțin 500Hz
- Comunicație fără intermediul firelor
- Consum mic de energie
- Viteză de prelucrare în concordanță cu frecvența de eșantionare
- Compact și fiabil

Alegem ca senzor de mișcare accelerometrul ADXL345 ce ne poate oferi o frecvență de eșantionare satisfăcătoare, iar ca master folosim microcontrolerul ESP32 ce dispune de comunicație fără fir. Se ia în considerare adăugarea unui ecran, dar acesta ar ridica consumul de energie ceea ce nu este de dorit. Sistemul este alimentat de un acumulator Li-Po ce poate fi reîncărcat cu ajutorul unui modul, TP4056. De asemenea, pentru ca datele să ajungă corect la modulul de preprocesare este folosit un calculator cu performanțe ridicate ce poate prelua datele cu viteză optimă.

Se ajunge la un sistem ce are schema bloc ilustată în figura 2.1.

Figura 2.1: Schema bloc a sistemului de achiziție a datelor

Sistemul are atât o componentă hardware, cât și una software. Acestea sunt prezentate în cele ce urmează.

2.1 Componenta hardware

În acest capitol sunt detaliate funcționalitățile utilizate ale componentelor ce se regăsesc în figura 2.2, dar și modul în care acestea comunică.

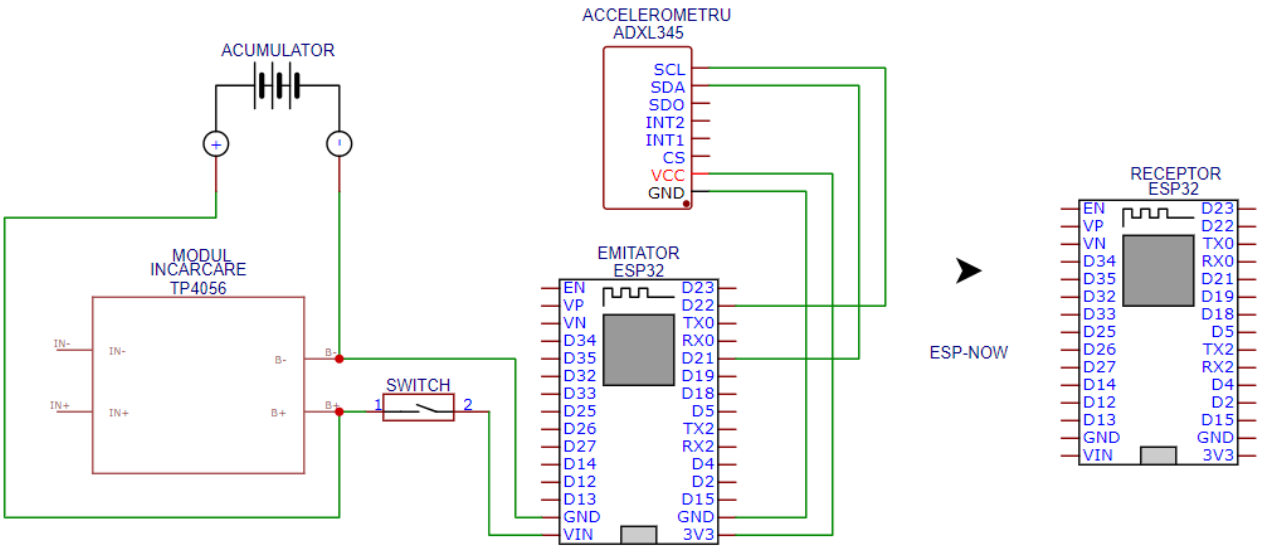

Figura 2.2: Schema electrică a sistemului de achiziție de date

2.1.1 Senzorul de mișcare ADXL345

Accelerometrul digital ADXL345 este folosit pentru a detecta miscarea pe cele 3 axe de libertate. Funcționează pe baza unei mase inerțiale, acesta fiind un accelerometru capacitiv. Mișcarea acesteia determină semnale electrice proporționale cu accelerația resimțită pe fiecare axă. În figura 2.3 se poate observa modul în care masa inerțială se opune mișcării, astfel fiind create tensiuni diferite.

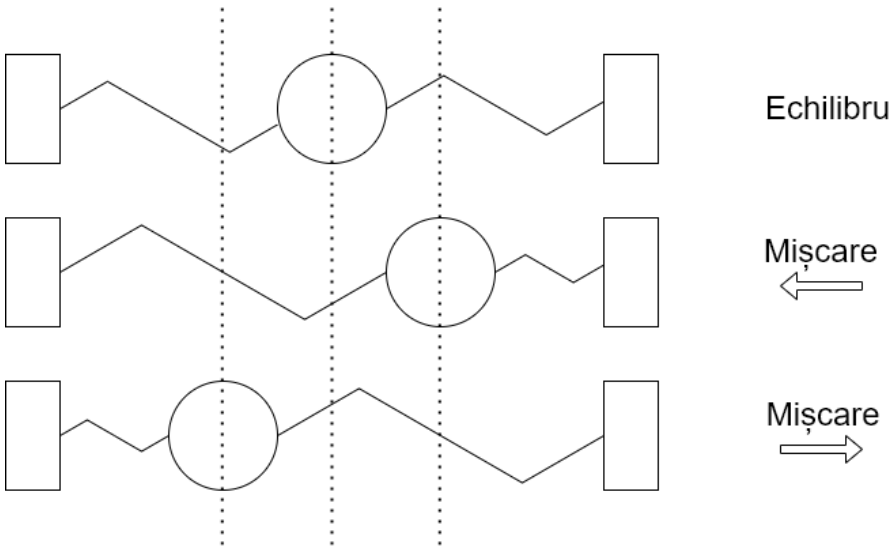

Figura 2.3: Comportamentul masei inerțiale

Pentru a înțelege cum variază capacitățile condensatoarelor avem figura 2.4. Capacitatea unui condensator este invers proporțională cu distanța dintre armăturile acestuia.

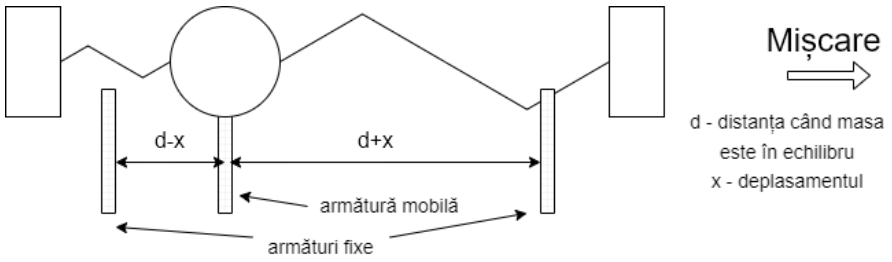

Figura 2.4: Principiul de funcționare a accelerometrului capacitiv

Acest sistem este implementat pe toate cele 3 axe. [13]

Sensibilitatea, intervalul în care poate lua valori accelerația, este ajustabil. Valorile posibile sunt:

- ± 2g - valorile accelerației sunt precise, fiind simțite chiar și cele mai fine mișcări
- ± 4g - valoare intermediară care reduce precizia, dar crește domeniul în care poate lua valori accelerația
- ± 8g - valoare aleasă în lucrarea de față, fiind un compromis preferat între precizie și mărimea domeniului în care poate lua valori accelerația
- ± 16g - valorile accelerației au cea mai scăzută precizie, dar aceasta poate avea valori foarte mari

Este de menționat că g reprezintă accelerația gravitațională, iar în acest caz are valoarea de $9.8m/s^2$.

De asemenea, un alt parametru care poate fi setat este frecvența de eșantionare, aceasta putând lua doar anumite valori. Cele mai mici sunt: 0.1Hz, 0.2Hz, 0,39Hz, iar frecvența maximă de eșantionare este 3.2kHz. Dorind o valoare de aproximativ 500Hz pentru aplicația în care este folosit accelerametrul, valorile care puteau satisface această cerință au fost 400Hz sau 800Hz. Deoarece valoarea de 400Hz ar fi scăzut performanțele, am ales pentru frecvența de eșantionare valoarea de 800Hz. O valoare mai mare ar fi mărit volumul de date, dar redundanța acestora ar fi foat ridicată. [14]

Accelerometrul este slave, răspunzând și fiind controlat de microcontroler. Cu ajutorul convertorului analog - digital integrat în dispozitiv, informația poate fi preluată prin intermediul protocolului I2C. [15]

Tabelul 2.1 oferă informații despre modul de funcționare al lui ADXL345.

VCC	3.3V
SDA	High - 3.3V/ Low - 0V
SCL	High - 3.3V/ Low - 0V
GND	0V

Tabela 2.1: Funcționalitatea lui ADXL345

2.1.2 Protocolul I2C

Interfața digitală I2C este un protocol de comunicație sincronă care permite transferul de date între dispozitive, în cazul de față, între accelerometru și microcontroler. Sunt utilizate două linii de comunicație:

- Lina de ceas (SCL) are rolul de a sincroniza transferul datelor între cele două dispozitive electronice. Pentru aplicația implementată este folosită o frecvență de 400kHz.
- Linia de date (SDA) este în concordanță cu frecvența la care este setată linia de ceas, fiind capabilă să transmită date cu o viteză de 400 kbps.

Informațiile specificate determină performanța interfeței, iar astfel protocolul în acest proiect se încadrează în limitele pentru I2C rapid. [16]

2.1.3 Microcontrolerul ESP32 - emițător

ESP32 este platforma de dezvoltare folosită pentru gestionarea activității accelerometrului ADXL345, având rolul de master în această configurație. Pinii digitali D21 și D22 sunt conectați la SDA, respectiv la SCL, iar astfel informația oferită de senzorul de mișcare este preluată de microcontroler.

Totodată, acesta este și emițător în raport cu al doilea ESP32 care are rolul de a recepționa informația, cele două microcontrolere formând o topologie peer to peer.

VCC	3.7V
D21	High - 3.3V/ Low - 0V
D22	High - 3.3V/ Low - 0V
3V3	3.3V
GND	0V

Tabela 2.2: Funcționalitatea lui ESP32 - emițător

Pentru transferul datelor preluate de la senzorul de mișcare se folosește protocolul ESP-NOW. [17]

2.1.4 Protocolul ESP-NOW

ESP-NOW este un mod de comunicație ce poate funcționa doar între dispozitive de tipul ESP32. Este un protocol ce permite transferul fără fir a datelor, fiind optimizat pentru transmitere eficientă. Este utilizată o topologie peer-to-peer, care facilitează schimbul de date fără susținerea altei infrastructuri Wi-Fi sau implicarea unor puncte de acces cum ar fi routerele.

