

MINISTERUL EDUCAȚIEI



**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

# CANCER LA SÂN: PREDICȚIA ARIEI TUMORII ȘI CLASIFICARE

PROIECT DE DIPLOMĂ

Autor: **Anca-Elena ANDREESCU**

Conducător științific: **Prof. dr. ing. Eva-Henrietta DULF**

**2024**



Vizat,

DECAN

**Prof. dr. ing. Mihaela DÎNȘOREANU**

DIRECTOR DEPARTAMENT AUTOMATICĂ

**Prof. dr. ing. Honoriu VĂLEAN**

Autor: **Anca-Elena ANDREESCU**

## **Cancer la sân: predicția ariei tumorii și clasificare**

1. **Enunțul temei:** *Realizarea unui model care poate să ofere o predicție cât mai corectă a ariei tumorii depistată în zona mamară. În plus, pentru același set de date, dezvoltarea unui model optim, pentru a oferi un diagnostic precis referitor la caracterul malign sau benign a leziunii.*
2. **Conținutul proiectului:** *Pagina de prezentare, Declarație privind autenticitatea proiectului, Sinteza proiectului, Cuprins, Introducere, Studiu Bibliografic, Analiză, Proiectare, Implementare, Concluzii, Bibliografie*
3. **Locul documentării:** *Universitatea Tehnică din Cluj-Napoca*
4. **Data emiterii temei:** 02.10.2023
5. **Data predării:** 24.06.2024

Semnătura autorului

\_\_\_\_\_

Semnătura conducătorului științific

\_\_\_\_\_

**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

**Declarație pe proprie răspundere privind  
autenticitatea proiectului de diplomă**

Subsemnatul(a) Anca-Elena ANDREESCU, legitimat(ă) cu  
CI/BI seria VX nr. 943559, CNP 6020317385573,  
autorul lucrării:

Cancer la sân: predicția ariei tumorii și clasificare

elaborată în vederea susținerii examenului de finalizare a studiilor de licență la  
**Facultatea de Automatică și Calculatoare**, specializarea **Automatică și Informatică  
Aplicată**, din cadrul Universității Tehnice din Cluj-Napoca, sesiunea iulie 2024 a anului  
universitar 2023-2024, declar pe proprie răspundere, că această lucrare este rezultatul  
propriei activități intelectuale, pe baza cercetărilor mele și pe baza informațiilor obținute  
din surse care au fost citate, în textul lucrării, și în bibliografie.

Declar, că această lucrare nu conține porțiuni plagiate, iar sursele bibliografice au  
fost folosite cu respectarea legislației române și a convențiilor internaționale privind  
drepturile de autor.

Declar, de asemenea, că această lucrare nu a mai fost prezentată în fața unei alte  
comisii de examen de licență.

În cazul constatării ulterioare a unor declarații false, voi suporta sancțiunile  
administrative, respectiv, *anularea examenului de licență*.

Data

24.06.2024

Anca-Elena ANDREESCU

(semnătura)



## SINTEZA

proiectului de diplomă cu titlul:

### **Cancer la sân: predicția ariei tumorii și clasificare**

Autor: **Anca-Elena ANDREESCU**

Conducător științific: **Prof. dr. ing. Eva-Henrietta DULF**

1. Cerințele temei: Implementarea unui model care este capabil să ofere o acuratețe crescută atât în procesul de predicție a ariei tumorii, cât și pentru partea de caracterizare a tumorii în cele două clase: malignă sau benignă.
2. Soluții alese: Prin utilizarea limbajului de programare Python, s-a antrenat o rețea neuronală artificială (ANN) care rezolvă problema de regresie bazată pe predicția ariei. În plus, o altă rețea artificială a fost construită pentru soluționarea subiectului bazat pe împărțirea tumorii în cele două categorii: malignă și benignă.
3. Rezultate obținute: Pentru partea de regresie s-a obținut o eroare medie pătratică de 0.0001 și un coeficient de determinare ( $R^2$ ) de 0.99. Pe de altă parte, pentru rețeaua neuronală care se ocupă de divizarea neoplaziei în cele două categorii, s-a obținut o acuratețe de aproximativ 98%.
4. Testări și verificări: Din momentul în care rețelele neuronale artificiale au fost construite, s-au încercat mai multe combinații între diverși parametri, astfel s-au realizat schimbări în rândul valorilor care descriu: numărul de straturi ascunse, numărul de neuroni de pe starturi, epoci. Modificările s-au putut observa în valoarea erorii mediei pătratice, coeficientului de determinare, dar și în modul în care se remodelau graficele destinate comparației între datele reale și cele furnizate prin predicție.
5. Contribuții personale: Documentarea asupra modului optim de construire a rețelelor neuronale și implementarea acestuia.
6. Surse de documentare: Sursele de documentare utilizate au constat în articole științifice, dar și cărți.

