

Predicția unor diagnostice medicale folosind modele bazate pe rețele neurale

Radu Breazu

27 mai 2024

În acest document sunt prezentate rezultatele obținute în urma antrenării și a testării unor modele bazate pe rețele neurale pe două seturi de date ce conțin informații medicale despre pacienți, în vederea prezicerii diagnosticului unui pacient pe baza informațiilor furnizate.

1 Seturile de date

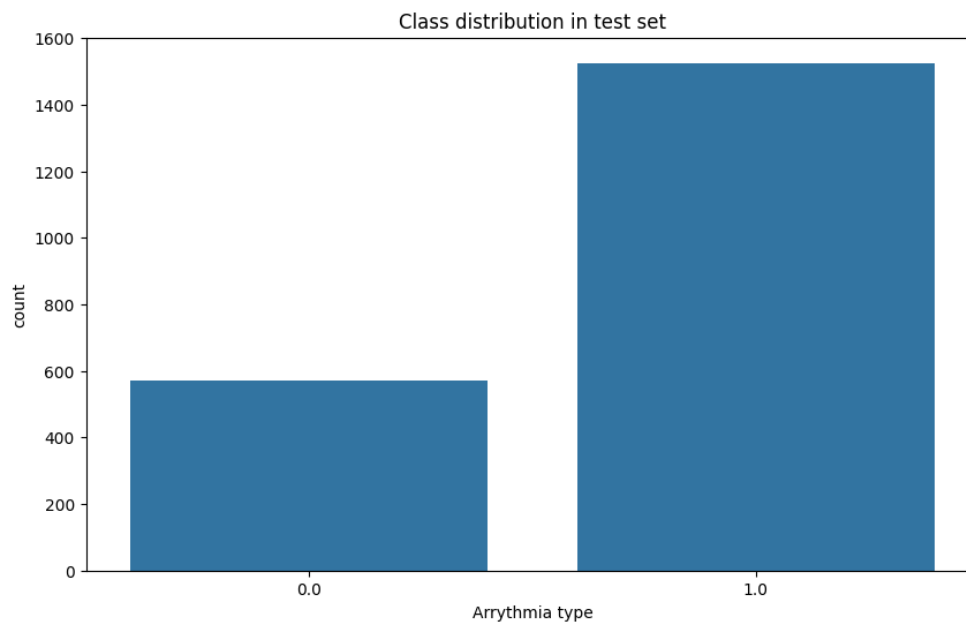
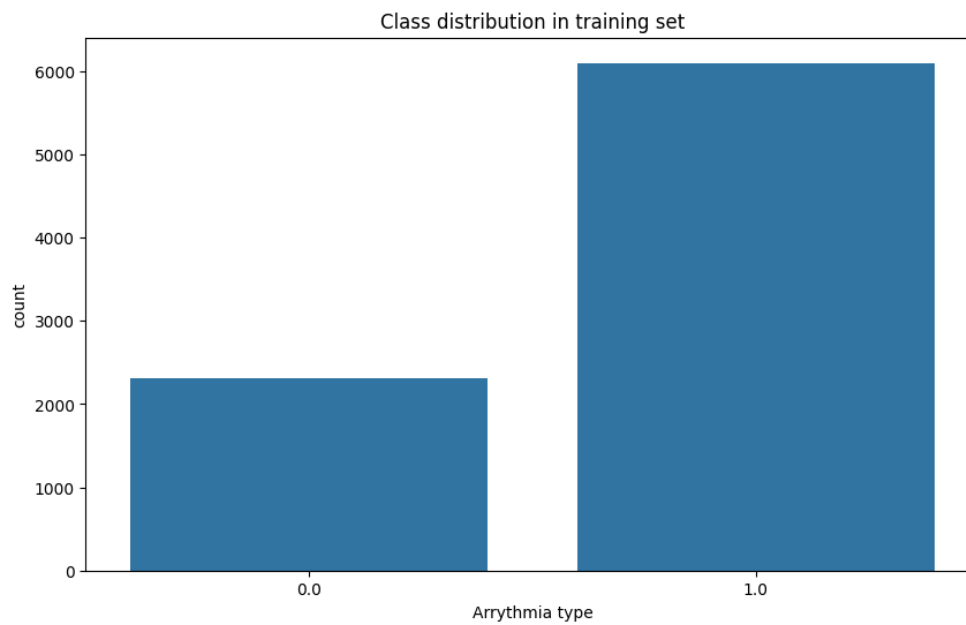
În acest experiment, am folosit două seturi de date: **Patients**, pe care l-am analizat în experimentul anterior (care trata diagnosticarea automată a unor pacienți folosind metode de învățare automată bazate pe clasificatori) și **PTB Diagnostic ECG**, care conține date secvențiale provenind de la 14552 de pacienți și reprezentând bătăi ale inimii, împărțite în două clase, normal și anormal, după cum pacientul de la care s-au colectat respectivele date suferă sau nu de aritmie cardiacă. Întrucât primul set de date a fost explorat anterior, în acest experiment numai setul PTB Diagnostic ECG a fost analizat în detaliu. În continuare sunt prezentate rezultatele acestei analize.

