Tema 2 ML

Paunoiu Darius Alexandru – 342C4

Cuprins

1. Introducere

Tema își propune analiza unui set de date care are ca ținta unul din 7 diagnosticuri posibile, analizând si performanța a patru algoritmi pe acest set de date. Setul de date a primit mici modificări înainte de orice, pentru consistenta datelor (de exemplu virgulele au fost înlocuite cu punct pentru a trata zecimalele). Timpul de execuție nu este unul ridicat. De asemenea, se caută si împărțirea in clase a unui set de aritmii.

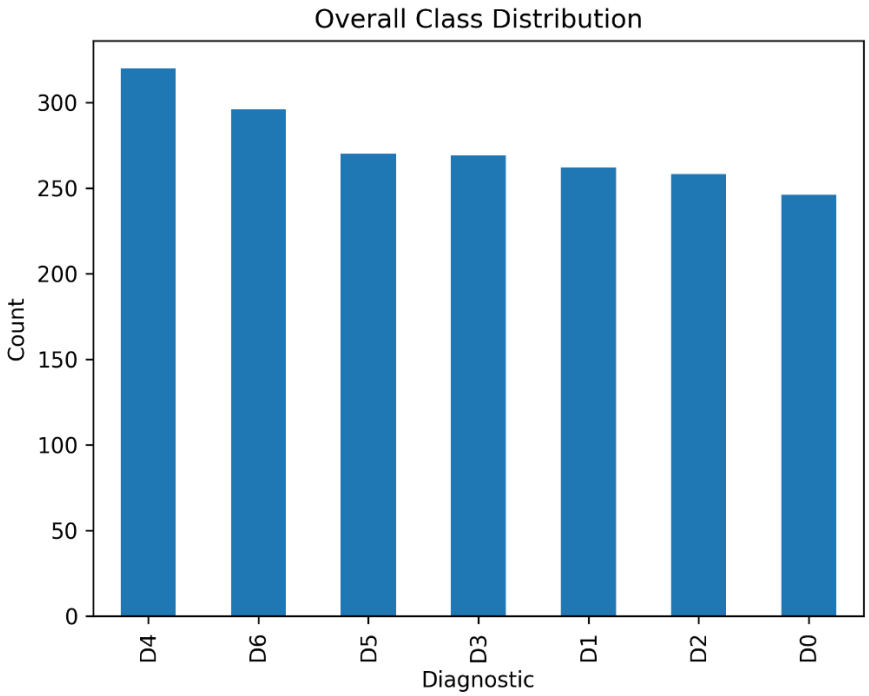
Toate testele au fost rulate pe același set de împărțire aleatoriu, si a fost folosita accelerarea CUDA. Sistemul pe care au fost rulate este:

- Procesor: I7 12700KF 3.6 GHz Base, 5.00 GHz Boost
- Memorie: 32 GB Ram DDR5 5600MHz
- Placa Video: NVIDIA RTX 3070 TI 8GB

2. Analiza datelor

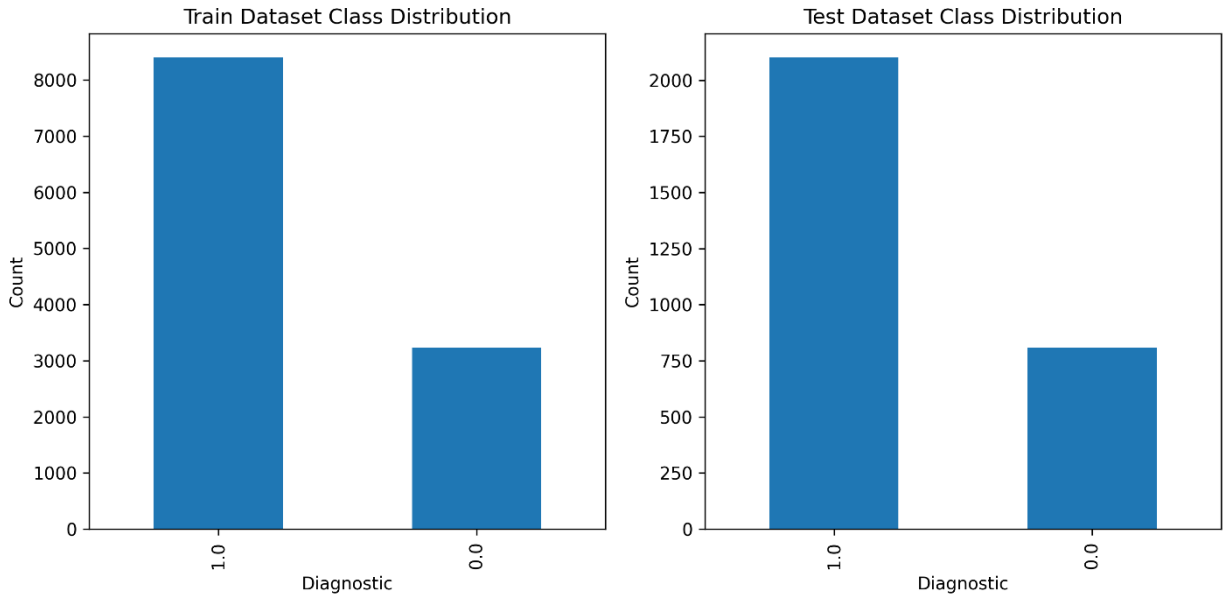
2.1. Analiza distribuție clase

Analiza pentru Diagnostic este aceeași ca cea de la tema 1.



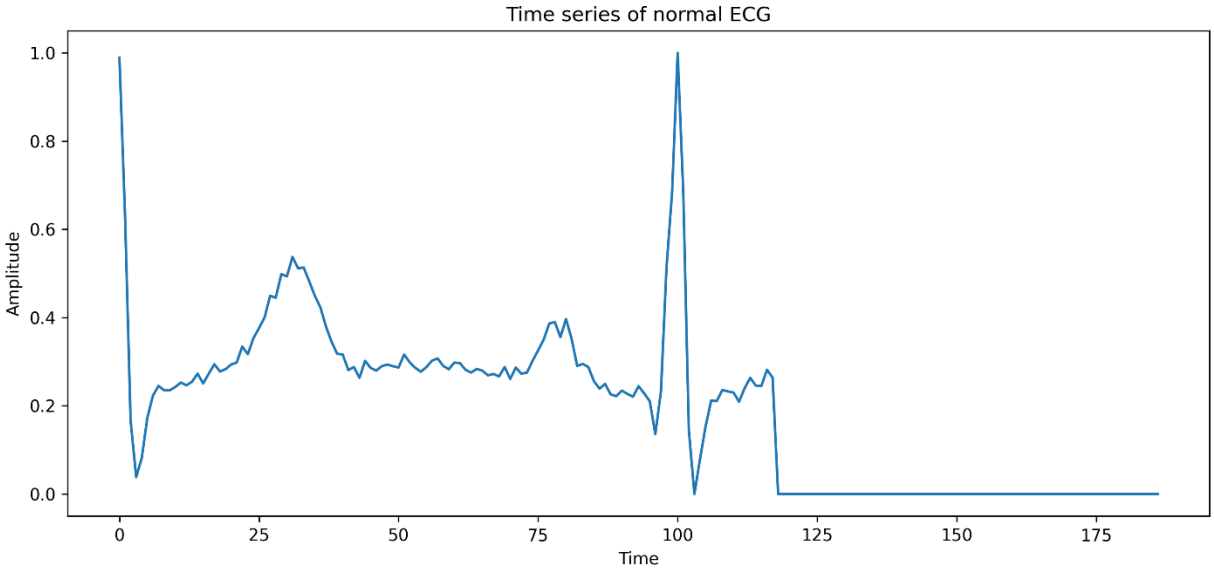
In primul rând putem vedea ca clasa este de tipul multi-clasa, iar pe total distribuția ei este echilibrata. Evident nu este o distribuție perfect egala, dar nici nu mă așteptam la așa ceva.