Semnătura autorului

\_\_\_\_\_

Semnătura conducătorului științific

\_\_\_\_\_

# Cuprins

<b>1</b>	<b>INTRODUCERE.....</b>	<b>2</b>
1.1	CONTEXT GENERAL .....	2
1.2	OBIECTIVE.....	4
1.3	SPECIFICAȚII .....	4
<b>2</b>	<b>STUDIU BIBLIOGRAFIC.....</b>	<b>5</b>
<b>3</b>	<b>ANALIZĂ, PROIECTARE, IMPLEMENTARE.....</b>	<b>15</b>
3.1	MEDIU DE DEZVOLTARE .....	15
3.2	CARACTERISTICILE SETULUI DE DATE .....	16
3.3	REȚEAUA NEURONALĂ ARTIFICIALĂ DESTINATĂ PREDICȚIEI ARIEI TUMORII MAMARE.....	18
3.4	REȚEAUA NEURONALĂ ARTIFICIALĂ PENTRU CLASIFICAREA TUMORII MAMARE .....	24
3.5	TESTE REALIZATE PENTRU A AJUNGE LA FORMA OPTIMĂ.....	27
3.5.1	<i>Teste realizate pentru rețeau neuronală artificială destinată predicției ariei .....</i>	<i>28</i>
3.5.2	<i>Teste realizate pentru rețeaua neuronală artificială destinată clasificării tumorii .....</i>	<i>33</i>
<b>4</b>	<b>CONCLUZII.....</b>	<b>38</b>
4.1	REZULTATELE OBTINUTE.....	38
4.2	DIRECȚII DE DEZVOLTARE.....	40
<b>5</b>	<b>BIBLIOGRAFIE.....</b>	<b>42</b>

# 1 Introducere

## 1.1 Context general

Cancerul de sân a reprezentat dintotdeauna unul dintre principalele motive de deces în rândul femeilor, aproximativ 15% din numărul total de cazuri raportate anual.[1] În anul 2020, statisticile au oferit un număr apropiat de 2.3 milioane de cazuri noi apărute în rândul persoanelor de sex feminin, dintre care 685.000 de femei au trecut în neființă. Se preconizează o creștere accelerată a numărului de cazuri, astfel în anul 2040, persoanele care o să dețină acest diagnostic o să depășească 3 milioane. Din această valoare, decesele o să reprezinte o treime.[7] Se poate observa tendința de înmulțire a numărului de femei care se confruntă cu această neoplazie, rezultând o necesitate acută pentru găsirea unor modalități de cunoaștere și de înțelegere mai aprofundată a modului de expansiune a bolii.

Din acest motiv, importanța identificării diagnosticului corect conduce la aplicarea unui tratament eficient aplicat pacienților. [1] Șansele de viață a femeilor, pot să fie influențate major de momentul în care este confirmat rezultatul medical, astfel se evidențiază relevanța cunoașterii timpurii a diagnosticului. Cu cât procedura medicală este administrată din timp, cu atât probabilitatea de supraviețuire a persoanelor se amplifică.