1.1 Considerații inițiale

În cazul ambelor seturi de date, am folosit împărțirea în set de antrenare (en. *train*) și set de testare (en. *test*), iar procentele au fost identice: 80% date de antrenare și 20% date de testare. Consider că această pondere (20%) este suficient de mică încât să nu afecteze capacitatea de învățare a modelelor folosite (seturile de date folosite, în special "Patients", conțin un număr relativ mic de exemple, lucru care ar putea influența negativ puterea predictivă a modelelor, dacă procentul de date de testare este prea mare), dar, totodată, suficient de mare încât să permită o evaluare corectă a rețelelor utilizate. Împărțirea în cele două seturi a fost făcută folosind metoda `train_test_split` din biblioteca `scikit-learn` setând corespunzător argumentul `stratify`, pentru a garanta păstrarea distribuției claselor din setul inițial în mulțimile rezultate.

1.2 Analiza echilibrului de clase

Am afișat câte o histogramă care arată distribuția claselor în fiecare dintre mulțimile specificate. Având în vedere paragraful anterior, în mod evident, am obținut aceeași distribuție a pacienților cu, respectiv fără, aritmie în mulțimile de antrenare și de test. Rezultatul este prezentat mai jos:

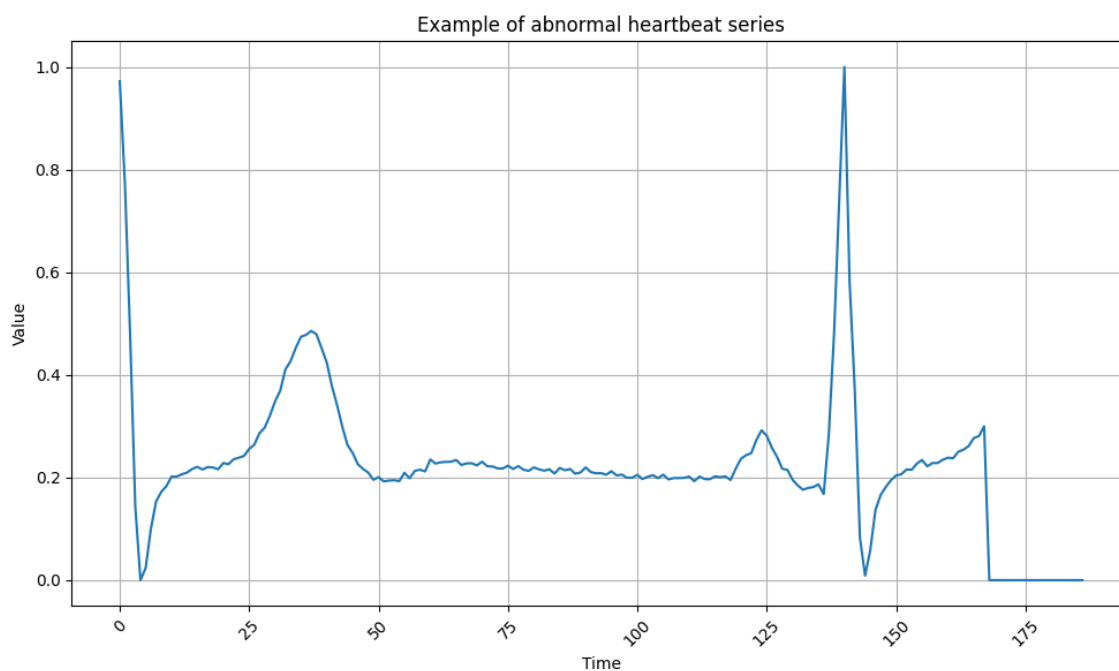
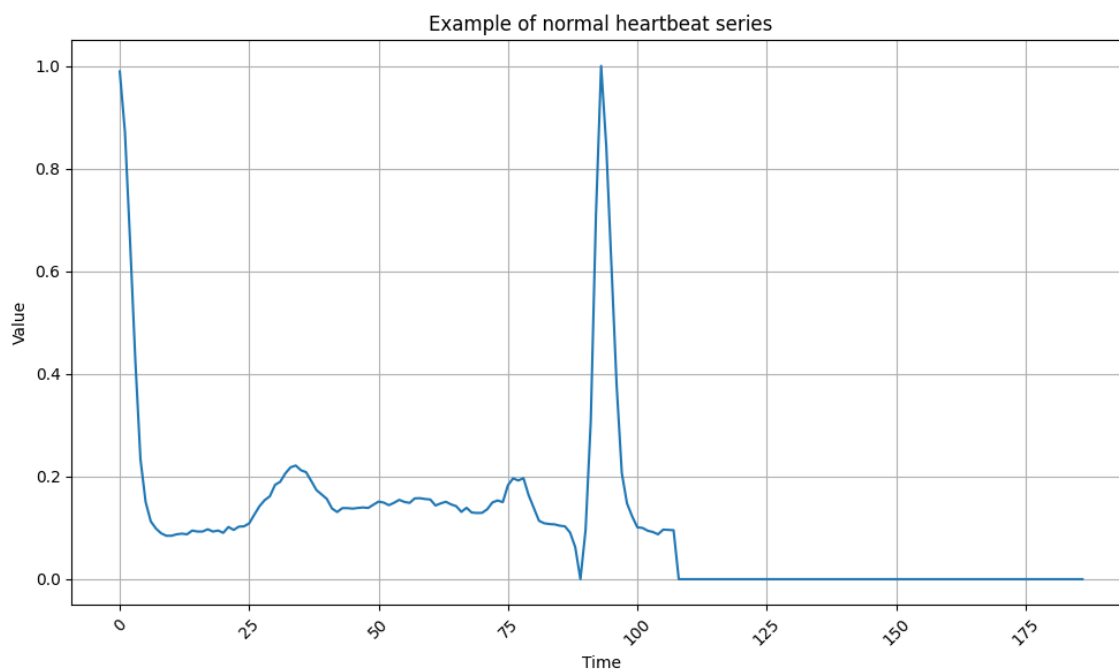