De asemenea, pachetele de date trimise prin ESP-NOW pot avea o dimensiune maximă de 250 de octeți. În ipoteza în care pachetele au dimensiuni mici, este necesară o viteză de trasmisie ridicată, astfel putând fi cauzat efectul de jitter. Acest fenomen se manifestă prin

întârzieri nedorite, iar periodicitatea evenimentelor are de suferit. Pentru ca transmiterea informației să fie optimă, ne dorim un echilibru între viteză și dimensiunea pachetelor. [18]

Protocolul utilizează frecvența de 2.4GHz, frecvență ce face parte din banda ISM care este destinată utilizării în domeniul industrial, al cercetării sau în cel medical. Banda ISM este cunoscută ca fiind o bandă de frecvențe care nu necesită licență, făcând parte din spectrul radio liber. Acest fapt este un avantaj deoarece nu implică costuri suplimentare, dar în același timp transferul datelor poate deveni dificil din cauza interferențelor ce pot apărea cu alte dispozitive din apropiere ce utilizează aceeași frecvență, fenomen nedorit. [19]

2.1.5 Acumulator Li-Po

Acumulatorul Li-Po alimentează sistemul prezentat anterior. Are o tensiune nominală de 3.7V, acesta putând atinge 4.2V când este încărcat complet, iar atunci când este descărcat tensiunea de întrerupere este 2.75V. Capacitatea acestuia este de 1200 mAh ceea ce asigură o funcționare de cel puțin o oră a sistemului.[20]

2.1.6 Modulul de încărcare TP4056

Încărcarea în siguranță a acumulatorului Li-Po se face cu ajutorul modulului TP4056. Are rolul de a menține o tensiune de încărcare constantă și de a preveni supraîncărcarea. Cele două leduri ne permit monitorizarea procesului. Cât timp acumulatorul încă se încarcă se poate observa LED roșu aprins, iar când încărcarea este completă se aprinde cel albastru. Conectarea la rețeaua electrică se face cu ajutorul portului microUSB.[21]

2.1.7 Microcontrolerul ESP32 - receptor

Cel de-al doilea microcontroler ESP32 este configurat să recepționeze pachetele de informație trimise prin ESP-NOW, făcând și el parte din topologia peer to peer. Acesta este conectat la un calculator căruia îi trimite datele folosindu-se de un cablu USB ce are rolul și de a alimenta microcontrolerul.

2.1.8 Comunicația USB

Transmiterea informației de la microcontrolerul ESP32 care este configurat ca receptor se realizează prin intermediul unei conexiuni seriale. Sistemul de operare al calculatorului recunoaște portul, iar astfel pachetele de date pot fi preluate.

În figura 2.5 se pot observa componentele prezentate anterior.

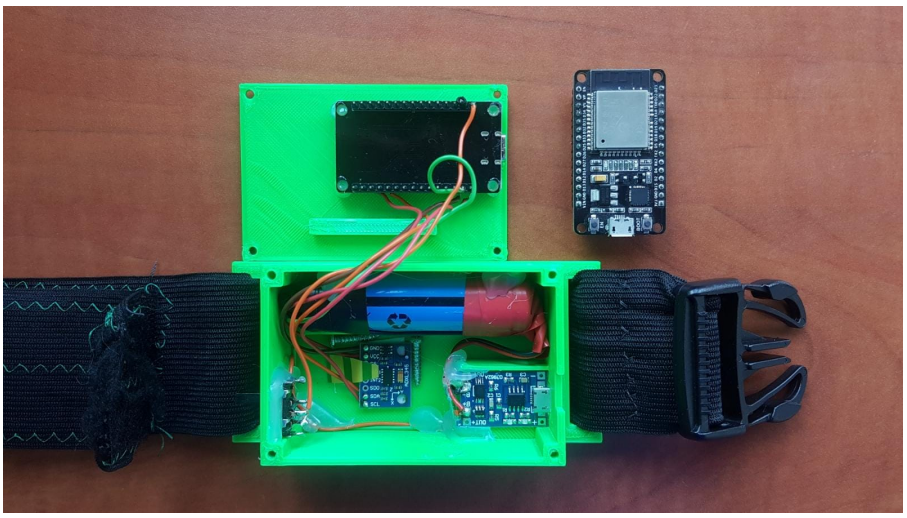

Figura 2.5: Sistemul de achiziție de date

2.2 Componenta software

Emițător

Programul încărcat pe microcontrolerul ESP32 cu rolul de emițător folosește câteva librării ce permit preluarea datelor de la senzorul de mișcare, prelucrarea și transmiterea acestora.

Configurarea constă în setarea parametrilor de funcționare doriți pentru ADXL345, fiind aleasă frecvența de eșantionare de 800Hz și intervalul în care poate lua valori accelerația la ± 8g. În cele din urmă, este stabilită conexiunea prin ESP-NOW cu receptorul.

În buclă preluăm datele cu o frecvență mai mică decât cea de eșantionare pentru a micșora redundanța datelor. Frecvența cu care sunt recepționate datele de la senzor este de aproximativ 640Hz. În fiecare secundă ajung la ESP32 640 de eșantioane, iar fiecare eșantion este format din 3 numere: accelerația pe axa X, accelerația pe axa Y și accelerația pe axa Z. Se producea un fenomen de jitter când eșantioanele erau trimise prin ESP-NOW imediat ce au fost preluate de la senzorul de mișcare, iar soluția preferată a fost formarea unor pachete mai mari de eșantioane. Astfel, sunt concatenate într-un pachet 10 eșantioane, fiecare conținând cele 3 valori ale accelerației. Pachetul este apoi transmis către ESP32 receptor. Se înțelege că frecvența cu care sunt transmise pachetele este de 64Hz, dar este păstrată frecvența de 640Hz cu care au fost preluate datele. În cele ce urmează, frecvența de eșantionare v-a fi considerată ca fiind 640Hz.

Receptor

Codul încărcat pe microcontrolerul ESP32 cu rolul de receptor îi permite acestuia preluarea pachetelor trasmise prin ESP-NOW și trimiterea acestora la calculator cu ajutorul serialei.

3 Preprocesarea datelor

În acest capitol este prezentat modul în care sunt preluate datele de către calculator de pe interfața serială și procesarea acestora până în punctul în care pot fi trimise rețelei neurale.

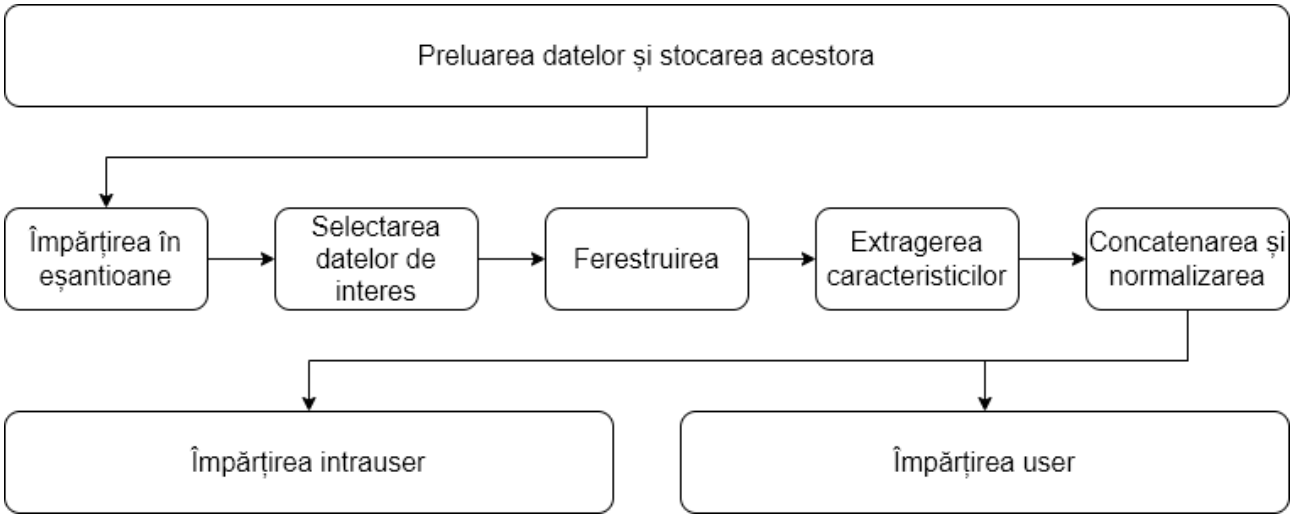

Figura 3.1: Modulul de preprocesare

3.1 Baza de date

3.2 Preluarea datelor

Pentru a putea procesa datele este nevoie ca acestea să fie în standardele așteptate și stocate într-un format convenabil. Se urmărește îndeplinirea aspectelor prezentate în lista următoare:

- Acces la datele primite pe portul serial
  Menționând numele portului la care este conectat cablul USB și viteza de transmisie a datelor prin intermediul acestuia, se reușește accesul la informații.
- Timp de achiziție fix
  Este ales un timp de 90 de secunde în care programul să preia date. În multe cazuri, în timpul procesului parcurs pentru baza de date, s-a observat că acesta este mai mult decât suficient.
- Controlarea momentului în care începe preluarea pachetelor
  După ce a fost configurat portul, programul așteaptă apăsarea tastei *space* pentru a porni cronometrul și a începe primirea datelor.
- Primirea unui pachet de date la intervale egale de timp
  Ținând cont că frecvența cu care sunt transmise pachetele este 64Hz, se așteaptă un grup de 10 eșantioane la fiecare 15 milisecunde.
- Verificarea datelor înainte de salvare, urmărind intervalul în care acestea iau valori, periodicitatea și numărul lor

Pentru a eficientiza verificarea datelor s-a ales să se utilizeze un grafic ce surprinde modul în care variază accelerația pe cele 3 axe în timpul în achiziția are loc.

- Introducerea manuală a numelui fișierelor pentru a evita eventuale erori
  În condițiile în care graficul este satisfăcător, programul va solicita introducerea numelui cu care dorim să fie salvate.
- Salvarea în formate ușor de manipulat
  În cele din urmă, datele vor fi salvate în două formate: în format CSV și în format *npy*. Aceste fișiere sunt folosite pentru prelucrarea ulterioară.

3.3 Baza de date

Programul explicat în secțiunea anterioară a fost folosit pentru a alcătui baza de date. În baza de date se regăsesc 22 de subiecți, atât de gen masculin, cât și de gen feminin, cu vârsta medie de 25 de ani. Tuturor li s-a fixat dispozitivul de achiziție pe membrul inferior drept, deasupra genunchiului, aspect exemplificat în imaginea 3.2.