Prin această lucrare, s-au atins două puncte vitale care au rolul de a caracteriza o tumoare: modul în care o să evolueze dimensiunea ariei leziunii, bazat pe anumite valori medicale, respectiv dacă aceasta are o caracteristică malignă sau benignă. Tumoarea reprezintă o acumularea excesivă de celule într-o zonă a organismului. Principalul motiv pentru care se formează această formațiune, fiind reprezentat de operația celulelor de a se divide anormal de mult într-un timp scurt sau din simplu fapt că acestea nu mor într-un timp optim așteptat de către organism. Tumoarea benignă în mod normal nu cauzează împrejurări dificile pacienților, dar chiar dacă are un ritm lent de creștere, poate să ajungă în punctul în care să influențeze funcționarea normală și eficientă a celorlalte organe din corpul uman. Pe de altă parte, tumoarea malignă se caracterizează printr-un ritm rapid și necontrolat de răspândire în tot organismul. Ele reprezintă un real pericol pentru corpul uman, reușind să acapereze zona inițială, dar și se răspândesc în tot corpul prin intermediul sângelui.[2] Aceste aspecte arată importanța cunoașterii tipului tumorii, astfel lucrarea prezintă procesul prin care s-a studiat și încercat găsirea unui algoritm cât mai potrivit pentru această sarcină.

De sigur, cunoscând caracteristica tumorii maligne, creștere sporadică, se justifică și motivația medicilor de a cunoaște maniera de evoluție a mărimii afecțiunii. Această lucrare reflectă dorința de a ajuta și de a ușura modul de gestionare a tratamentului oncologic aplicat oamenilor. Cancerul reprezentând o problemă medicală cu o frecvență

de expansiune înaltă în viața cotidiană, este foarte important orice detaliu suplimentar despre maniera de dezvoltare în organismul uman.

De-a lungul anilor, aria medicinei a cunoscut o dezvoltare cu un imens impact în îngrijirea medicală, astfel reușind să ofere tratamente adecvate și mult mai eficiente pentru pacienți. Totuși, învățarea automată poate să aducă un beneficiu considerabil în clasificarea tumorii, dar și pentru modul în care aria formațiunii o să avanseze pe parcursul timpului. [1] O astfel de predicție eficientă asupra particularității neoplasmului poate să îi ajute și pe specialiștii oncologi să ofere o îngrijire medicală adecvată și personalizată fiecărui pacient. Pe parcursul anilor s-a încercat introducerea învățării automate în sprijinul medicinei, iar acest lucru a dus la lucruri inovatoare, având o consecință pozitivă în tratarea diferitelor afecțiuni. S-au efectuat foarte multe studii pe toate sferele medicinei. Partea destinată cancerului este una care reușește să ocupe un loc principal în majoritatea studiilor și cercetărilor, deoarece pentru această afecțiune nu există un leac sau un tratament care să ofere o marjă crescută în cea ce privește vindecarea.

Inteligența artificială (AI) a devenit tot mai prezentă în viața de zi cu zi a oamenilor. Programatorii au început să o utilizeze pentru a putea oferi metode simplificate și cu o acuratețe crescută, în majoritatea problemelor existente. Fiind un domeniu în continuă dezvoltare și reprezentând o sferă de interes pentru cei mai mulți informaticeni, se încearcă răspândirea ei pe diverse domenii de activitate. Învățarea automată este un subdomeniu al inteligenței artificiale. Cu ajutorul acestei tehnologii se crează anumiți algoritmi care pot să învețe cum să ofere o predicție cât mai aproape de adevăr, pe baza unor seturi de date și relații între acestea.

Învățarea automată pune la dispoziție diferite modalități și algoritmi mai complecși pentru rezolvarea problemelor. Programatorul are sarcina de a alege tehnica care îl poate conduce la rezultate cât mai satisfăcătoare. O metodă foarte des întâlnită pentru învățarea automată este reprezentată de rețelele neuronale artificiale (ANN). Această metodă încearcă să simuleze modul de gândire a oamenilor. Creierul uman este programat să ofere o capacitate mare de acumulare a informațiilor. Acesta este alcătuit dintr-o organizare de aproximativ 10 miliarde de neuroni care comunică între ei cu ajutorul sinapselor. Fiecare celulă de neuron are posibilitatea de a primi, procesa și învăța o informație nouă. Astfel se vor executa două tipuri de procese relevante, unul în care rețeaua încearcă să învețe ce trebuie să ofere la ieșire și unul în care se testează rezultatele obținute.[4]

Lucrarea de față oferă o soluție pentru medicii de pe secția de oncologie, să cunoască mai multe informații semnificative la adresa tumorii. Acesta combină necesitatea medicală cu învățarea automată, astfel putând să se ajungă la un mod de lucru mai eficient. Cunoșcând felul de evoluție a patologiei, cadrele din domeniul medical pot să aibă o idee despre cum o să se dezvolte stadiul bolii, astfel reușind să prescrie un tratament mult mai performant.