Se observă că numărul de oameni sănătoși (i.e. care nu prezintă aritmie) este de aproximativ 2.5 ori mai mic decât al celor cu aritmie în setul de antrenare (cu aproximație, sunt 2250 de oameni sănătoși și 6000 de pacienți). Raportul rămâne aproape neschimbat în setul de test, în care există, cu aproximație, 600 de oameni sănătoși și 1550 de pacienți cu aritmie.

1.3 Vizualizarea seriilor de timp

1.3.1 Exemple de serii de timp

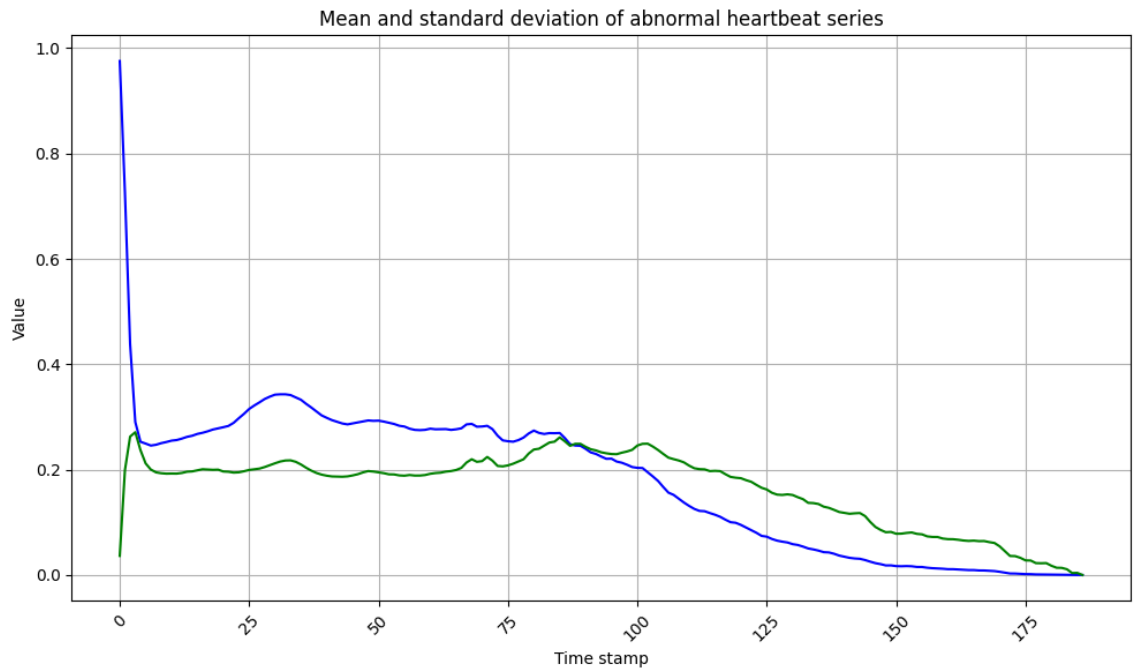
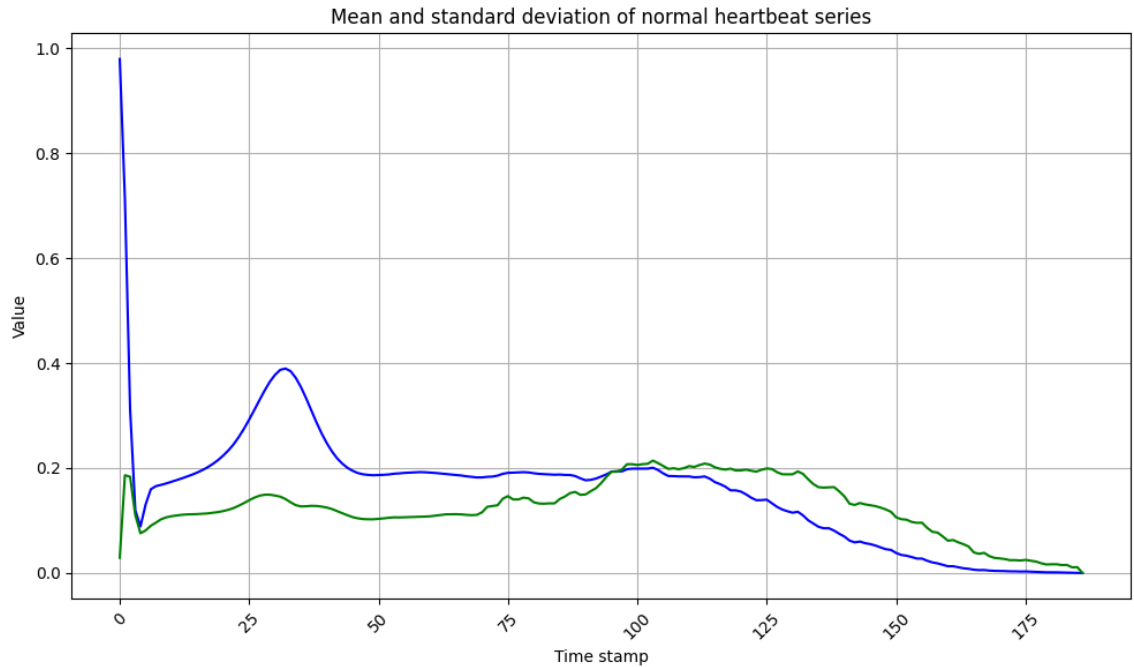
Am afișat câte un grafic cu o serie de timp (aleasă la întâmplare) din setul de date “PTB Diagnostic ECG” din fiecare categorie: non-pacient, respectiv pacient. Rezultatele sunt prezentate mai jos:



Se observă tiparul total diferit al activității cardiace între un individ care nu suferă de aritmie și unul care are această afecțiune.

1.3.2 Statistici referitoare la seriile de timp

Am realizat grafice (câte unul per clasă) care ilustrează media și abaterea standard ale valorilor semnalelor electrice ale inimii, pentru fiecare moment al electrocardiografei. Pentru calculul statisticilor am considerat toate exemplele (atât pe cele din setul de antrenare, cât și pe cele din setul de testare). Rezultatele sunt următoarele:



2 Modelele de rețele neurale

În acest experiment, am testat și am evaluat două arhitecturi de rețele neurale: una de tip **MLP** (multi-layered perceptron) (i.e. mai multe straturi conectate “în cascadă” de perceptroni) și una de tip **convoluțional**. Caracteristicile arhitecturilor și rezultatele obținute sunt prezentate în cele ce urmează.

2.1 Arhitectura de tip MLP

Pentru această arhitectură, am folosit 5 straturi liniare (i.e. de tipul `nn.Linear`), astfel:

- a) primul strat primește la intrare toate caracteristicile disponibile pentru o înregistrare din setul de date considerat (caracteristici care sunt în număr de `input_features` – acest număr este furnizat ca parametru constructorului modelului) și scoate la ieșire un număr de 64 de caracteristici
- b) al doilea strat ia la intrare 64 de caracteristici (cele furnizate de ieșirea primului strat), și are drept ieșire 32 de caracteristici
- c) al treilea strat are drept intrare 32 de caracteristici și furnizează la ieșire 16 caracteristici
- d) al patrulea strat are drept intrare 16 caracteristici și furnizează la ieșire 8 caracteristici
- e) ultimul strat primește la intrare 8 caracteristici și scoate la ieșire atâtea “caracteristici” câte clase există în setul de date

Ieșirea ultimului strat este trecută printr-o funcție de activare înainte de a fi prezentată ca predicție a rețelei. În urma încercărilor, am constatat că funcția de activare cea mai bună pentru setul de date “Patients” este *LeakyReLU()*, iar pentru “PTB Diagnostic ECG” este *tanh()*.

Optimizatorul este un alt element ce diferențiază modelele folosite pentru cele două seturi de date. Astfel, în urma experimentelor, cel mai bun optimizator pentru “Patients” este **Adam** cu rata de învățare $lr = 3e-4$ și `weight_decay = 1e-4`, în timp ce pentru “PTB Diagnostic ECG” este **SGD** cu rata de învățare $lr = 5e-4$, `momentum = 0.9` și fără `weight_decay`. Pentru primul set de date, folosirea `weight_decay` a avut un efect minim asupra acurateții rețelei neurale, pe când pentru al doilea set, folosirea unui `weight_decay` chiar și de 10^{-4} duce la scăderea acurateții rețelei. Acest lucru înseamnă că modelul nu beneficiază de pe urma regularizării pe setul al doilea de date, deci că nu există overfitting care să necesite corectare, drept care am ales să nu folosesc regularizare pentru “PTB Diagnostic ECG”.

Încă un element diferit între modelele aplicate pe seturile de date este numărul de epoci de antrenare folosite. Pentru “Patients”, cele mai bune rezultate au fost obținute antrenând rețeaua neurală timp de 200 de epoci, spre deosebire de “PTB Diagnostic ECG”, în cazul căruia rezultatele optime sunt atinse după doar 10 epoci de antrenare.

Având în vedere atât parametrii, cât și hiperparametrii menționați mai sus, rezultatele obținute sunt prezentate (în procente¹) în tabelele de mai jos:

¹Toate numerele prezentate în tabele sunt rotunjite prin trunchiere la 2 zecimale exacte.

Clasă	Acuratețe	Precizie	Recall	Scor F1
0	79.48	76.36	85.71	80.77
1	79.48	69.23	67.92	68.57
2	79.48	68.63	67.31	67.96
3	79.48	84.85	51.85	64.37
4	79.48	71.25	89.06	79.17
5	79.48	92.45	90.74	91.59
6	79.48	96.72	100.00	98.33
Medie	79.48	79.92	78.94	78.68
Abatere standard	0.00	10.64	15.68	11.82

Tabelul 1: Performanțele rețelei MLP pe setul de date “Patients”