Figura 3.2: Fixarea dispozitivului de achiziție

Pentru fiecare persoană au fost salvate 7 fișiere în format *npy* și 7 în format *csv*. S-a dorit alegerea unor clase care să nu constrângă prelevarea datelor să aibă loc într-un anumit spațiu, o altă clasă propusă fiind urcarea scărilor. Cele 7 clase în care se regăsesc fișierele sunt următoarele:

- Clasa 0 - stat - grafice în figura 3.3
- Clasa 1 - mers în linie dreaptă cu viteză constantă pe teren plat - grafice în figura 3.4

- Clasa 2 - genuflexiuni - grafice în figura 3.5
- Clasa 3 - jumping jacks - grafice în figura 3.6
- Clasa 4 - alergare pe loc - grafice în figura 3.7
- Clasa 5 - ridicarea piciorului în plan frontal - grafice în figura 3.8
- Clasa 6 - dans liber; mișcare în mod aleator - grafice în figura 3.9

Pentru genul masculin se folosește 1, iar pentru genul feminin numărul 2.

Numele fișierelor au formatul următor $Clasa\_IDsubiect\_Gen$ la care se adaugă extensia tipului de date salvate.

$$\text{align Clasa} \in \{0, 1, 2, 3, 4, 5, 6\}$$
$$\text{IDsubiect} \in \{01, 02, 03, \cdots, 20, 21, 22\}$$
$$\text{Gen} \in \{1, 2\}$$

În următoarele grafice se regăsesc 4 secunde din fiecare fișier a unui subiect ales aleator. În stânga este imaginea datelor brute, cum au fost trimise de accelerometru, iar în dreapta sunt aceleași date doar că trecute printr-un filtru de mediere.

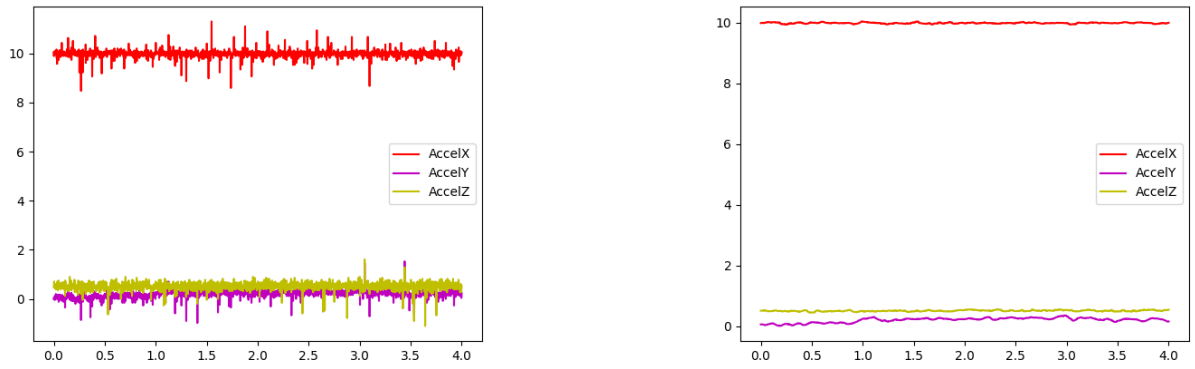

Figura 3.3: Clasa 0 - stat - date brute și date mediate

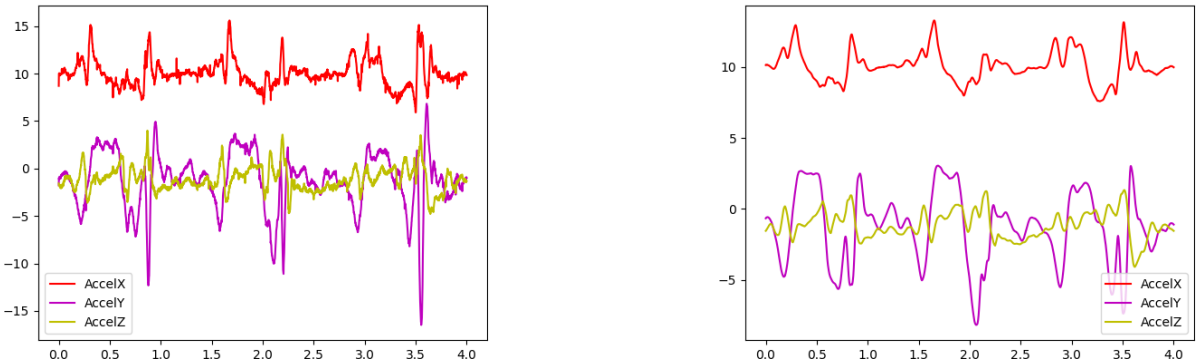

Figura 3.4: Clasa 1 - mers - date brute și date mediate

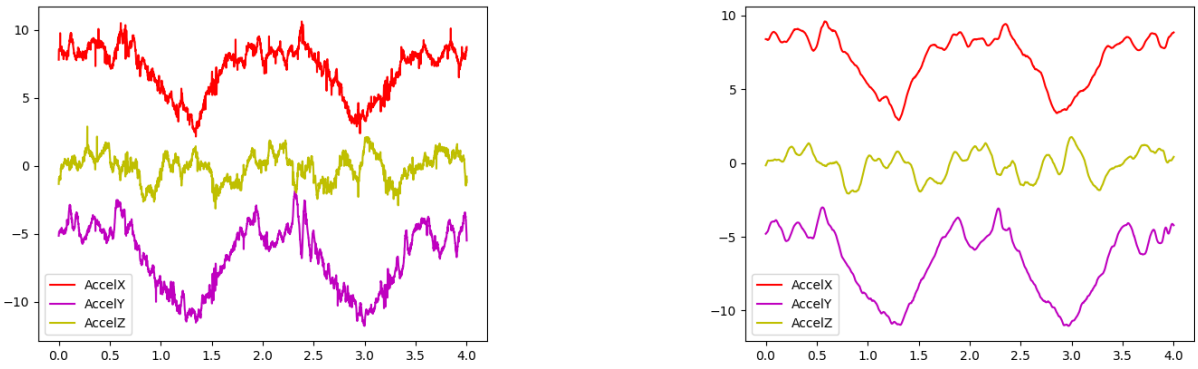

Figura 3.5: Clasa 2 - genuflexiuni - date brute și date mediate

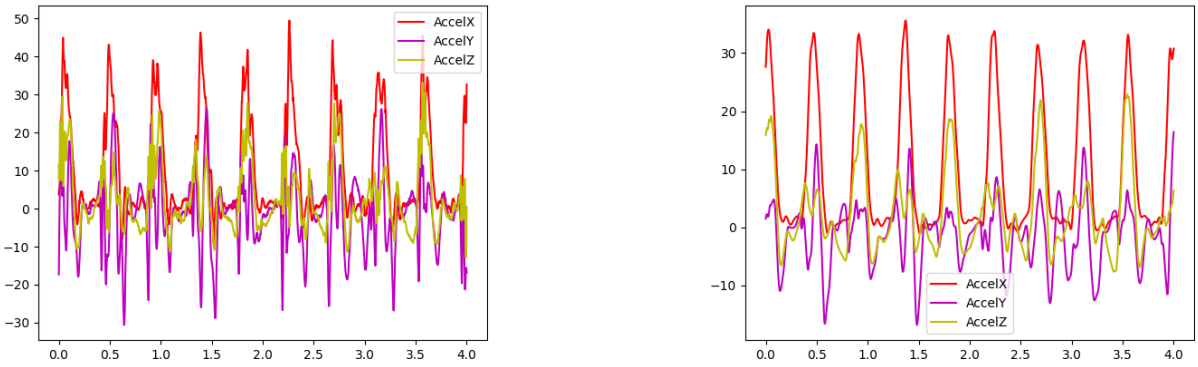

Figura 3.6: Clasa 3 - jumping jacks - date brute și date mediate

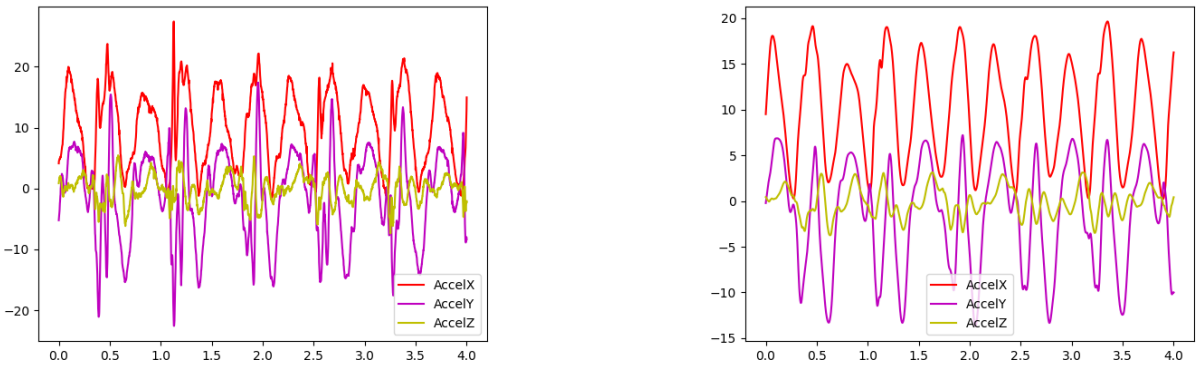

Figura 3.7: Clasa 4 - alergare pe loc - date brute și date mediate

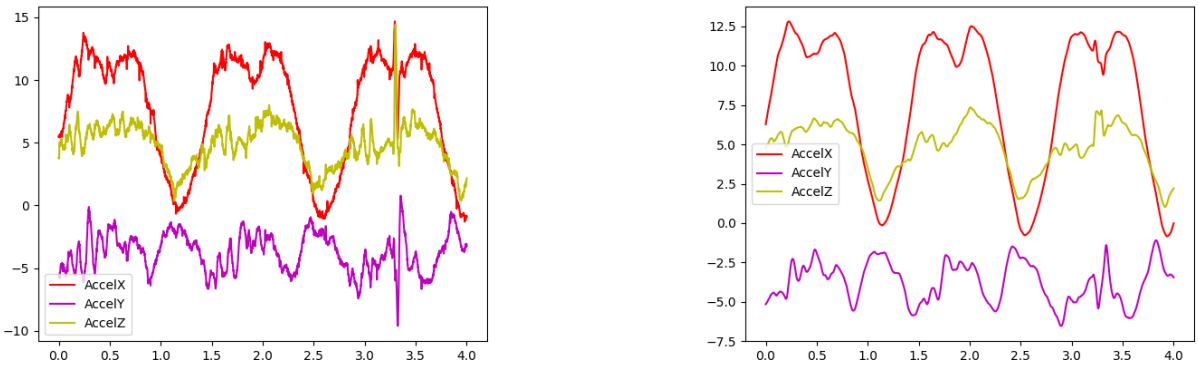

Figura 3.8: Clasa 5 - ridicare picior în plan frontal - date brute și date mediate

Primele 6 clase au un comportament periodic, acest aspect fiind ușor de observat în graficele datelor mediate.