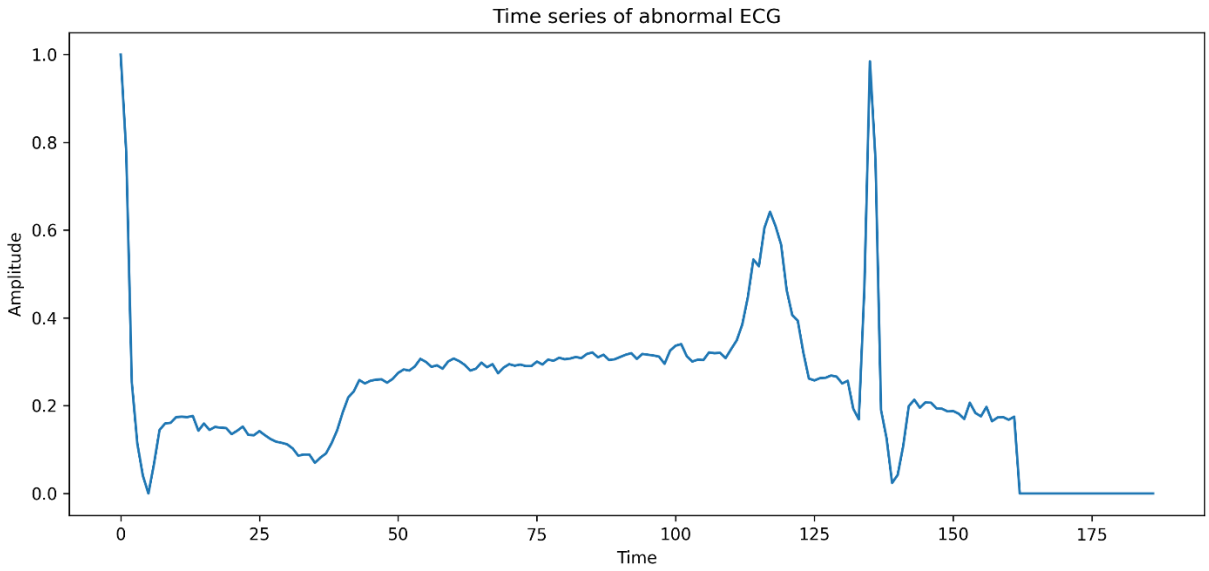
Se poate observa cum proporțional vorbind, împărțirea este relativ aceeași, ceea ce este un lucru dorit, având suficiente date din fiecare tip de Diagnostic in setul de antrenare.



De asemenea, se poate observa ca distribuția pe train si test a aritmiilor este relativ egala, fiind mai multe date normale decât anormale, lucru ce era de așteptat, deoarece sunt mult mai multe date in categoria de anormale (notata cu 1, cea normala este notata cu 0).