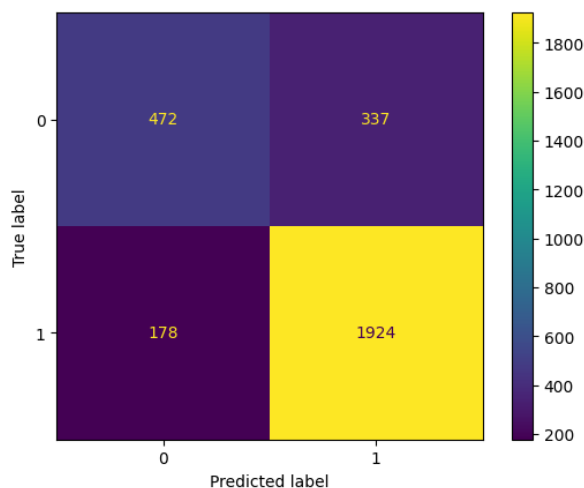
Clasă	Acuratețe	Precizie	Recall	Scor F1
0	82.31	72.62	58.34	64.70
1	82.31	85.10	91.53	88.20
Medie	82.31	78.86	74.93	76.45
Abatere standard	0.00	6.24	16.59	11.75

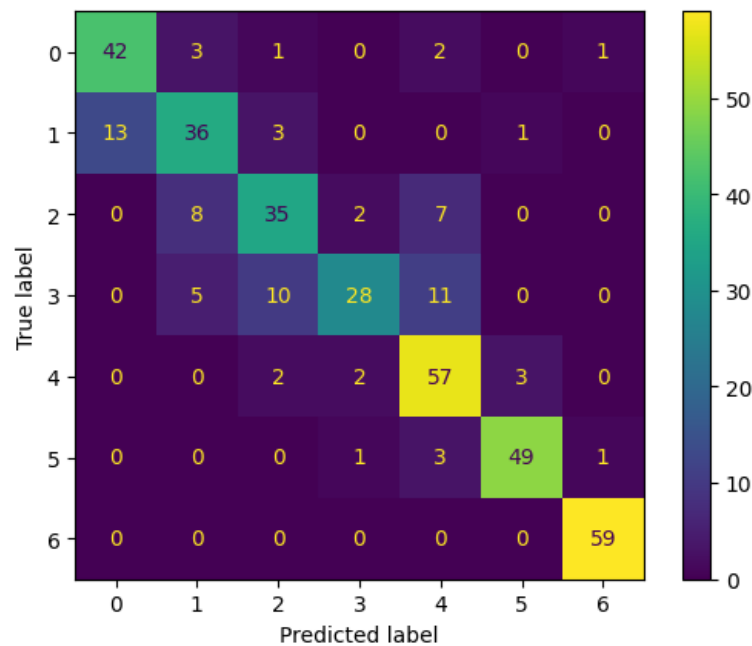
Tabelul 2: Performanțele rețelei MLP pe setul de date “PTB Diagnostic ECG”

După cum se observă, pe setul de date “Patients”, rețeaua de tip MLP prezice cel mai bine diagnosticul D6, înregistrând pentru clasa corespunzătoare atât cea mai mare precizie (96.72%), cât și cel mai mare recall (100%) (implicit și cel mai mare scor F1). Cu toate acestea, performanțele modelului pe acest set de date sunt mai slabe decât cele înregistrate de clasificatorii folosiți în experimentul anterior (a se vedea rezultatele prezentate în documentul respectiv).

După cum reiese din Tabelul 2, procentele rămân similare și în cazul setului de date “PTB Diagnostic ECG”. Diagnosticul “pacient” este mai bine prezis decât cel de “non-pacient”, cu un scor F1 de 88.20%, față de 64.70%. De remarcat este faptul că, pentru acest diagnostic, atât precizia, cât și recall-ul sunt mai mari.

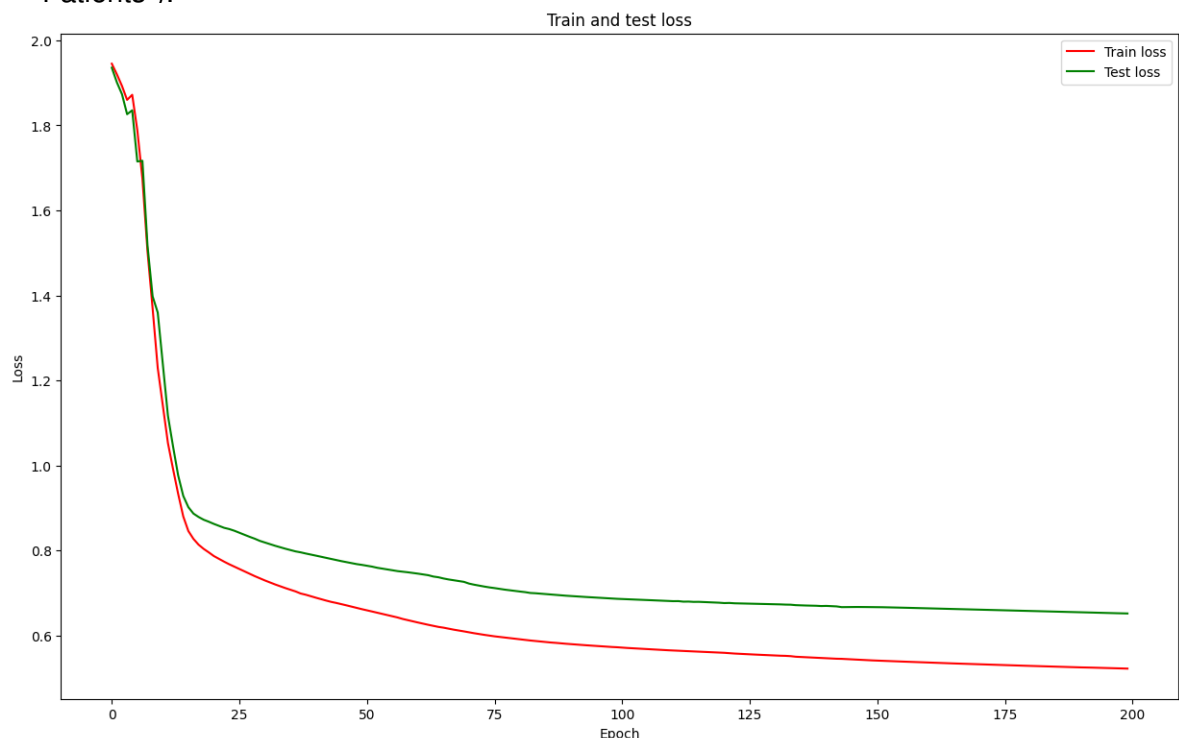
După antrenarea și testarea pe fiecare set de date, am afișat matricea de confuzie corespunzătoare. Graficele astfel obținute sunt prezentate mai jos (prima matrice corespunde setului de date “PTB Diagnostic ECG”):