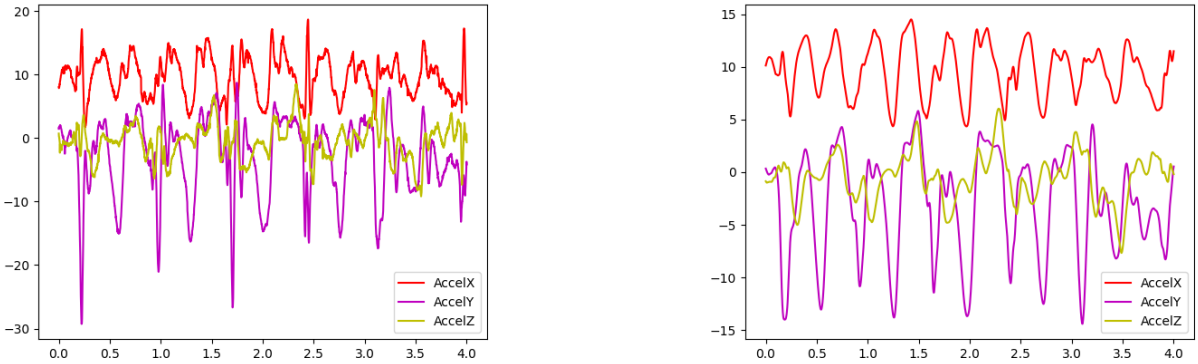

Figura 3.9: Clasa 6 - dans liber; mișcare în mod aleator - date brute și date mediate

Ultima clasă, mișcare în mod aleator, este sigura în care semnalele accelerațiilor sunt neperiodice.

3.4 Preprocesarea

În continuare sunt explicați pașii prin care trec datele neprelucrate pentru a ajunge în forma dorită de rețeaua neurală.

Împărțirea în eșantioane

Fiecare linie preluată de calculator pe portul serial reprezintă un pachet și este format din 30 de valori, acesta conținând 10 eșantioane concatenate. Pentru înțelegere se ia în continuare un exemplu, procesul fiind ilustrat în figura 3.10.

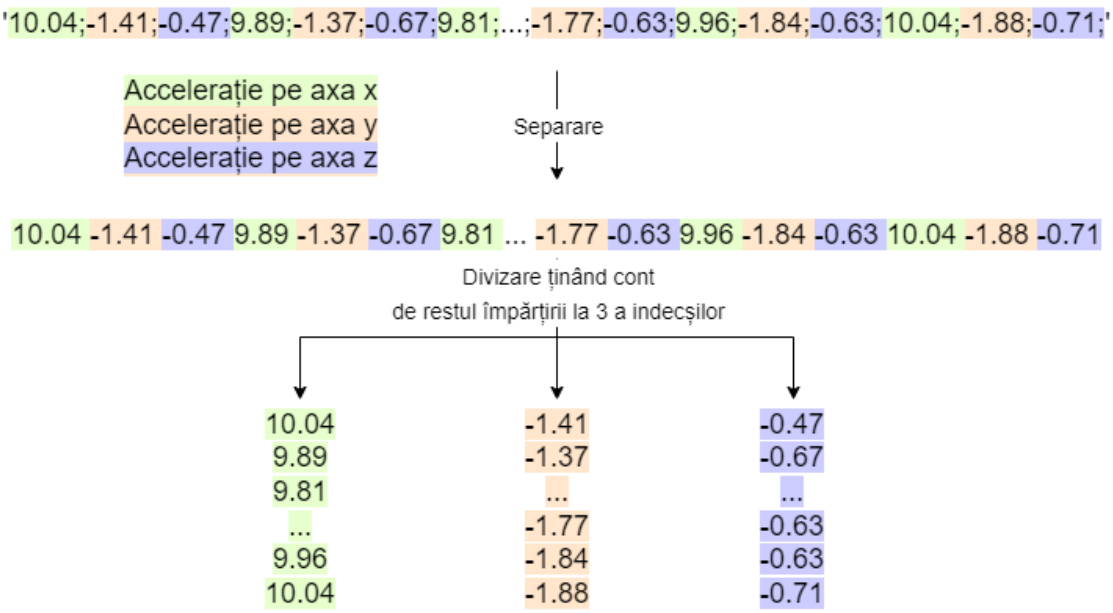

Figura 3.10: Împărțirea datelor

Identic se procedează pentru fiecare linie primită pe portul serial. Datele divizate se concatenează, în final având o matrice cu 3 linii: accelerațiile pe axa x, accelerațiile pe axa y, accelerațiile pe axaz. Dacă selectăm o coloană a acestei matrici vom avea informația despre accelerație pe toate cele 3 axe la un anumit moment de timp. Numărul de coloane se poate calcula înmulțind numărul de eșantioane din fiecare pachet cu frecvența de transmitere a pachetelor și cu timpul stabilit în care are loc procesul de preluare a datelor.

$$10 \cdot 64 \cdot 90 = 57600$$

Selectarea datelor de interes

Au fost situații în care subiecții nu au putut efectua mișcarea timp de 90 de secunde fiind solicitant din punct de vedere fizic. În alte cazuri, în partea de început a procesului de preluare a datelor puteau fi observate mici diferențe privind periodicitatea. S-a decis selectarea unui număr de eșantioane mai mic, corespunzător unui timp de 60 de secunde de preluare a datelor. La finalul acestui pas, fiecare matrice are 3 linii și 38400 de coloane.

$$10 \cdot 64 \cdot 60 = 38400$$

Ferestruirea

În continuare ne dorim fiecare linie să fie împărțită în porțiuni mai mici pentru care se pot calcula mai departe carcteristicile. Astfel, este utilizată o fereastră glisantă de 0.5 secunde pe care o mutăm cu pas de 0.25 secunde pentru a obține o suprapunere de 50%. În 0.5 secunde sunt cuprinse 320 de eșantioane de pe fiecare axă a accelerației, acestea fiind concatenate și formând o singură linie ulterior. Pentru a deduce câte linii va avea noua matrice putem împărți

numărul de valori ale accelerației regăsite pe fiecare axă la pasul cu care este deplasată fereastra.

$$\left[\frac{38400 - 0.5 \cdot 640}{(0.5 - 0.25) \cdot 640}\right] + 1 = 239$$

Se ajunge ca fiecare fișier să aibă o structură de matrice cu 239 de linii și 960 de coloane.

În paralel, sunt formate matricele în care este reținută clasa la care aparține fiecare fereastră, informație extrasă din numele fișierelor. Acestea au o singură coloană și 239 de linii.

3.5 Extragerea caracteristicilor

Pentru a urmări evoluția accelerației în timpul mișcărilor se folosesc 5 caracteristici care sunt aplicate fiecărei ferestre în parte. A se înțelege că pe o linie matricea anterioară are 3 ferestre.

1. Prima caracteristică calculată este media absolută a eșantioanelor.

$$MAV(x) = \frac{1}{320} \sum_{i=0}^{320-1} |x_i| \tag{3.1}$$

2. A doua caracteristică este rate trecerilor prin zero a semnalului. În situația în care două valori consecutive au diferența mai mare decât un prag α și semnele acestora sunt diferite, valoarea caracteristicii este incrementată.

$$ZC(x) = |\{i : (|x_i - x_{i-1}| \geq \alpha) \wedge (sgn(x_i) \neq sgn(x_{i-1}))\}| \tag{3.2}$$

3. Cea de-a treia caracteristică este *skewness*.

$$Skewness(x) = \frac{1}{320} \sum_{i=0}^{320-1} \left(\frac{x_i - \overline{x}}{\sigma}\right)^3 \tag{3.3}$$

4. O altă caracteristică o reprezintă lungimea semnalului.

$$WL(x) = \sum_{i=1}^{320-1} |x_i - x_{i-1}| \tag{3.4}$$

5. Ultima cracteristică este deviația standard, dispersia datelor selectate.

$$\sigma = \sqrt{\frac{1}{320} \sum_{i=0}^{320-1} (x_i - \overline{x})^2} \tag{3.5}$$

În formulele prezentate anterior N, numărul total de valori dintr-o fereastră, a fost

înlocuit cu 320.

Fiecare dintre cele 3 ferestre regăsite pe o linie a matricei vor fi înlocuite cu cele 5 caracteristici, iar numărul coloanelor scade de la 960 la 15. Numărul de linii rămâne 239.

Se remarcă faptul că extragerea carcteristicilor a fost făcută pe baza datelor brute, fără a intraduce o etapă de mediere.

3.6 Concatenarea și standardizarea

În acest punct baza de date are forma a $7 \cdot 22$ de fișiere de tip *npy* (NumPy) fiecare fiind o matrice cu 239 de linii și 15 coloane. În continuare acestea sunt concatenate pe axa verticală, ajungându-se la o matrice care conține toate informațiile necesare. Numărul de linii crește la $36806 = 239 \cdot 22 \cdot 7$, iar numărul de coloane se păstrează.

Matricele care păstrează informația despre clasa în care se regăsește fiecare linie sunt și ele concatenate având grijă să fie respectată ordinea în care au fost așezate matricele în care sunt stocate caracteristicile.

Pentru a reduce intervalul numeric în care se regăsesc caracteristicile se alege normalizarea acestora. Se doresc numere cu valori mici pentru a optimiza resursele necesare antrenării rețelei neurale. Astfel, valorile regăsite pe fiecare coloană sunt aduse într-o distribuție are tinde a fi normală, de medie 0 și dispersie 1, sunt standardizate. Dacă volumul de date ar fi fost mai mare, distribuția era gaussiană, iar majoritatea datelor ar fi fost cuprinse în intervalul $[-3, 3]$, conform regulii celor 3 sigma. [22]

În figura 3.11 se poate observa cum media absolută a datelor avea inițial valoarea 10, iar în urma standardizării media a devenit 0. În figurile 3.12, 3.13 și 3.14 se fac observații similare pentru skewness, lungimea semnalului și dispersia valorilor.