În continuare sunt prezentate capitole în care sunt explicate concis partea de implementare a codului, concluzii, dar și testele realizate asupra algoritmilor obținuți. Setul de date pe care s-au realizat aceste rețele neuronale este obținut de pe UI machine

learning, sustrase din imagini digitalizate a unor mase mamare. Baza de date se numeste Breast Cancer Wisconsin (Diagnostic). Acesta include 569 de instanțe care descriu caracteristicile celulelor neoplaziei. [3]

## **1.2 Obiective**

Lucrarea de față are două obiective principale, să ofere o înțelegere mai profundă a modului în care o să se dezvolte aria tumorii prezente în zona mamară, dar să pună la dispoziție și un algoritm de clasificare a tipul formațiunii tumorale cu o acuratețe crescută.

Predicția dimensiunii neoplasmului și clasificarea tipului acestuia, au fost realizate cu ajutorul unui set de date de tip numeric. Acestea conțin informații importante legate de diametru, textură, rază (distanța medie de la punctele de pe perimetru), compactitate, concavitate (severitatea porțiunilor concave ale conturului), puncte concave (numărul de porțiuni concave ale conturului), simetrie, netezimea (variația locală a lungimilor razei), diagnostic (B = benignă, M= malignă), aria tumorii.[3] Toate aceste informații provenite de la mase mamare, au contribuit la realizarea rețelelor neuronale artificiale, având ca și scop final perfecționarea diagnosticului și tratamentului cancerului la sân.

Prin aceste modele, doctorii ar avea posibilitatea să beneficieze de o privire de ansamblu asupra modului în care urmează să gestioneze situația în care se află pacientul. Tot odata, aceste modele au ca și scop să elimine eroarea umană, riscul unei diagnosticări eronate, dar și crearea unui tratament cât mai personalizat.

## **1.3 Specificații**

Domeniul pe care îl vizează lucrarea de față, este cel medical, venind în sprijinul doctorilor de pe secția de oncologie care oferă tratamente pentru femeile care suferă de afecțiuni canceroase în zona sânilor. Cunoașterea tipului tumorii, malignă sau benignă, impactează tipul medicației oferit către administrarea persoanelor. Medicul având o viziune pe ansamblu asupra modului în care trebuie abordată problema de tratare, v-a putea să își folosească cunoștințele medicale pentru a oferi o medicație sau o soluție fezabilă pentru a crește șansele de viață a persoanelor de sex feminin.

Pentru un asemenea domeniu este foarte important să nu se greșească diagnosticul, fiind o boală care se agravează rapid. De asemenea, este foarte periculoasă din motivul că poate să nu aibă simptome vizibile care să ofere un semnal de alarmă, astfel femeile putând ajunge să descopere destul de târziu că au o asemenea patologie prezentă în regiunea mamară. Din acest motiv, pentru a putea să fie folosite rezultatele obținute în această documentație, modele realizate trebuie să aibă o acuratețe suficient de mare și să fie apte să ofere rezultate cât mai apropiate de cele din viața reală. Eroarea de diagnostic trebuie să fie scăzută exponențial, din acest motiv introducerea învățării automate în sfera medicală ar micșora rata administrării unor tratamente care nu o să aibă eficiență. Astfel, printr-o implementare avansată, lucrarea ar trebui să furnizeze un mijloc de ajutor prin care să faciliteze modul de lucru pentru secțiile destinate tratării afecțiunilor oncologice.