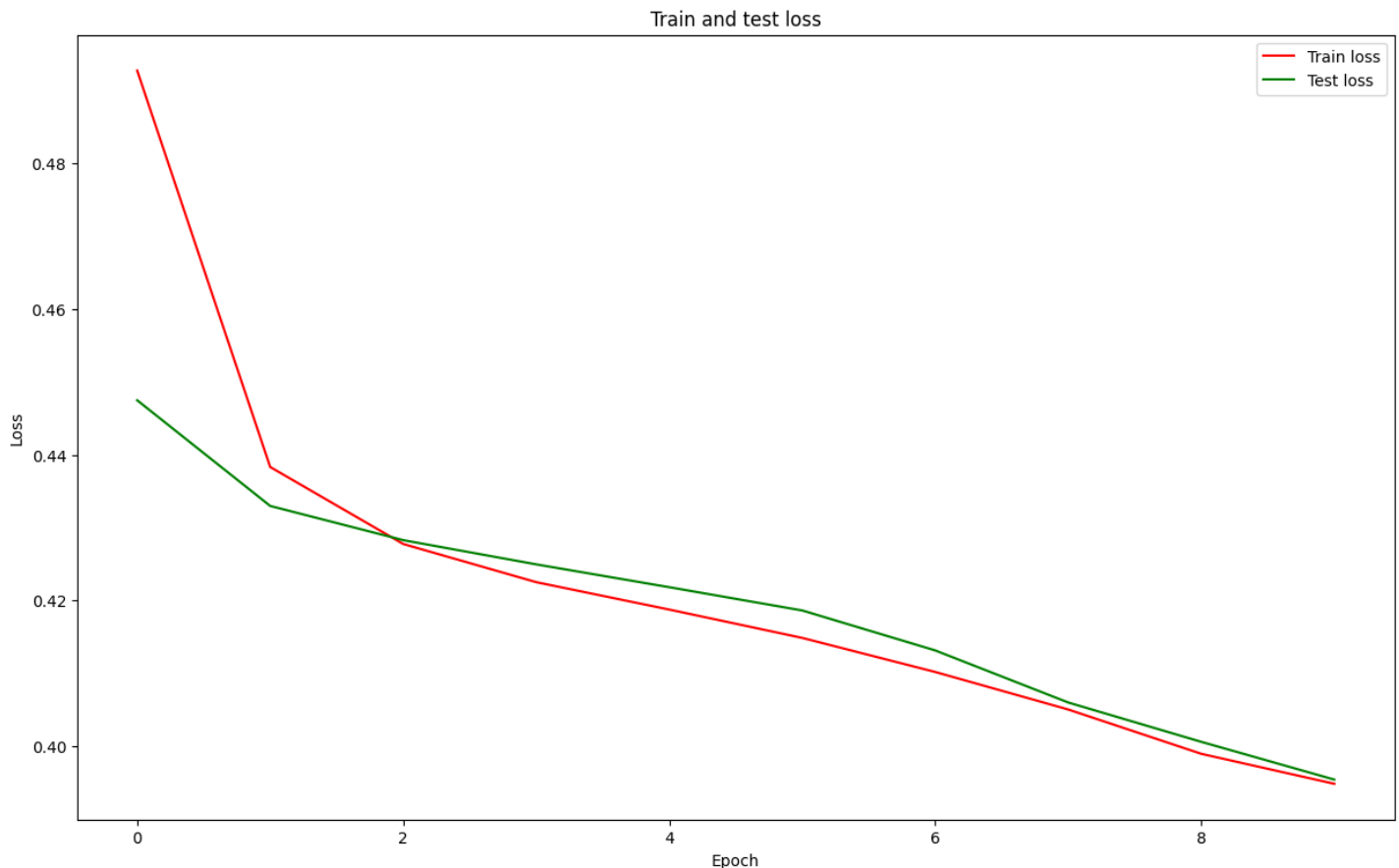




Pentru “Patients”, se observă că modelul are tendința de a raporta mai multe cazuri fals pozitive (326) decât fals negative (192). Dat fiind că aritmia este o boală relativ gravă, consider că raportarea mai multor cazuri fals pozitive este o problemă pe care o are modelul de rețea MLP. De asemenea, pentru “PTB Diagnostic ECG”, matricea de confuzie ilustrează în mod vizibil ratele cele mai mici ale preciziei (care se înregistrează pentru diagnosticul D4 – diagnostic pentru care 11 pacienți sunt clasificați ca suferind, de fapt, de D3, 7 pacienți sunt clasificați ca suferind de D2, 3 pacienți ca suferind de D5, iar 2 pacienți ca suferind de D0) și ale recall-ului (care se înregistrează pentru diagnosticul D3 – diagnostic pentru care 5 pacienți sunt atribuiți greșit diagnosticului D1, 10 pacienți sunt diagnosticați cu D2, iar 11 pacienți sunt diagnosticați cu D4).

Valorile funcției de eroare la antrenare, respectiv la testare pe cele două seturi de date sunt ilustrate în graficele de mai jos (primul grafic corespunde setului de date “Patients”):





Cele două grafice arată fără echivoc că, pentru ambele seturi de date, valorile funcției de eroare scad până la finalul antrenării, ceea ce arată că modelul nu suferă de overfitting.

2.2 Arhitectura de tip convoluțional

Am antrenat (și am testat) arhitectura convoluțională doar pe setul de date “PTB Diagnostic ECG”. Pentru aceasta, am folosit:

- un strat convoluțional care primește la intrare un singur canal (deoarece secvența de bătaii ale inimii este unidimensională) și scoate la ieșire 32 de canale, cu un kernel de dimensiune 3, `stride = 1` și `padding = 1`. Ieșirea acestui strat este trecută prin funcția de activare *ReLU()*, și mai apoi prelucrată de un strat de average pooling cu dimensiunea kernelului de 2 (în cod: `nn.AvgPool1d(kernel_size = 2)`)