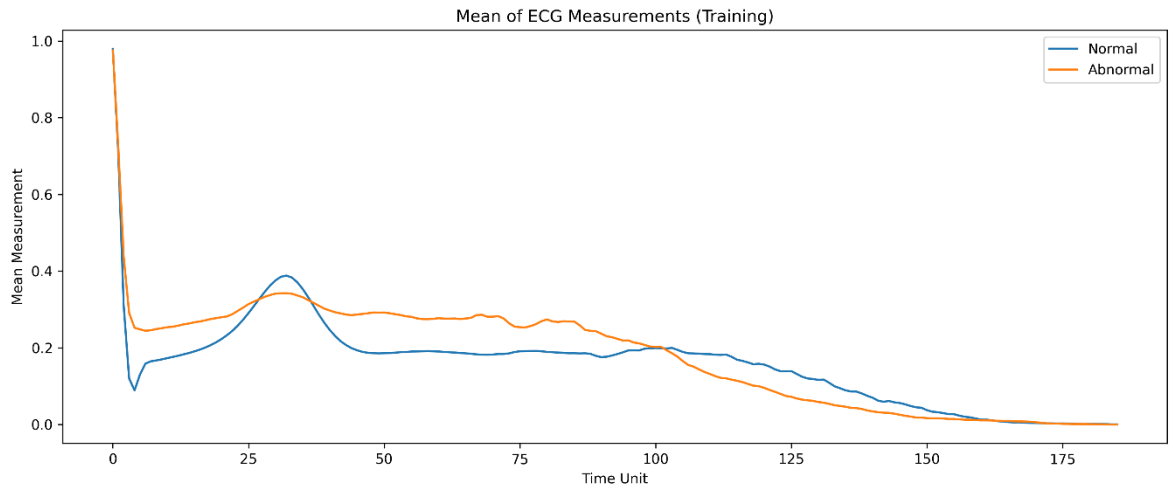
2.2. Seriile timp

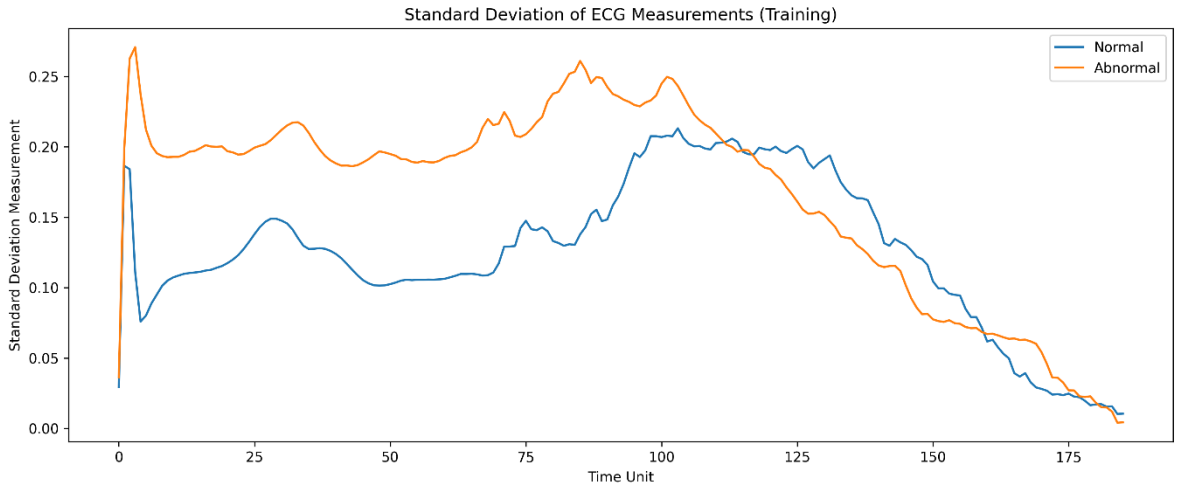


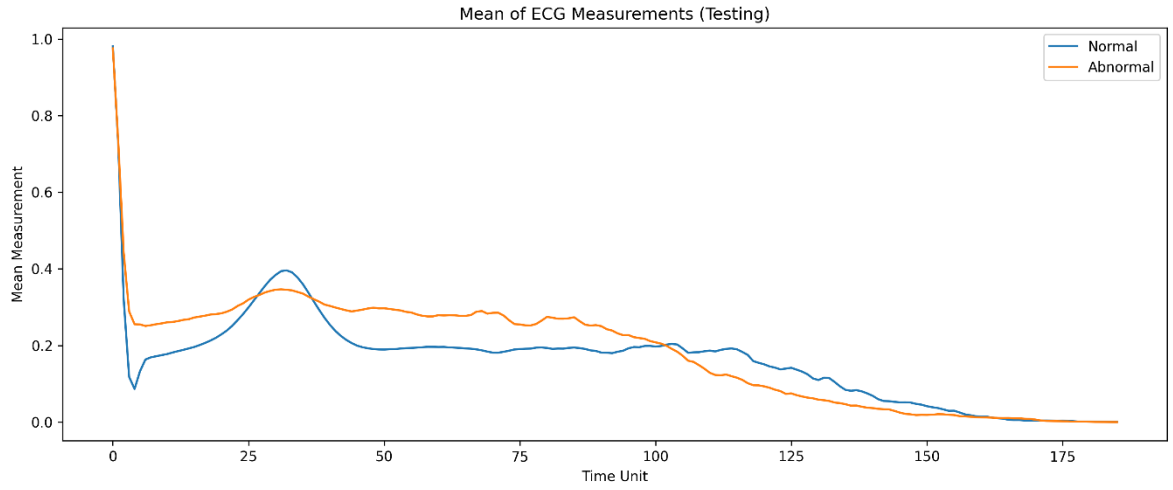


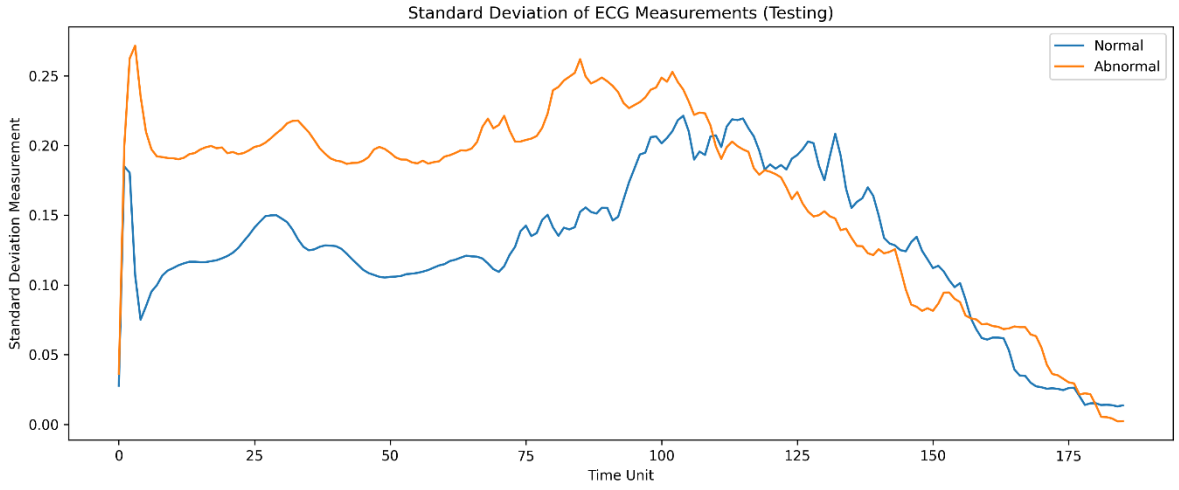
Ochiometric vorbind, pare ca diferența intre o aritmie normala si anormala este lipsa un spike anterior cu pauza înaintea spike-ul mare, si de asemenea si o decalare in spike.

In ceea ce privește media si variația acestor măsurători (figura de mai jos), pare ca se confirma ce am observat mai sus, si anume ca in mod normal, intre 25-50 ar trebui sa fie un spike, care lipsește la anormal, fapt care se vede atât prin media scăzută in acel interval, dar si din variația foarte mare in toata zona 0-75.


Mean of ECG Measurements (Training)
Normal
Abnormal
Mean Measurement
Time Unit


Standard Deviation of ECG Measurements (Training)
Normal
Abnormal
Standard Deviation Measurement
Time Unit


Mean of ECG Measurements (Testing)
Normal
Abnormal
Mean Measurement
Time Unit


Standard Deviation of ECG Measurements (Testing)
Normal
Abnormal
Standard Deviation Measurement
Time Unit

3. Analiza rezultatelor

Pentru toate rețelele folosite, calcul de loss a fost facut cu Cross Entropy.

3.1. Arhitectura de tip Multi-Layered Perceptron

Structura de tip MLP folosita este următoarea:

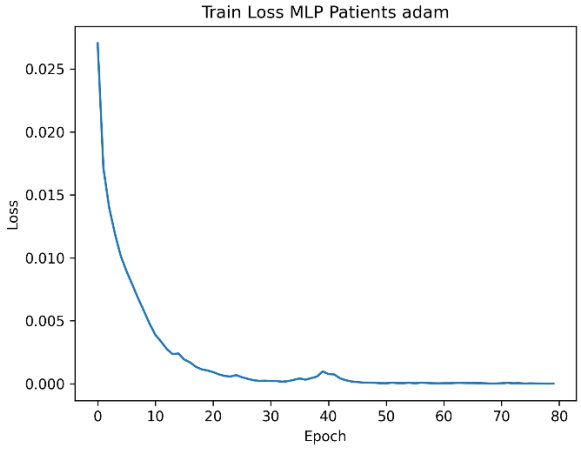
- Fully-Connected(in_features, 512)
- ReLU
- Fully-Connected(512, 256)
- ReLU
- Fully-Connected(256, 128)
- ReLU
- Fully-Connected(128, 64)
- ReLU
- Fully-Connected(64, num_classes)

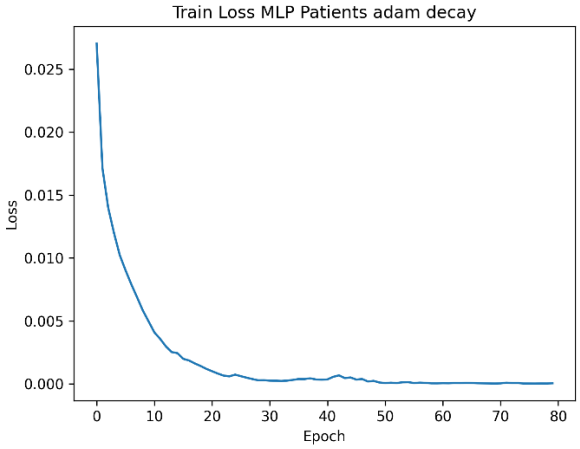
Pentru testare, am variat optimizatorul folosit, dar si batch size-ul, observând ca ADAM merge mai bine pe un batch size de 64, iar SGD pe un batch size de 32. Pentru SGD, learning rate-ul este setat la 0.01, iar momentum-ul este setat la 0.9 (Aceste valori aduc cele mai bune rezultate). Pentru decay am folosit valoarea 0.001.