## 2 Studiu bibliografic

Învățarea automată este studiul care se ocupă cu crearea de algoritmi bazați pe realizarea unei sarcini fără a fi programați explicit pentru acest lucru. Ea a fost explorată cu scopul de a ușura viața oamenilor și de a oferi soluții mai optime la probleme avansate. Principala sursă care oferă posibilitatea învățării automate să funcționeze atât de bine, o reprezintă seturile de date voluminoase. Având în vedere cantitatea mărită de seturi de date care există în ziua de astăzi, învățarea automată a reușit să se impună ca fiind o alternativă fezabilă în soluționarea eficientă și simplificată a provărilor bazate pe date.[5]

Ea utilizează elemente cu o capacitate de optimizare avansată față de procedurile existente în trecut. Are potențialul să creeze relații și corelații între datele primite, astfel reușind să observe un tipar sau anumite caracteristici ale datelor de intrare. De-a lungul anilor s-au realizat diverse comparații între învățarea automată și diferite metode clasice de predicție, dar învățarea automată a reușit să ofere rezultate mult mai apropiate de adevăr.[1]

Învățarea automată oferă posibilitatea a trei tipuri de învățare: supravegheată, nesupravegheată sau semi-supravegheată. Diferențele dintre ele sunt reprezentate de modul de funcționare a algorimilor. În cazul celei supravegheate, ieșirea este precisă în funcție de caracteristicile intrării, iar seturile de date sunt divizate în set de antrenare și set de testare.[5] Pe baza setului de antrenare se realizează învățarea, iar pe cel de validare o să se facă compararea între ce se dorește să se obțină și ce a reușit algoritmul să prezică. Se folosește un set diferit pentru partea de validare, pentru că se dorește verificarea algoritmului pe un set diferit față de cel pe care și-a realizat procesul de antrenare. Algoritmi caracteristici învățării nesupravegheate au ca și particularitate faptul că au libertatea să învețe prin descoperire.[5]

Învățarea semi-supravegheată este o combinație între cele două prezentate mai sus. Este potrivită pentru date care sunt etichetate, dar și ne-etichetate.[10] Toate trei metodele oferă avantaje și dezavantaje în modul de implementare, programatorul trebuie să aleagă cu atenție ce i se potrivește în soluționarea cazului destinat lui.

Lucrarea aceasta se bazează pe o învățare supravegheată. Datele de intrare sunt strâns legate de datele dorite la ieșire, astfel creându-se o relație care ajută algoritmul să ofere răspunsuri cu o acuratețe crescută. În plus, s-a realizat și o împărțire a datelor de antrenare și de validare. Setul de date corespunzător antrenării reprezintă 80% din total, iar cel de testare reprezintă un procent de 20%. În general, pentru probleme de predicție și de clasificare este recomandat folosirea învățării supravegheate.

În figura 1, se poate observa modul de funcționare a metodei de învățare supravegheată.