- un strat convoluțional care primește la intrare 32 de canale și scoate la ieșire 64 de canale, cu parametri și postprocesare identice cu cele ale primului strat (`kernel_size = 3`, `stride = 1`, `padding = 1`, funcție de activare *ReLU()* și `nn.AvgPool1d(kernel_size = 2)`). În plus față de primul strat, ieșirea rezultată în urma operației de average pooling este trecută printr-un strat de aplatizare (`nn.Flatten()`), pentru a o putea furniza ca intrare următorului strat (care este de tip complet conectat – en. *fully connected*).

- c) un strat complet conectat care primește la intrare un vector de dimensiune $64 \cdot \left\lceil \frac{\text{input_features}}{4} \right\rceil$, unde `input_features` este numărul de caracteristici din vectorul inițial (care, în cazul acestui set de date, este egal cu 187), și scoate la ieșire un vector cu 128 de elemente. Componentele sale sunt mai apoi trecute prin funcția *ReLU()*, și trimise ultimului strat
- d) un strat complet conectat care primește la intrare un vector cu 128 de elemente și scoate la ieșire un vector cu `num_classes` componente, unde `num_classes` este numărul de clase din problema de clasificare (în cazul setului de date furnizat, acest număr este egal cu 2).

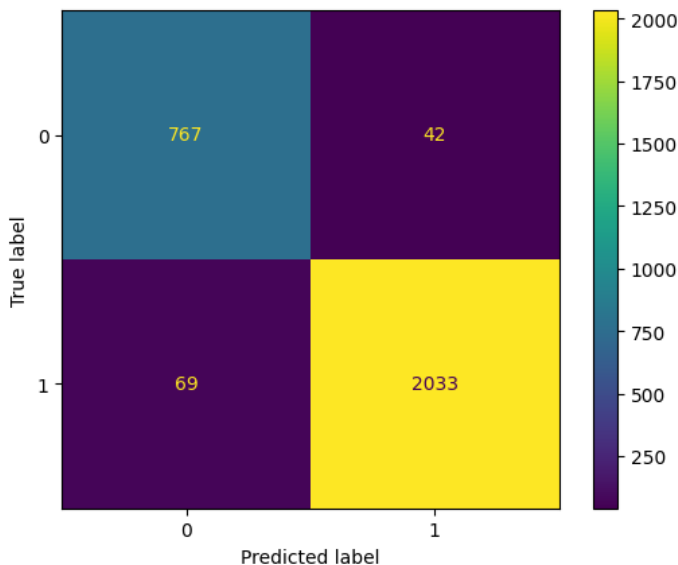
Funcția de eroare folosită a fost `nn.CrossEntropyLoss()`, optimizatorul a fost SGD cu rata de învățare `lr = 5e-4` și `momentum = 0.9`, iar antrenarea a fost făcută timp de 15 epoci. Având în vedere toți parametrii și hiperparametrii prezentați, performanțele obținute sunt următoarele:

Clasă	Acuratețe	Precizie	Recall	Scor F1
0	96.19	91.75	94.81	93.25
1	96.19	97.98	96.72	97.34
Medie	96.19	94.86	95.76	95.29
Abatere standard	0.00	3.11	0.95	2.04

Tabelul 3: Performanțele rețelei convoluționale pe setul de date “PTB Diagnostic ECG”