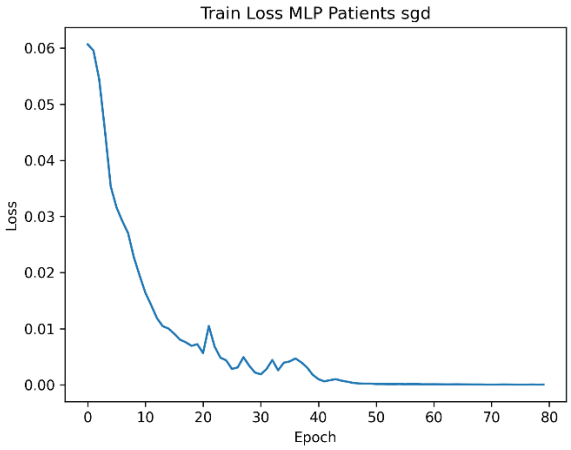
Vom analiza mai întâi setul de date Patients. Rezultatele obținute pe setul de test se pot observa in tabelul de mai jos:

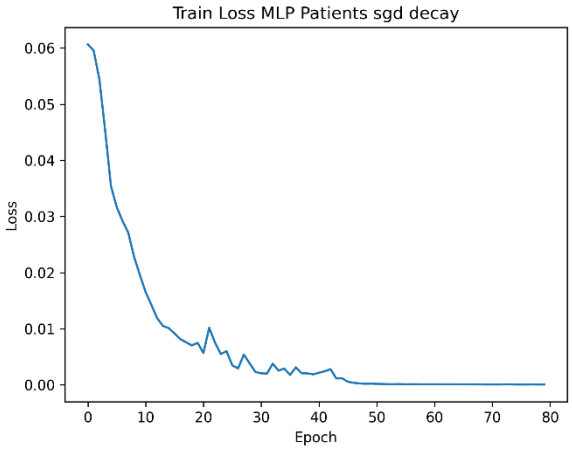
Configuratie	Accuracy	Precision	Recall	F1-score
Adam, Decay	0.99935	0.99935	0.99935	0.99935
Adam, No Decay	0.99935	0.99935	0.99935	0.99935
SGD, Decay	0.99935	0.99935	0.99935	0.99935
SGD, No Decay	0.99935	0.99935	0.99935	0.99935

Se poate observa ca rezultatele obținute sunt foarte bune. O sa analizam training loss-ul pentru a putea vedea ce a dus la astfel de rezultate.