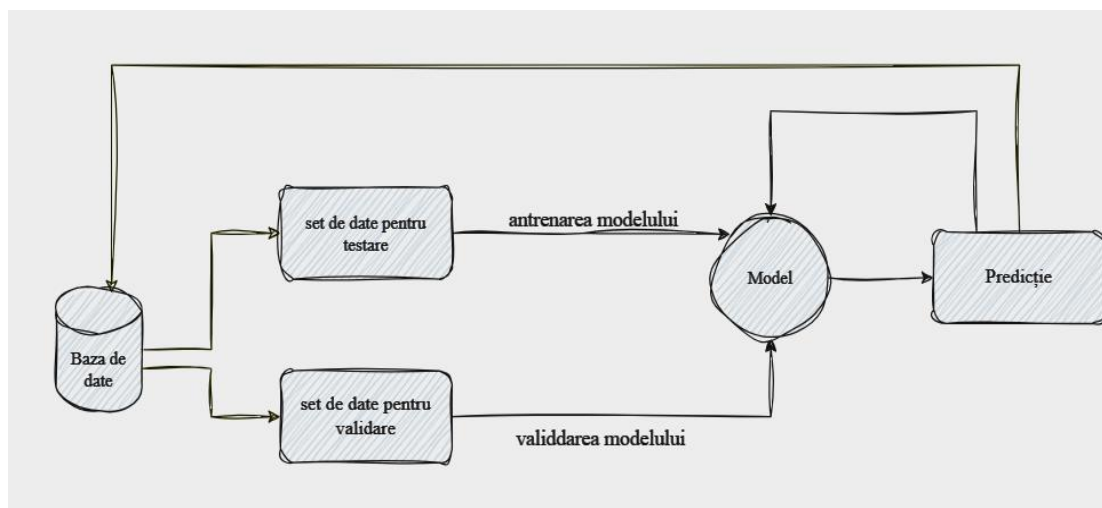


Figura 2.1. Învățare supravegheată

Datorită acestei împărțiri, învățarea automată a cunoscut apoi o nouă fragmentare, astfel sunt anumite particularități ai algoritmilor care îi fac mai potriviți pentru rezolvarea unor probleme de regreie sau de clasificare. Clasificarea se bazează pe distribuirea valorilor de ieșire în anumite clase, practic se oferă o etichetă ieșirilor pe baza anumitor proprietăți. [10]

Pentru a soluționa problema de clasificare a trebuit să se folosească o metodă de codificare a etichetelor (label encoding). În setul de date utilizat, coloana în care se află informația despre tumoare, dacă este malignă sau benignă, este prezentă sub formă de text. Pentru a putea lucra cu aceste date, a trebuit să fie convertite în numere.[8] Prin intermediul acestei tehnici, datele care descriau tumoare ca fiind malignă (M) sau benignă (B) au fost transformate în valorile de 0 sau 1. În acest mod, persoanele care citesc datele o să știe că în momentul în care văd valoarea de 0, tumoarea este benignă, iar pentru valoarea de 1, tumoarea este considerată periculoasă, malignă.

Pe de altă parte, regresia soluționează situații în care se dorește predicția unor valori continue [10]. În acest context, valorile care simbolizează aria tumorii sunt numere reale care nu pot să fie clasificate într-un anumit domeniu.

Prin utilizarea unei învățări supravegheate se pot utiliza mai mulți algoritmi avansați cum ar fi: rețele neuronale artificiale (ANN), decision tree (DT), support vector machine (SVM). În cadrul acestui proiect s-a ales abordarea implementării unor rețele neuronale artificiale.

O rețea neuronală artificială este formată din trei straturi: stratul de intrare, straturile ascunse și stratul de ieșire. Stratul de intrare este cel care conține informațiile care vor participa la procesul de învățare. Straturi ascunse fac legătura între stratul de intrare și cel de ieșire. Ultimul nivel reprezintă de fapt rezultatul la care dorim să ajungem, în cazul lucrării prezente dacă tumoarea are o caracteristică canceroasă sau nu, și aria suprafeței tumorii. Ca și în cazul oamenilor, aceste rețele au împărțită informația în două, una pe care învață și o parte pe care testează ce a învățat. De asemenea, rețelele folosesc modul de repetare a informației, trece de mai multe ori prin testul de antrenare ca să ajusteze ce a învățat. Tot acest proces este asemănător cu ce se petrece în creierul unui

om în momentul în care absoarbe un lucru nou.[1] Omul își crează diferite conexiuni între ce trebuie să memoreze și ce cunoaște deja. El ajunge să repete o informația pe care trebuie să o rețină până în momentul în care acesta este capabil să reproducă cu o acuratețe crescută informația de care are nevoie. Același lucru se petrece și într-o rețea neuronală artificială, cu ajutorul epocilor repetă conținutul. În acest proiect ambele scopuri au fost atinse prin realizarea a două rețele neuronale, una care rezolvă problema de regresie și una care se pretează pentru problema de clasificare.