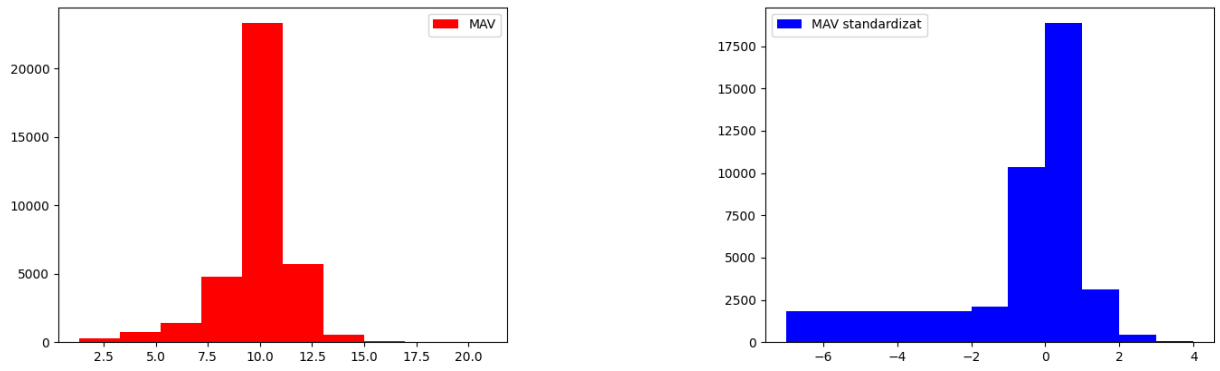

Figura 3.11: Media absolută - valori brute și valori standardizate

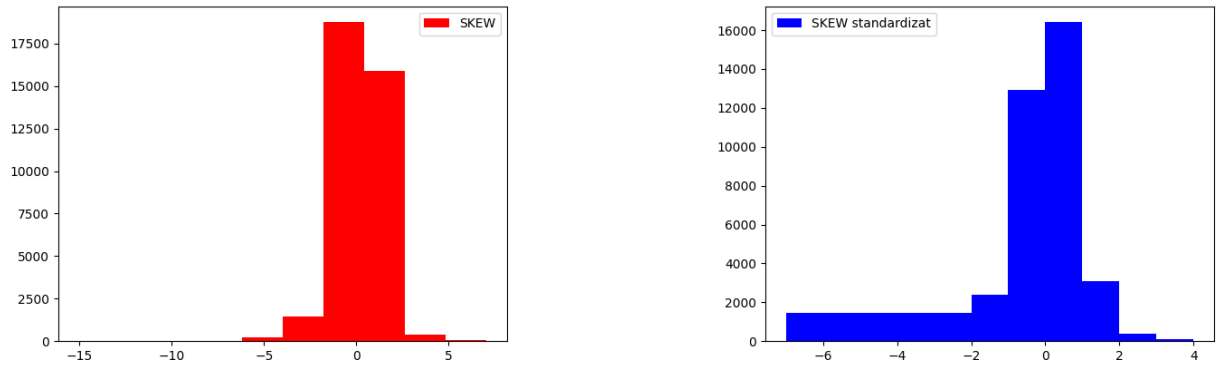

Figura 3.12: Skewness - valori brute și valori standardizate

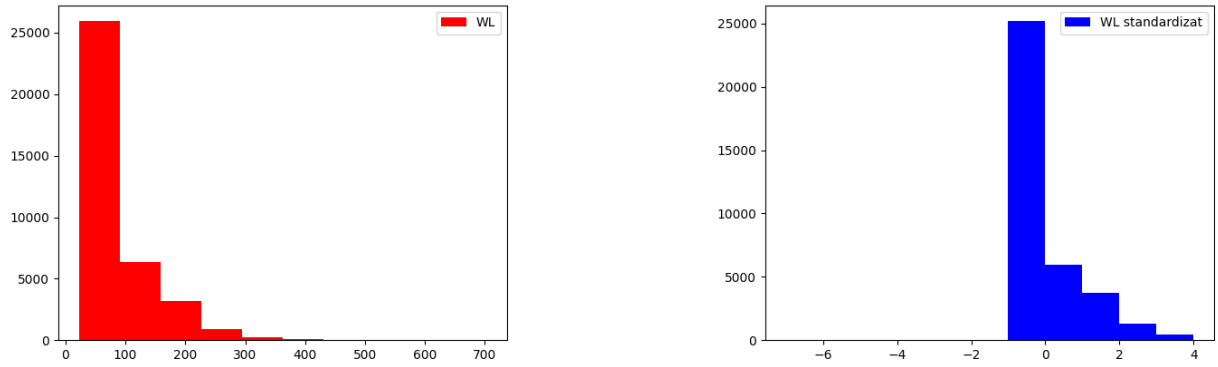

Figura 3.13: Lungimea semnalului - valori brute și valori standardizate

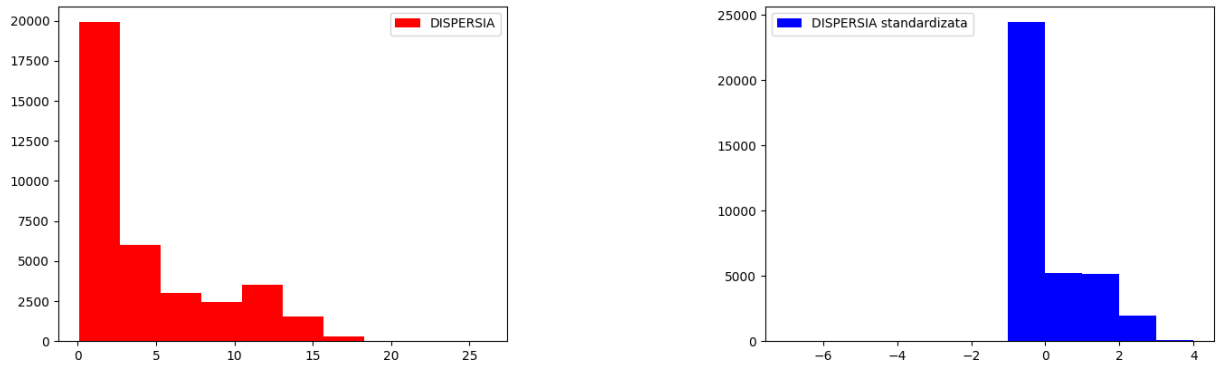

Figura 3.14: Dispersia - valori brute și valori standardizate

4 Rezultate experimentale

Pentru aplicația prezentată în această lucrare este potrivită folosirea unei rețele neurale supervizate, fiecare exemplu din lotul de antrenare având atribuit un număr care reprezintă codificarea clasei din care acesta face parte. Acest număr poartă denumirea de *etichetă* și simbolizează ieșirea ideală a rețelei neurale. Eticheta împreună cu ieșirea estimată sunt folosite pentru a obține un semnal de eroare care se dorește minimizat, acesta fiind folosit pentru ajustarea ponderilor sinaptice.

Arhitectura implementată în această lucrare este *perceptronul multinivel*.

4.1 Împărțirea datelor

Pentru a pregăti datele în vederea realizării rețelei neurale, se dorește împărțirea acestora în următoarele categorii:

- Lotul de antrenare - procentaj de 70%
  Acesta este setul de date folosit pentru antrenarea modelului.
- Lotul de validare - procentaj de 20%
  Acesta este un set de date utilizat în timpul antrenării pentru a ajusta hiperparametrii modelului și pentru a evalua performanța atinsă în etape intermediare.
- Lotul de testare - procentaj de 10%
  Acesta este setul de date utilizat pentru a evalua performanța finală a modelului.

Cele două moduri în care aceste seturi pot fi alese sunt explicate în cele ce urmează. [23]

Intrauser

Modul intrauser de împărțire a datelor se bazează pe selectarea din fiecare fișier a procentajelor de date menționate. A se înțelege prin fișier o anumită clasă a unui anumit subiect.

Cunoscând că un fișier cuprinde 239 de exemple, toate acestea sunt împărțite astfel: 167 în lotul de antrenare, 48 în lotul de validare, 24 în lotul de testare. Procentajele obținute sunt:

- $\frac{167}{239} \cdot 100 = 69.87\%$ lotul de antrenare
- $\frac{48}{239} \cdot 100 = 20.08\%$ lotul de validare
- $\frac{24}{239} \cdot 100 = 10.04\%$ lotul de testare

Se obțin loturile de date cu structura expusă în tabelul 4.2.

-	X	Y
Antrenare	(25718, 15)	(25718, 1)
Validare	(7392, 15)	(7392, 1)
Testare	(3696, 15)	(3696, 1)

Tabela 4.1: Împărțire set de date - intrauser

Notația (m, n) reprezintă o structură de m linii și n coloane, iar X exemplele și Y etichetele aferente. Pentru a calcula numărul de linii s-a înmulțit numărul de exemple cu numărul de subiecți 22 și cu numărul de clase 7. Pentru înțelegerea modului de împărțire a datelor intrauser, s-a atașat figura 4.1. În aceasta este ilustrată clasificarea exemplelor unui fișier.

Figura 4.1: Modul de împărțire a datelor intrauser

User

Modul user se bazează pe împărțirea subiecților în cele trei categorii specificate, încercând respectarea procentajelor. Când un anumit subiect aparține unei categorii se înțeleg toate cele 7 fișiere asociate acestuia.

Având un număr de 22 de persoane în baza de date, acestea sunt împărțite astfel în cele 3 categorii: 15 în lotul de antrenare, 4 în lotul de validare, 3 în lotul de testare. Procentajele obținute sunt:

- $\frac{15}{22} \cdot 100 = 68.18\%$ lotul de antrenare
- $\frac{4}{22} \cdot 100 = 18.18\%$ lotul de validare
- $\frac{3}{22} \cdot 100 = 13.63\%$ lotul de testare

Se obțin loturile de date cu structura expusă în tabelul 4.2.

-	X	Y
Antrenare	(25095, 15)	(25095, 1)
Validare	(6692, 15)	(6692, 1)
Testare	(5019, 15)	(5019, 1)

Tabela 4.2: Împărțire set de date - user

Notația (m, n) reprezintă o structură de m linii și n coloane, iar X exemplele și Y etichetele aferente. Pentru a calcula numărul de linii s-a înmulțit numărul de subiecți cu numărul de exemple cuprinse într-un fișier 239 și cu numărul de clase 7. Pentru înțelegerea modului de selecție a persoanelor din cele 3 categorii, s-a atașat figura 4.2.