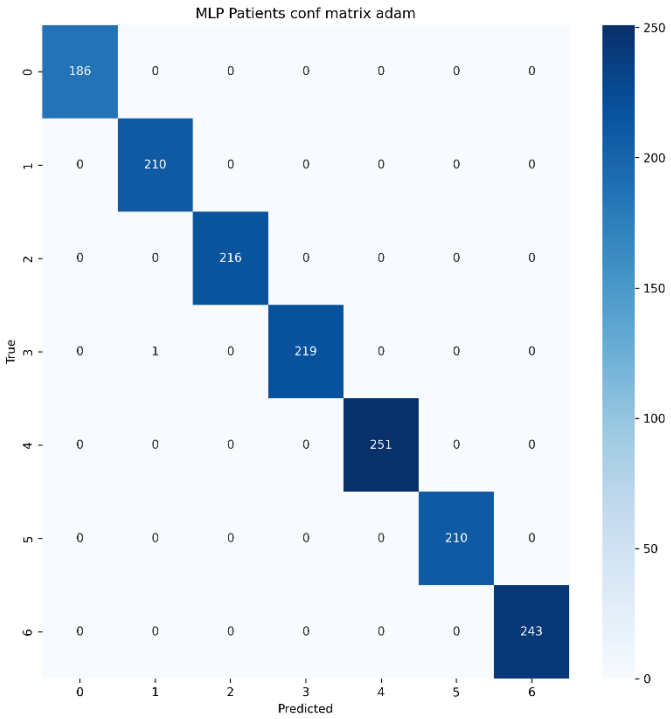


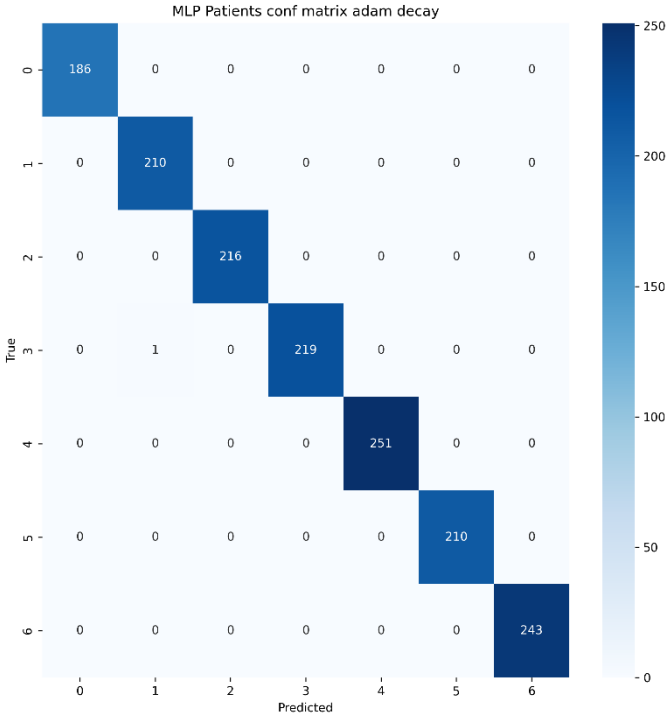


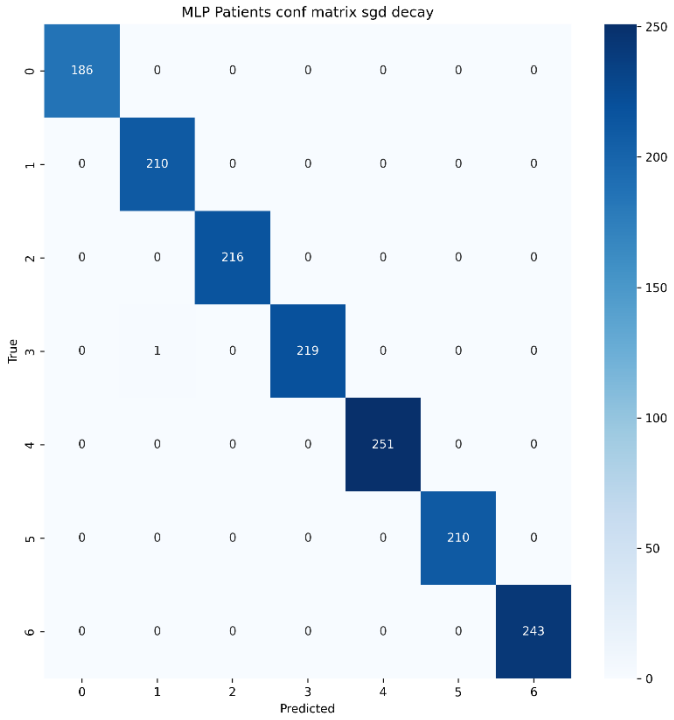


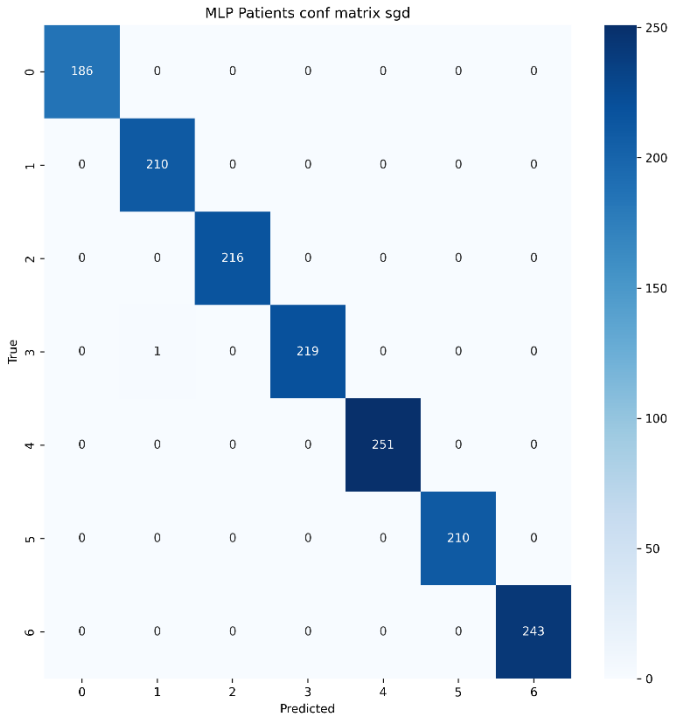


Se poate observa ca toate 4 metodele converg spre zero, dar Adam este puțin mai rapid in acest sens. Acum putem analiza matricele de confuzie generate de fiecare algoritm. Mă aștept sa fie similare, in ideea ca numărul de greșeli ar trebui sa fie egal. De asemenea, garantez ca nu am rulat același model de 4 ori.


MLP Patients conf matrix adam
True
Predicted
0 1 2 3 4 5 6
186 0 0 0 0 0 0
0 210 0 0 0 0 0
0 0 216 0 0 0 0
0 1 0 219 0 0 0
0 0 0 0 251 0 0
0 0 0 0 0 210 0
0 0 0 0 0 0 243
250 200 150 100 50 0


MLP Patients conf matrix adam decay
True
Predicted
0 1 2 3 4 5 6
186 0 0 0 0 0 0
0 210 0 0 0 0 0
0 0 216 0 0 0 0
0 1 0 219 0 0 0
0 0 0 0 251 0 0
0 0 0 0 0 210 0
0 0 0 0 0 0 243
250 200 150 100 50 0



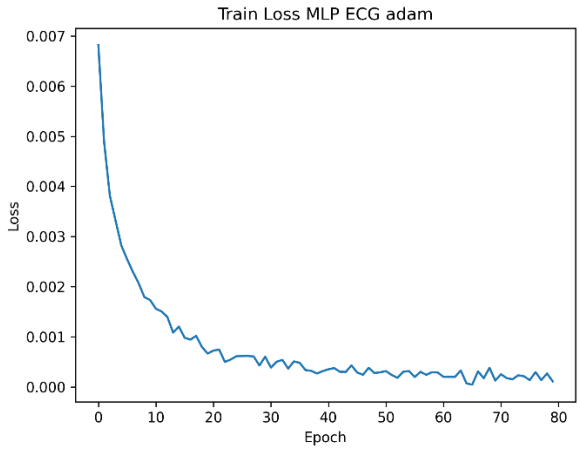


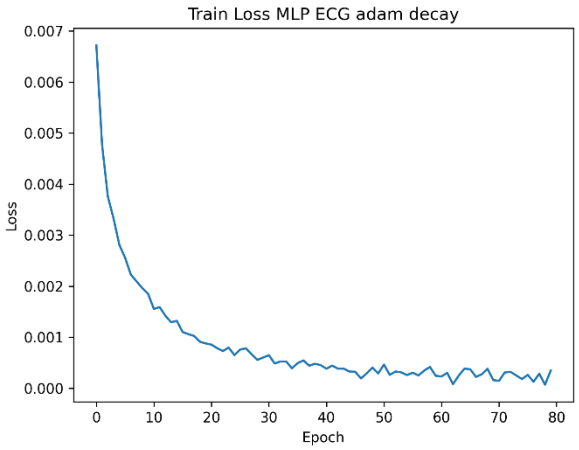
Se poate observa ca in toate cele 4 cazuri, exista doar o eroare, pe diagnosticul D1.

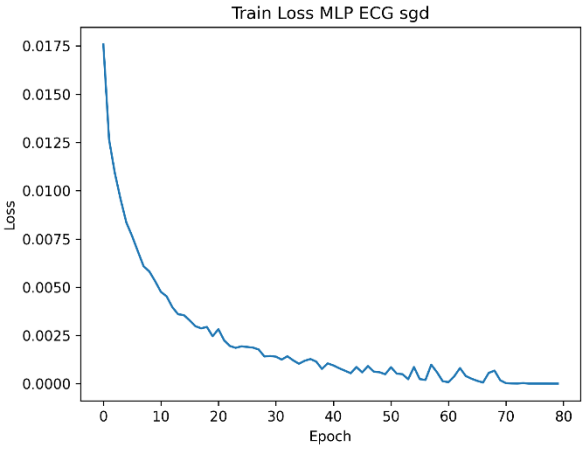
Acum vom analiza setul de date PTB. Rezultatele obtinute sunt:

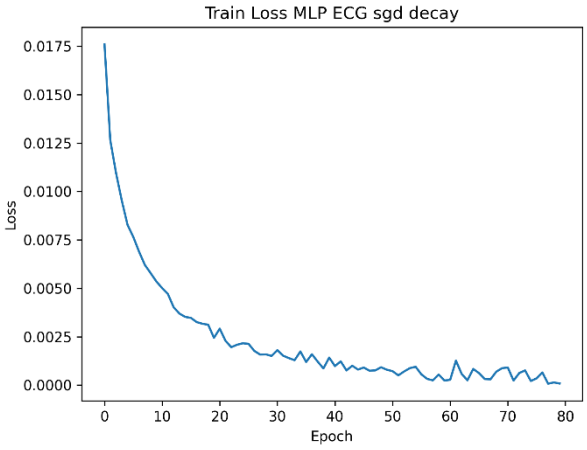
Configuratie	Accuracy	Precision	Recall	F1-score
Adam, Decay 0.001	0.9975	0.9975	0.9975	0.9975
Adam, No Decay	0.9998	0.9998	0.9998	0.9998
SGD, Decay	1.0	1.0	1.0	1.0
SGD, No Decay	1.0	1.0	1.0	1.0

Din nou, niște rezultate foarte bune, aproape perfecte. Din nou, ne așteptam ca training-ul loss-ul sa fie 0 sau spre 0.