În figura 2.2, este atașată structura de baza a unui ANN. Acesta prezintă un singur strat ascuns cu 4 neuroni, un strat de input de 3 neuroni și un strat care înfățișează ieșirea. Se pot observa relațiile care se construiesc între neuronii de pe toate straturile rețelei. Fiecare linie trasată de la un neuron la altu, semnifică conexiunile pe care neuronii și le formează. Fiecare conexiune are ca și caracteristică o pondere care alterează semnalul trimis.[4]

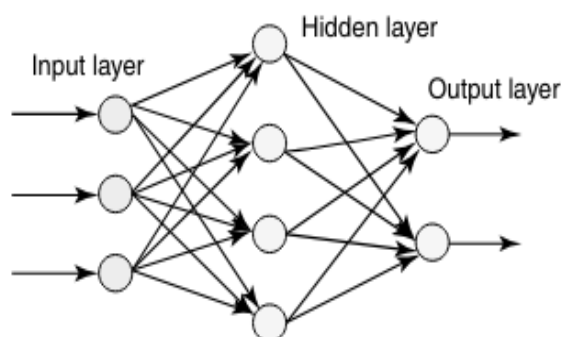


Figura 2.2 înfățișarea unui ANN

După momentul în care s-a construit rețeaua neuronală artificială, urmează pasul în care acesta este supusă proceselor de antrenare și de testare. Anterior rețelei, cum s-a pus în evidență mai sus, setul de date a trebuit să suporte și el un proces de preprocesare și de împărțire. Acesta poate să fie împărțit după cum consideră programatorul, dar și după felul în care modelul reușește să furnizeze o acuratețe suficient de mare. Regula ar fi că setul de date de antrenare ar trebui să fie mai mare sau egal cu cel de validare. Pentru acest proiect, modelul a funcționat cel mai bine și a reușit să ofere cele mai bune rezultate la o împărțire de 80% antrenare și 20% testare.

Un obstacol foarte des întâlnit în problemele care utilizează învățarea automată, este evidențiat prin prezența conceptului de rețea neuronală artificială supra – antrenată sau sub-antrenată. În momentul în care o rețea este supra – antrenată, acesta pe parcursul procesului de antrenare memorează setul de antrenare, ajungând ca pe un alt set de date nou să aibă o eroare foarte mare. Practic aceasta nu își mai îmbunătățește în niciun fel abilitatea de a oferi o predicție bună și începe să se axeze prea mult pe detalile dintre date. Acest lucru este des întâlnit la seturile de date mici, pentru că modelul ajunge să țină minte toate caracteristicile setului mic de antrenare, iar în momentul în care se întâlnește cu un set nou (validare), nu o să poată să funcționeze corespunzător. [7]

Pe de altă parte, sub-antrenarea este și ea o dificultate care poate să apară în momentul antrenării. La supra-antrenare, modelul începea să memoreze datele pe care

își efectua antrenarea, în loc să sustragă reguli. În cazul sub-antrenării, modelul nu reușește să capteze anumite reguli între date, prin care să ofere o predicție suficient de aproape de realitate, nu are suficiente exemple pe care să învețe. [7]

În figura 2.3, se creionează o comparație sugestivă între ambele situații distincte, supra- antrenarea și sub-antrenarea, dar include și forma corectă a unui model robust.