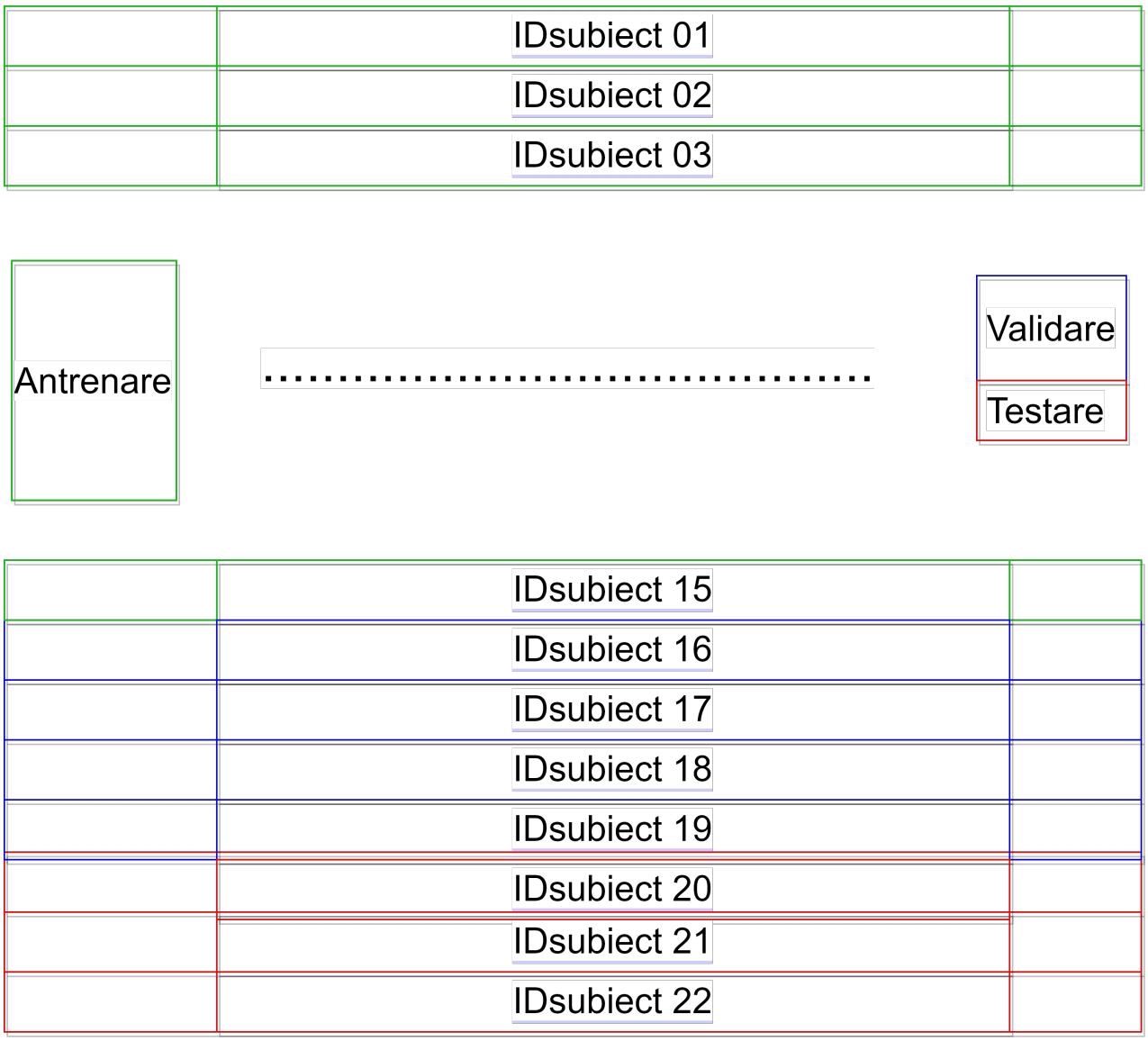

Figura 4.2: Modul de împărțire a datelor user

4.2 Optimizarea rețelei neurale

În continuare se va explica procesul prin care s-a obținut modelul cu rezultatele cele mai bune, acuratețe mare și funcție cost minimizată. Atât pentru modul *user* de împărțire a datelor, cât și pentru cel *intrauser*, s-a procedat similar. Procesul parcurs are la bază noțiuni din referința [24].

O noțiune importantă este cea de *hiperparametru*. Un hiperparametru este o setare sau un parametru care trebuie specificat înainte de antrenarea rețelei. Acesta nu este modificat în timpul antrenării, fiind același pe toată perioada învățării. În următoarea listă sunt enumerați hiperparametrii care au fost modificați pentru a obține rezultate optime. [25]

- Numărul de straturi intermediare
- Numărul de neuroni de pe fiecare strat intermediar
- Funcția de activare folosită de neuronii de pe straturile intermediare

Numărul de straturi intermediare

Spre deosebire de stratul de intrare și de stratul de ieșire cărora le cunoaștem hiperparametrii, straturile intermediare pot fi configurate avantajos. Primul hiperparametru care se alege este numărul acestora.

S-au încercat arhitecturi ale perceptronului multinivel cu 2, 3, 4 sau 5 straturi intermediare.

Numărul de neuroni de pe fiecare strat intermediar

În urma alegerii unui număr de straturi intermediare, pentru fiecare dintre acestea este necesar să decidem numărul de neuroni. Deși nu există o regulă strictă care să menționeze acest lucru, numărul posibil de neuroni prezenți pe un strat intermediar s-a dorit a fi putere a lui 2. Unele dintre avantajele alegerii unui astfel de număr sunt optimizările hardware și software pentru utilizarea adecvată a resurselor sau eficientizarea algoritmului de împărțire în loturi, batching.

Valorile dintre care se poate alege numărul de neuroni prezenți pe fiecare strat sunt 8, 16, 32 și 64.

Funcția de activare a straturilor intermediare

Un ultim hiperparametru pentru a finaliza structura modelului este funcția de activare folosită de toți neuronii de pe straturile intermediare. [26]

Despre modul în care se comportă funcțiile de activare folosite pentru proiectarea modelelor s-a discutat pe larg în secțiunea 1.9.3. Cele 5 utilizate în procesul de optimizare sunt: liniară, ReLU, sigmoid, tangenta hiperbolică și SELU.

În urma fixării mulțimilor pentru hiperparametri s-au obținut 6800 de modele posibile pentru perceptronul multinivel.

$$N_{modele} = (4^2 + 4^3 + 4^4 + 4^5) \cdot 5 = 6800 \tag{4.1}$$

Deoarece resursele hardware necesare antrenării tuturor modelelor pe un număr mare de epoci le depășeau pe cele disponibile, s-a procedat după cum urmează:

1. Toate modelele sunt antrenate o singură epocă. Pentru fiecare se calculează acuratețea pe cele 3 loturi: antrenare, validare și testare.
2. Sunt selectate 5 modele, criteriul folosit fiind acuratețea cea mai bună obținută pe setul de validare.
3. Modelele selectate sunt antrenate 100 de epoci. Cel care dă acuratețea cea mai mare pe lotul de testare este considerat cel mai bun perceptron multinivel.

4.3 Rezultate experimentale

S-au obținut două seturi de rezultate deoarece am avut două moduri de împărțire a datelor, singura diferență făcând-o loturile de date încărcate pentru antrenare, validare și

testare.

4.3.1 Intrauser

Cel mai bun model pentru împărțirea datelor în modul intrauser este *M247*. Are 4 straturi ascunse și folosește ca funcție de activare *ReLU*. Acuratețea pe lotul de testare în urma antrenării cu 100 de epoci este 89.55%. Vezi tabelul 4.4 și figura 4.3.

-	M944	M991	M255	M240	M247
$ACC_{antrenare}$	0.8606	0.8696	0.8657	0.8620	0.8618
$ACC_{validare}$	0.8509	0.8502	0.8490	0.8479	0.8479
$ACC_{testare}$	0.8382	0.8501	0.8414	0.8406	0.8387

Tabela 4.3: Antrenare o singură epocă - intrauser

-	M944	M991	M255	M240	M247
$ACC_{antrenare}$	0.9662	0.9755	0.9750	0.9570	0.9738
$ACC_{validare}$	0.8984	0.8993	0.9030	0.8992	0.8984
$ACC_{testare}$	0.8915	0.8933	0.8947	0.8855	0.8955

Tabela 4.4: Antrenare 100 de epoci - intrauser

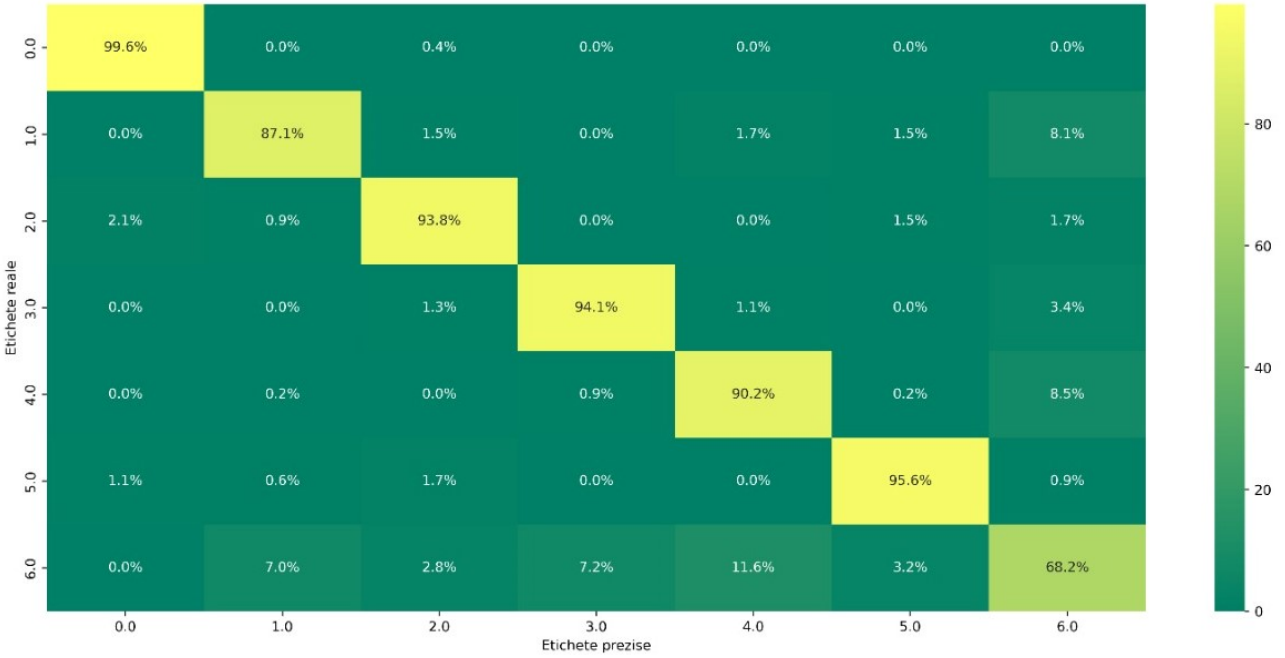

Figura 4.3: Matricea de confuzie a celui mai bun model - intrauser

- M944 - 5 straturi ascunse cu [64, 32, 32, 64, 64] neuroni - funcția de activare SELU