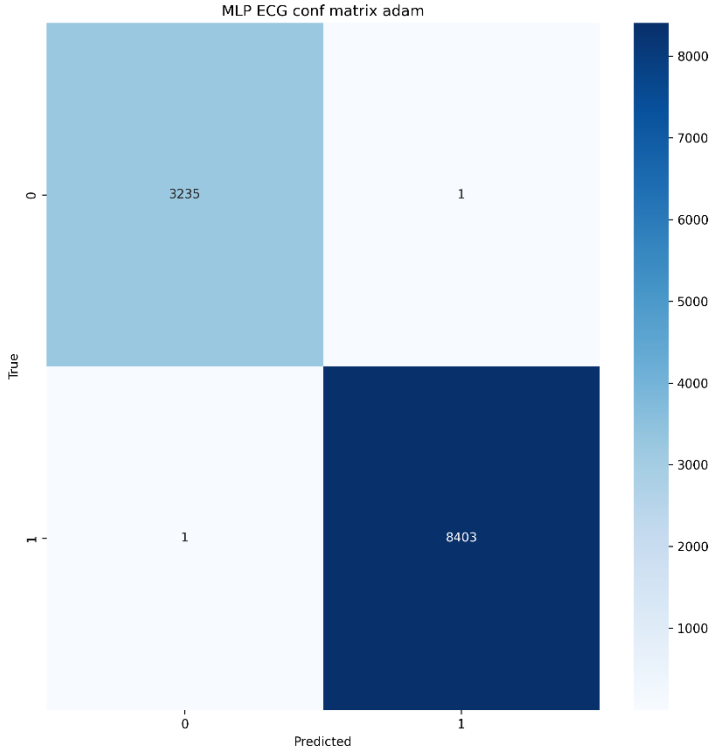


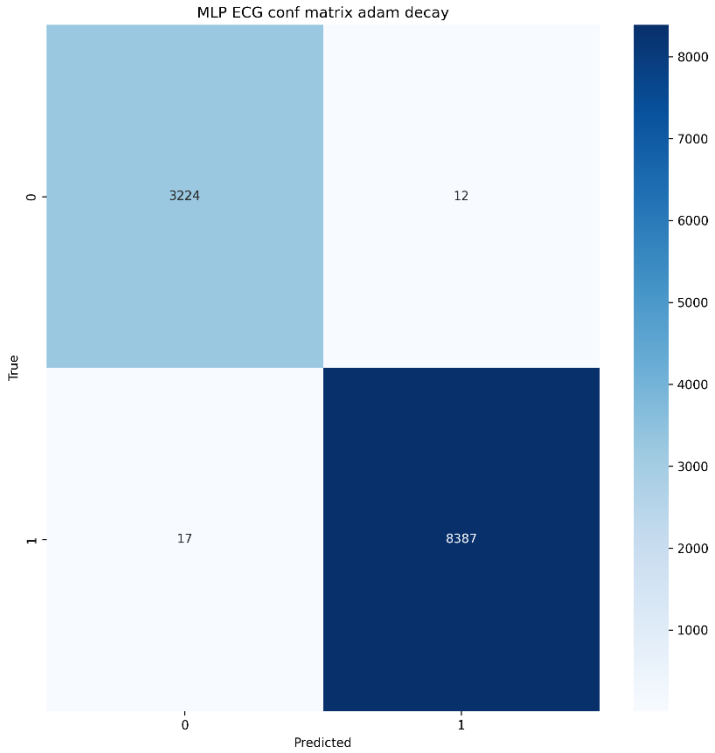


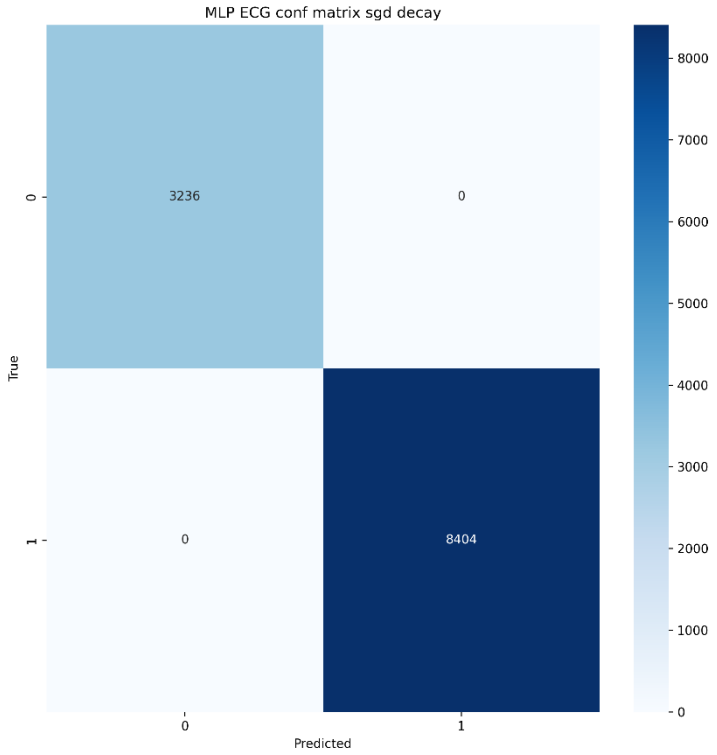


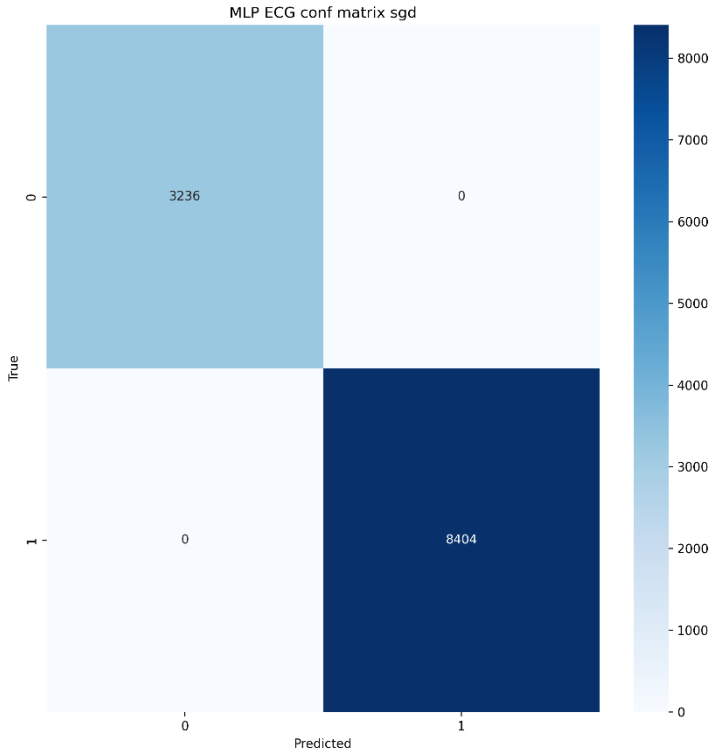


Graficul poate părea destul de instabil, dar este de observat cat de mica este pierderea. (0.0025).