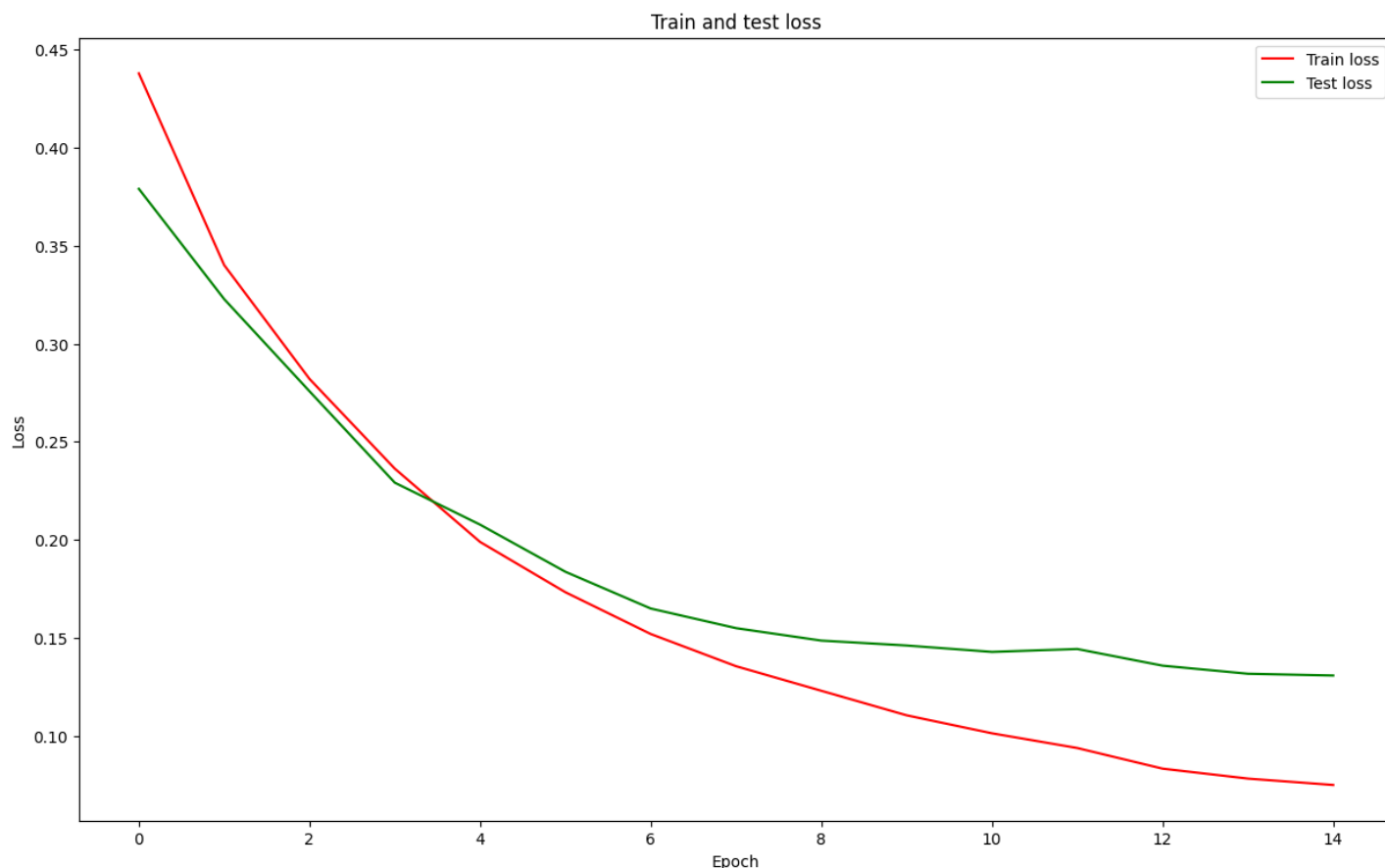
Încă o dată, se observă că diagnosticul “bolnav de aritmie” (i.e. clasa 1) este prezis cel mai bine, rețeaua înregistrând atât cea mai bună precizie, cât și cel mai bun recall pe această clasă. De asemenea, toate metricile considerate au valori de peste 90%, iar acuratețea generală este de peste 96%, ceea ce arată faptul că rețeaua convoluțională are capacitatea de a învăța foarte bine să distingă o persoană sănătoasă de una care suferă de aritmie.

Matricea de confuzie obținută este prezentată mai jos:



Se observă că modelul prezice în mod eronat prezența unei aritmii în 42 de cazuri, mai puține decât cele în care prezice greșit absența ei (69 de cazuri), ceea ce reprezintă un punct forte al acestei rețele, comparativ cu cea de tip MLP. Totodată, comparativ cu modelul MLP, numărul de pacienți încadrați corect în categoria “normal” a crescut simțitor (de la 472 la 767), ceea ce se reflectă în creșterea semnificativă a recall-ului pe clasa 0, de la 58.34% la 94.81%. Și numărul de pacienți depistați corect a crescut cu aproximativ 100 (de la 1924 la 2033), fapt care este vizibil în creșterea preciziei pe clasa 1 de la 85.10% la 97.98%.

Evoluția funcției de eroare pe antrenare și pe testare este redată în graficul de mai jos:



Forma graficului funcției de eroare pe mulțimea de test, comparativ cu cea de pe mulțimea de antrenare, arată faptul că nu există overfitting (pe setul de test, valorile funcției de eroare scad, până se ajunge într-un platou, la valoarea minimă a funcției).