- M991 - 5 straturi ascunse cu [64, 64, 16, 64, 32] neuroni - funcția de activare ReLU
- M255 - 4 straturi ascunse cu [64, 64, 64, 32] neuroni - funcția de activare SELU
- M240 - 4 straturi ascunse cu [64, 32, 64, 64] neuroni - funcția de activare SELU
- **M247** - 4 straturi ascunse cu [64, 64, 16, 32] neuroni - funcția de activare ReLU

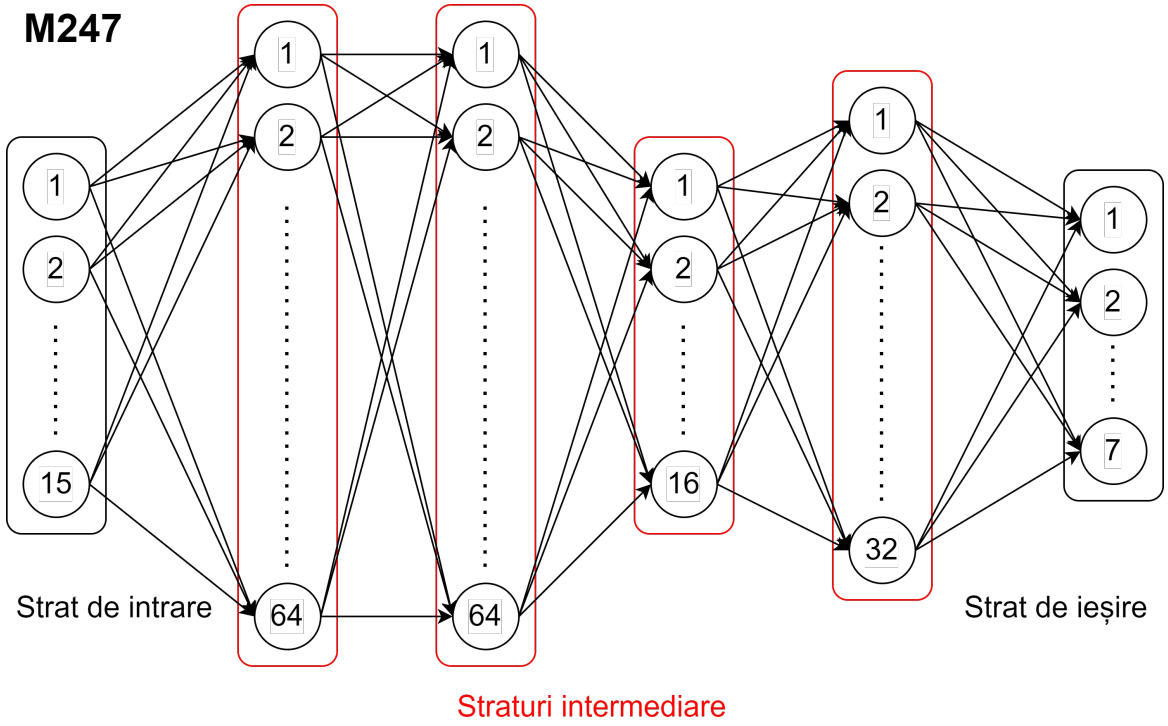

Figura 4.4: Arhitectura celui mai bun model - intrauser

4.3.2 User

Cel mai bun model pentru împărțirea datelor în modul user este *M956*. Are 5 straturi ascunse și folosește ca funcție de activare *ReLU*. Acuratețea (ACC) pe lotul de testare în urma antrenării cu 100 de epoci este 79.45%. Vezi tabelul 4.6 și figura 4.5.

-	M237	M1015	M999	M956	M945
$ACC_{antrenare}$	0.8496	0.8617	0.8618	0.8655	0.8437
$ACC_{validare}$	0.8557	0.8547	0.8543	0.8531	0.8528
$ACC_{testare}$	0.7664	0.7505	0.7704	0.7433	0.7401

Tabela 4.5: Antrenare o singură epocă - user

-	M237	M1015	M999	M956	M945
$ACC_{antrenare}$	0.9605	0.9640	0.9552	0.9658	0.9562
$ACC_{validare}$	0.8511	0.8445	0.8575	0.8598	0.8475
$ACC_{testare}$	0.7603	0.7573	0.7742	0.7945	0.7505

Tabela 4.6: Antrenare 100 de epoci - user

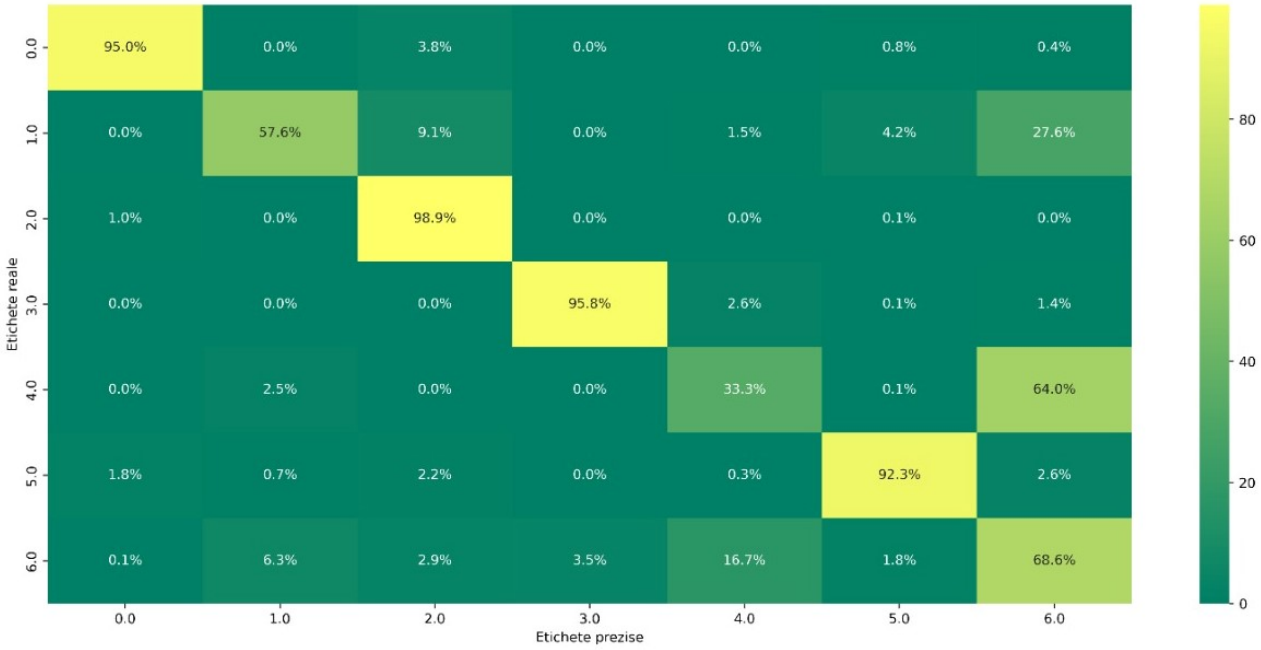

Figura 4.5: Matricea de confuzie a celui mai bun model - user

- M237 - 4 straturi ascunse cu [64, 32, 64, 8] neuroni - funcția de activare ReLU
- M1015 - 5 straturi ascunse cu [64, 64, 64, 16, 32] neuroni - funcția de activare SELU
- M999 - 5 straturi ascunse cu [64, 64, 32, 16, 32] neuroni - funcția de activare SELU
- **M956** - 5 straturi ascunse cu [64, 32, 64, 32, 64] neuroni - funcția de activare ReLU
- M945 - 5 straturi ascunse cu [64, 32, 64, 8, 8] neuroni - funcția de activare SELU

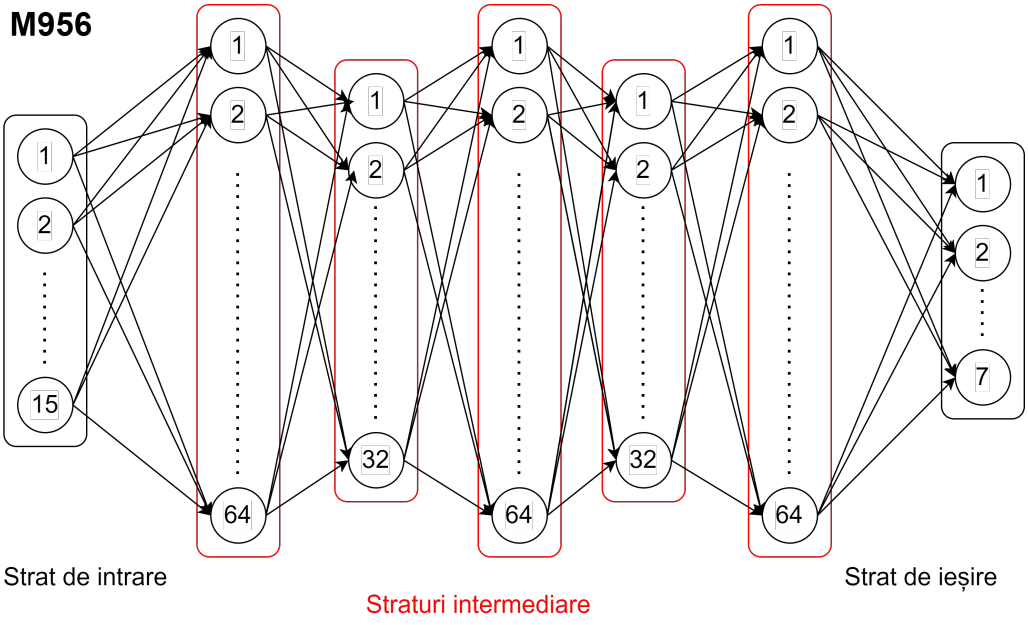

Figura 4.6: Arhitectura celui mai bun model - user

4.4 Interpretarea rezultatelor

În final sunt aduse câteva observații cu privire la rezultatele obținute.

Un aspect important care nu trebuie pierdut din vedere este faptul că nu toate arhitecturile rezultate sunt antrenate 100 de epoci pentru a vedea performanța fiecăreia, acesta fiind scenariul ideal. Se oservă cum clasamentul celor mai bune 5 modele se modifică în urma efectuării antrenării efectuate cu 100 de epoci.