MLP ECG conf matrix adam
3235
1
1
8403
True
Predicted
0
1
8000
7000
6000
5000
4000
3000
2000
1000


MLP ECG conf matrix adam decay
3224
12
17
8387
True
Predicted
0
1
8000
7000
6000
5000
4000
3000
2000
1000





Matricele arata exact cum ne-am fi așteptat, mai exact cele de la SGD au 0 predicții greșite.

3.2. Rețeaua convolutionala

Rețeaua convolutionala folosita este una făcută de mine, inspirata din LeNet5. Structura ei este:

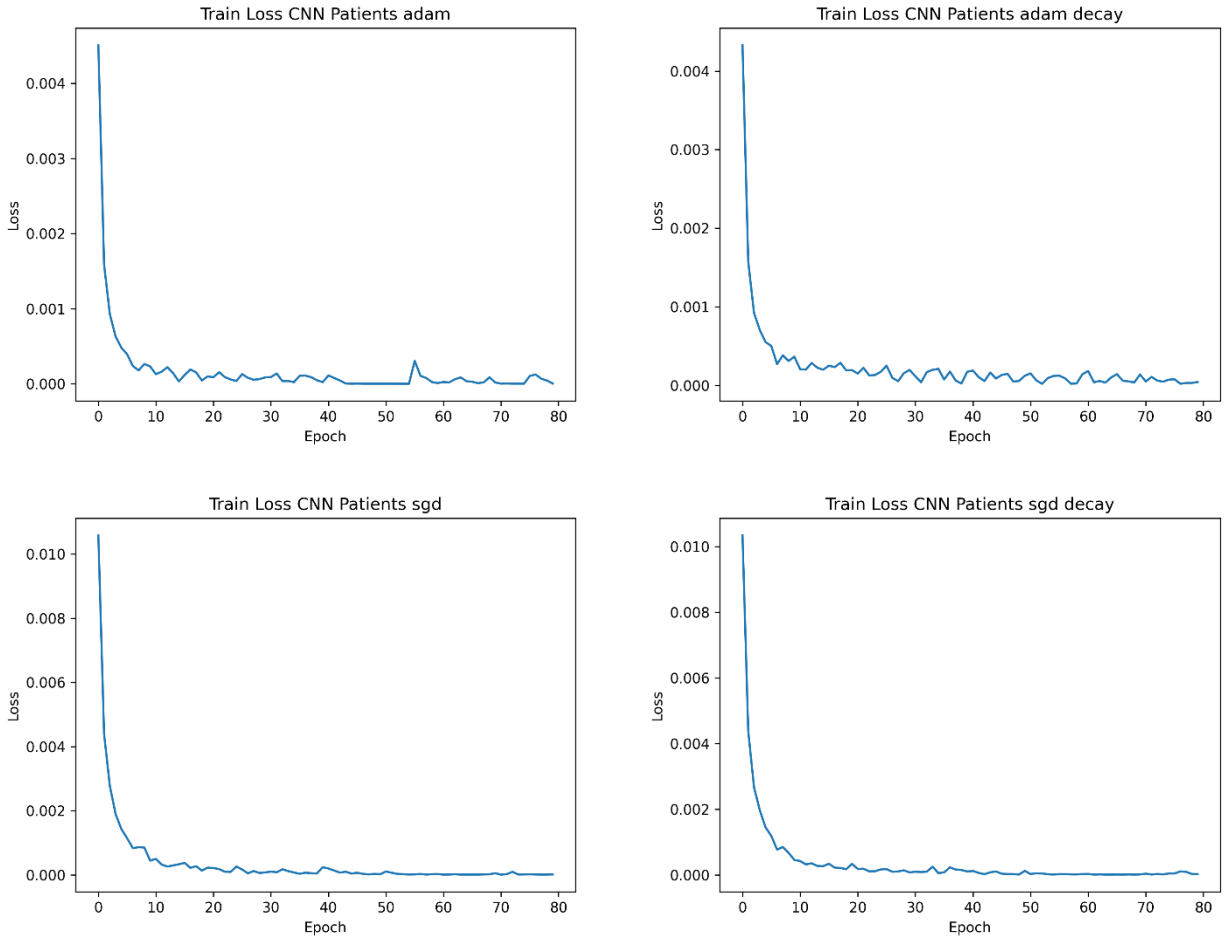
- Conv1d(1, 64, kernel=5)
- BatchNorm1d(64)
- ReLU
- MaxPool1d(kernel=2)
- Conv1d(64, 128, kernel=5)
- BatchNorm1d(128)
- ReLU
- MaxPool1d(kernel=2)
- FullyConnected(128 * 43, 256)
- BatchNorm1d(256)
- ReLU
- FullyConnected(256, 64)
- BatchNorm1d(64)
- ReLU
- FullyConnected(64, 2)
- ReLU

Pentru testare, am variat optimizatorul folosit, dar si batch size-ul, observând ca ADAM merge mai bine pe un batch size de 64, iar SGD pe un batch size de 32. Pentru SGD, learning rate-ul este setat la 0.01, iar momentum-ul este setat la 0.9 (Aceste valori aduc cele mai bune rezultate). Pentru decay am folosit valoarea 0.001.