În ambele moduri de împărțire a datelor, cel mai bun model este inițial pe locul 4 în clasament, iar după antrenarea efectuată cu 100 de epoci, acesta ajunge cel mai performant. Această observație crește plauzibilitatea ipotezei în care clasamentul ar fi diferit dacă toate cele 6800 de modele ar fi fost antrenate 100 de epoci. Atât în tabelul 4.7, cât și în tabelul 4.8, criteriul folosit este acuratețea pe lotul de testare.

De asemenea, se remarcă faptul că cele 5 modele selectate în urma antrenării cu o singură epocă, folosesc toate fie funcția de activare ReLu, fie funcția de activare SELU. Sunt singurele care au codomeniul infinit superior, nelimitând rezultatul obținut. Spre deosebire de tangenta hiperbolică sau sigmoid, funcții ce au ieșirea redusă la un interval finit, ReLU și SELU pot captura mai fin anumite caracteristici. Acest aspect este prezent atât la împărțirea datelor în modul intrauser, cât și la împărțirea datelor în modul user.

Făcând o comparație între acuratețea celui mai bun model intrauser și accuratețea celui mai bun model user, se confirmă aspectul intuit în care atingerea unor performanțe mai ridicate are loc atunci când rețelei neuronale i se pun la dispoziție date ce includ toți subiecții din baza de date. Accuratețea celui mai bun model proiectat pentru situația în care subiecții sunt împărțiți în cele 3 categorii (79.45%) este cu aproximativ 10% mai mică față de modul intrauser de împărțire a datelor (89.55%).

Se pot face numeroase observații pe baza matricelor de confuzie, figura 4.3 și 4.5, a celor două modele. Privind în ansamblu, atât în modul intrauser, cât și în modul user de împărțire a datelor, cele mai mari valori se regăsesc pe diagonala principală a matricelor, ceea ce denotă un procentaj de clasificare corectă ridicat. De remarcat este și accuratețea scăzută cu care au fost clasificate exemplele din clasa 6 (mișcare în mod aleator). Exemplele având informația a unei secvețe de 0.5 secunde, multe dintre cele care se regăseau în clasa 6 au părut similare cu cele din clasa 4 (alergare pe loc). O soluție poate fi mărirea ferestrelor. Clasele cel mai bine clasificate au fost cele care au o periodicitate bine definită, clasa 0 (stat), clasa 2 (genuflexiuni) și clasa 3 (jumping jacks).

Totodată, creșterea performanțelor rețelelor implementate poate fi posibilă dacă numărul de date ar fi mărit. În lucrarea de față s-a luat decizia de a selecta un singur minut din toate fișierele pentru a avea un număr echilibrat de exemple pentru fiecare clasă. Cum am menționat anterior, pentru clasa 2 (genuflexiuni) a fost necesar un efort fizic pe care unii dintre subiecți nu l-au putut face.

-	Antrenare o singură epocă	Antrenare 100 de epoci
1	M991	**M247**
2	M255	M255
3	M240	M991
4	**M247**	M944
5	M944	M240

Tabela 4.7: Clasament intrauser

-	Antrenare o singură epocă	Antrenare 100 de epoci
1	M999	**M956**
2	M237	M999
3	M1015	M237
4	**M956**	M1015
5	M945	M945

Tabela 4.8: Clasament user

5 Concluzii

În timpul realizării acestui proiect au fost întâlnite aspecte care nu au fost preîntâmpinate în stadiul incipient. Acestea au devenit provocări care au solicitat soluții creative. Câteva lucruri care puteau fi realizate altfel, dar și aspecte încă incerte, se regăsesc în lista următoare:

- S-ar fi vrut realizarea sistemului de achiziție cu ajutorul unui senzor de mișcare care să conțină și un giroscop? Am fi preferat informație legată de înclinația unghiulară la schimbul frecvenței de eșantionare ridicate?
- Ce se dorește? O reproducere detaliată a mișcărilor sau un volum de date acceptabil ținând cont de limitările hardware?
- Clasele alese pentru baza de date sunt reprezentative pentru motricitatea membrelor inferioare? Cum s-ar fi comportat alte clase de mișcări?
- În ce măsură persoanele care formează baza de date au realizat mișcările într-un mod reprezentativ pentru a putea face o generalizare?
- Ar fi fost indicat să fie incluse în baza de date doar persoane ce pot rezista la un volum de efort ridicat sau alegerea unor mișcări mai puțin solicitante?
- Am fi dorit ca preprocesarea datelor să includă și o etapă de mediere? Ar fi crescut performanța atinsă de rețeaua neuronală?
- Dacă realizam extragerea mai multor caracteristici se obțineau rezultate mai bune?
- Ce valori pentru hiperparametrii rețelei neuronale am fi putut să mai adăugăm? O altă funcție de activare sau mai multe straturi ascunse de neuroni ar fi atins performanțe mai ridicate?
- Cât de important era ca antrenarea pe un număr mare de epoci să se facă pentru toate arhitecturile?
- Cum decidem dacă rezultatele obținute sunt bune sau nu? Cu ce le comparăm ținând cont că baza de date nu a mai fost folosită în alte analize?

Contribuții personale

S-a proiectat și implementat un sistem de achiziție de date care poate monitoriza mișcarea și îndeplinește anumite criterii enumerate în secțiunea 2. Dificultățile întâlnite au constat în limitările hardware ale componentelor, făcându-se deseori compromisuri pentru găsirea soluției optime.

S-a creat o bază de date ce poate sta la baza unor viitoare proiecte. Chiar dacă subiecții nu pot fi reprezentativi deoarece se află într-un interval de vârstă restrâns și numărul lor este scăzut, mișcările din care este formată baza de date sunt un început bun pentru analiza unor aspecte ce țin de motricitatea membrelor inferioare.

Rețeau neurală care a folosit datele pentru învățare a trecut printr-un proces de

optimizare corect, care a fost cea mai bună soluție ținând cont de resursele limitate. Performanța atinsă de aceasta este mulțumitoare.

Dezvoltări ulterioare

Proiectul prezentat în această lucrare poate fi dezvoltat pe viitor atât din punct de vedere hardware, cât și din punct de vedere software. Câteva opțiuni posibile sunt:

- Folosirea mai multor senzori de mișcare ce pot aduce date suplimentare, cum ar fi un giroscop sau un magnetometru
- Creșterea frecvenței de eșantionare
- Mărirea bazei de date prin includerea mai multor clase sau prin adăugarea mai multor subiecți de vârste diferite
- Etichetarea exemplelor într-un mod mai complex, cum ar fi includerea genului subiectului
- Extragerea mai multor trăsături
- Încercarea unor modele de învățare nesupervizate

Referințe

[1] *Journal of Orthopaedic Sports Physical Therapy* **1998**, *28*, 400–404.

[2] R. M. Udrea, Prelucrarea digitală a semnalelor - suport de curs, **2022**.

[3] A. A. Neacsu, G. Cioroiu, A. Radoi, C. Burileanu în 2019 42nd International Conference on Telecommunications and Signal Processing (TSP), **2019**, pp. 232–235.

[4] V. E. V. NEAGOE, Recunoașterea formelor și inteligență artificială - suport de curs, **2022**.

[5] R. Zunino, P. Gastaldo în 2002 IEEE International Symposium on Circuits and Systems. Proceedings (Cat. No.02CH37353), *vol. 2*, **2002**, pp. II–II.

[6] C.-H. Chen, P.-H. Lin, J.-G. Hsieh, S.-L. Cheng, J.-H. Jeng în 2020 3rd IEEE International Conference on Knowledge Innovation and Invention (ICKII), **2020**, pp. 200–203.

[7] C.-C. Yu, B.-D. Liu în Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290), *vol. 2*, **2002**, 1218–1223 vol.2.

[8] S. Stalin, T. Sreenivas în IEEE International Conference on Neural Networks, **1993**, 1427–1432 vol.3.

[9] A. Mustapha, L. Mohamed, K. Ali, *Journal of Physics: Conference Series* **2021**, *1743*, 012002.

[10] A. Lodwich, Y. Rangoni, T. Breuel în 2009 International Joint Conference on Neural Networks, **2009**, pp. 1877–1884.

[11] C. VERTAN, Analiza imaginilor - suport de curs, **2023**.

[12] M. Kaloev, G. Krastev în 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), **2021**, pp. 1–5.

[13] K.-L. Chau, S. Lewis, Y. Zhao, R. Howe, S. Bart, R. Marcheselli, *Sensors and Actuators A: Physical* **1996**, *54*, 472–476.

[14] Data Sheet ADXL345, Analog Devices, **2022**.

[15] V. DRAGOMIR, Circuite integrate digitale - suport de curs, **2021**.

[16] Inter-Integrated Circuit, Philips Semiconductor, **1982**.

[17] ESP32 Series Data Sheet, Espressif Systems, **2023**.

[18] M. Shinagawa, Y. Akazawa, T. Wakimoto, *IEEE Journal of Solid-State Circuits* **1990**, *25*, 220–224.

[19] A. Kamerman, N. Erkocevic în Proceedings of 8th International Symposium on Personal, Indoor and Mobile Radio Communications - PIMRC '97, *vol. 3*, **1997**, 1221–1227 vol.3.

[20] K. H. Kwon, C. B. Shin, T. H. Kang, C.-S. Kim, *Journal of Power Sources* **2006**, *163*, Special issue including selected papers presented at the Second International Conference on Polymer Batteries and Fuel Cells together with regular papers, 151–157.

[21] TP4056 1A Standalone Linear Li-Ion Battery Charger with Thermal Regulation in SOP-8, NanJing Top Power ASIC Corp., **2010**.

[22] M. CIUC, Decizie și estimare în prelucrarea informației - suport de curs, **2022**.

[23] Y. Sutcu, H. T. Sencar, N. Memon în 2010 20th International Conference on Pattern Recognition, **2010**, pp. 1469–1472.

[24] O. GRIGORE, Tehnici de optimizare - suport de curs, **2022**.

[25] D.-Y. Yeung în [Proceedings] 1991 IEEE International Joint Conference on Neural Networks, **1991**, 158–164 vol.1.

[26] K. A. A. Kamarulzaini, N. Ismail, M. H. F. Rahiman, M. N. Taib, N. A. M. Ali, S. N. Tajuddin în 2018 IEEE 14th International Colloquium on Signal Processing Its Applications (CSPA), **2018**, pp. 250–254.

Anexa 1 - Codul sursă al aplicației

https://github.com/catalinaaaaaa/Licenta