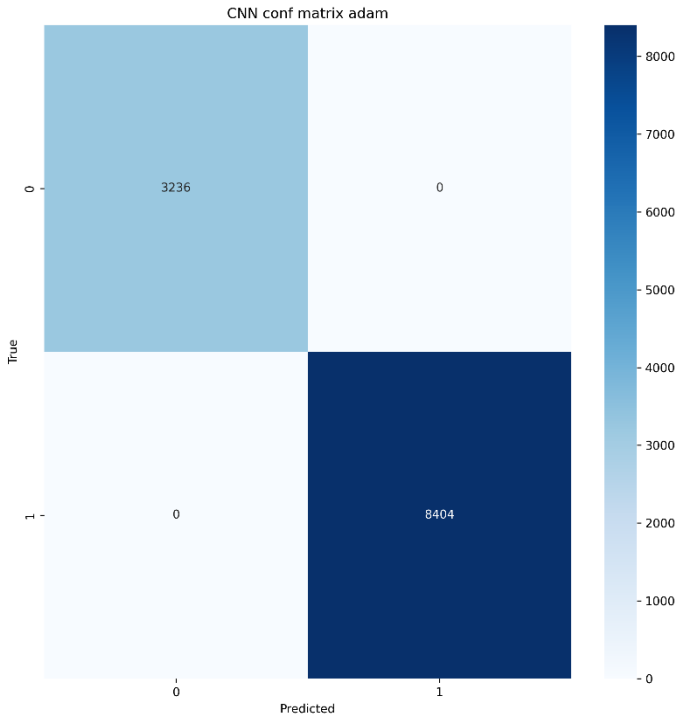
Rezultatele obținute pe setul de test se pot observa in tabelul de mai jos:

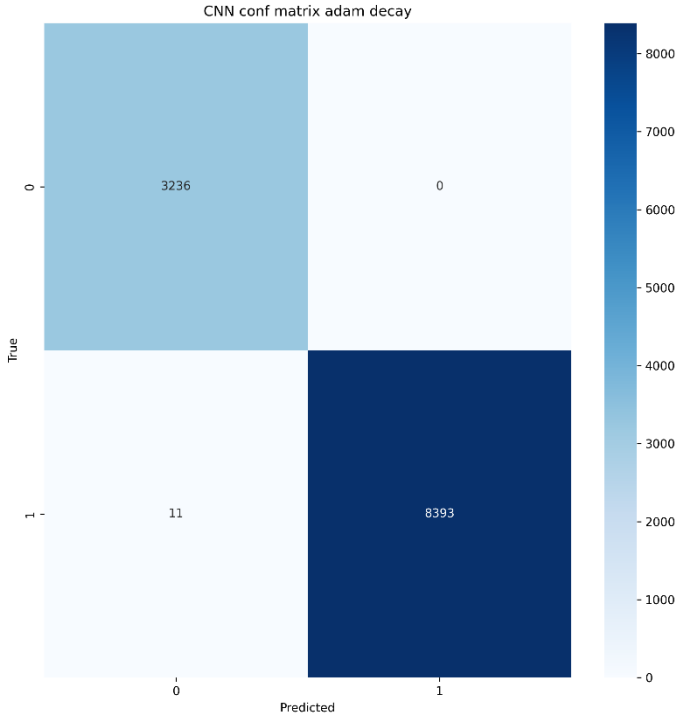
Configuratie	Accuracy	Precision	Recall	F1-score
Adam, Decay	0.999054	0.999056	0.999054	0.999054
Adam, No Decay	1.0	1.0	1.0	1.0
SGD, Decay	1.0	1.0	1.0	1.0
SGD, No Decay	0.99935	0.99935	0.99935	0.99935

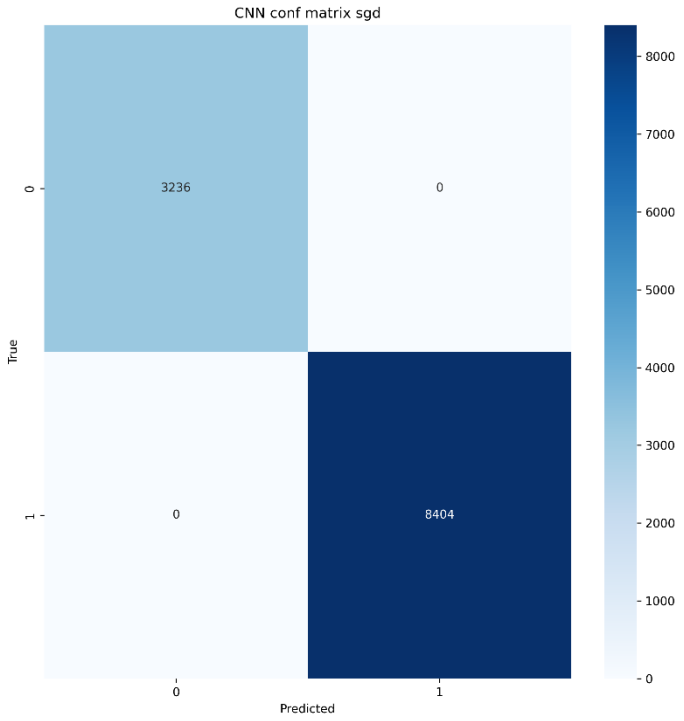
Se poate observa ca rezultatele obținute sunt foarte bune. O sa analizam training loss-ul pentru a putea vedea ce a dus la astfel de rezultate.

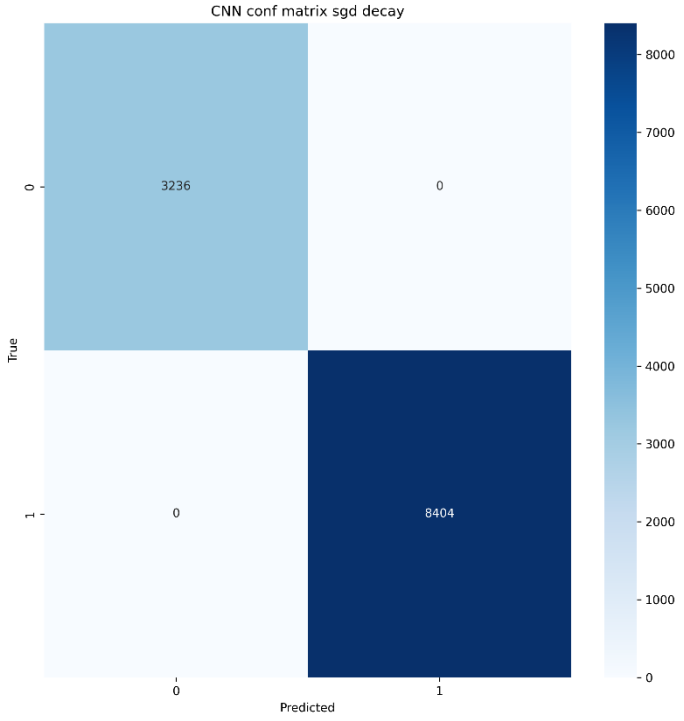


Pentru acest set de date si pe CNN, pare ca optimizatorul SGD s-a comportat mult mai bine, ajunând la flat line foarte rapid, si ne mai având fluctuații după, de aici si rezultatele mai bune.


CNN conf matrix adam
3236
0
0
8404
True
Predicted
0
1
8000
7000
6000
5000
4000
3000
2000
1000
0


CNN conf matrix adam decay
3236
0
11
8393
True
Predicted
0
1
8000
7000
6000
5000
4000
3000
2000
1000
0





Evident, matricele de confuzie sunt perfect, mai puțin pentru Adam cu decay, unde avem 11 false negatives.

4. Concluzii

Per total, rezultatele obținute sunt excepționale, nu mă așteptam la niște preziceri atât de bune de la niciuna dintre rețele. Am rămas cel mai impresionat de rețeaua MLP, crezând ca nu o sa poate sa scoată nici măcar o acuratețe de 95%, dar aparent acele straturi FullyConnected nu sunt chiar așa inutile pe date tip serie timp.