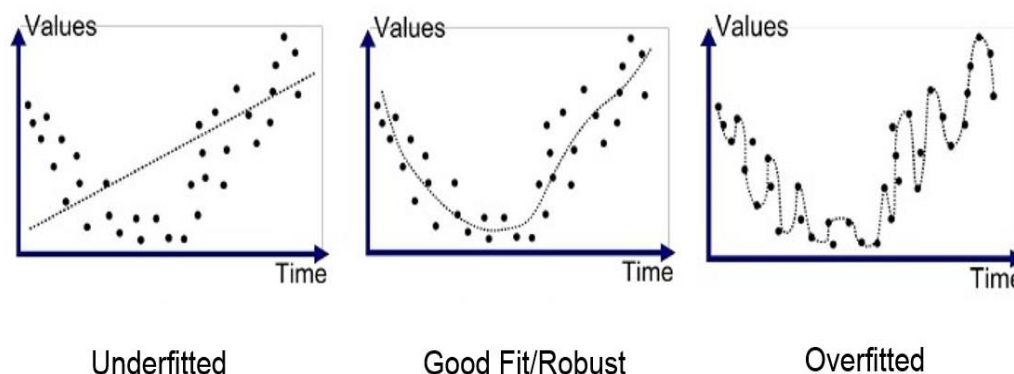


Figura 2.3 Stări ale rețelei

Un alt aspect notabil este numărul de straturi ascunse pe care o să le dețină rețeaua neuronală artificială. Fiecare strat ascuns adăugat o să conțină un număr de neuroni diferit. Cantitatea trebuie să fie descoperită în așa fel încât modelul să ajungă la o performanță apropiată de ideal. Dacă rețeaua are prea puține straturi ascunse, este posibil ca aceasta să nu fie capabilă să construiască suficiente legături cu ajutorul cărora să învețe să pună la dispoziție un rezultat conform realității.[4] Importanța numărului de neuroni pe fiecare strat a fost exemplificată pe modelul realizat în soluționarea problemei, prin grafice reprezentative. În literatura se specifică faptul că în rețeaua neuronală artificială e indicat să fie utilizate un minim de aproximativ 2 straturi ascunse.[11]

Modelul neuronal artificial este alcătuit din mai mulți hiperparametri care pot să fie modificați. Prin modificarea acestora se influențează maniera de funcționare a modelului. Schimbările oferă programatorului posibilitatea să observe cum este afectat modelul creat, reușind să găsească o combinație optimă care să-l ajute să atingă o acuratețe ridicată. Spre exemplu, numărul de straturi ascunse simbolizează și el un hiperparametru al arhitecturii modelului.

În momentul antrenării modelului neuronal, se folosesc diferiți algoritmi de optimizare. Există diferite tipuri de algoritmi de optimizare, câteva exemple sunt următoarele: Adam, RMSProp, SGD.

Algoritmul de optimizare Adam este unul dintre cei mai exploatați algoritmi de optimizare, în probleme care includ învățarea automată. El are mai multe avantaje, printre care și nevoia de memorie mică este un atu. Oferă o recalculare a ratei de învățare pentru fiecare pondere a rețelei. [9]

RMSProp este tot un algoritm care este deseori în multe situații care implică modele neuronale artificiale. Acesta are o formulă de calcul diferită pentru rata de învățare, față

de cea prezentat în cazul algoritmului numit Adam. Metoda de calcul se bazează pe media pătratelor gradientelor. [9]

O altă metodă fezabilă de optimizare este descrisă de algoritmul SGD. El calculează funcția de pierdere pentru un singur eșantion într-un anumit moment specificat. Acesta nu ia în considerare tot setul de date de antrenare pentru a efectua calculul. Este un algoritm care este ales și preferat în multe probleme de lucru cu date.[9]

Fiecare algoritm de optimizare oferă o funcție unică de calculare a ratei de învățare. Persoana care construiește rețeaua neuronală trebuie să se informeze atent și să încerce diferite combinații între algoritmul de optimizare, arhitectura rețelei și hiperparametrii modelului, pentru a putea ajunge la niște concluzii concludente. Pentru a putea observa care combinație conduce spre rezultate cu o eroare mult mai mică sunt necesare teste cât mai variate și observarea modificărilor pe grafic a valorilor.

Fiecărui algoritm de optimizare i se poate atribui o rată de învățare diferită. Aceasta poate juca un rol important în performanțele modelului. Există două cazuri, când rata de învățare este mult prea mică și momentul în care este prea mare. În cazul în care rata este mult prea mare, de fiecare dată minimul local o să fie ignorat, conducând spre rezultate nedorite. O să se observe prezența oscilațiilor, dar și prezența unei grad lent care o să îndrume spre o eroare mică. Pe de altă parte, o rată de învățare foarte mică poate să necesite un număr mare de epoci. Rețeaua nefiind capabilă să acumeleze destule reguli între date, astfel că ea nu reușește să ajungă la o eroare suficient de mică. Performanțele rețelei ar putea să ajungă să fie foarte lente, neputând să îndeplinească scopul final al modelului.[4]

Mai sus s-au specificat asemănările între modul de funcționare a creierului uman și procedeul prin care o rețea neuronală artificială acumulează informații. Pentru a oferi predicții cât mai corecte, algoritmi folosesc toate datele de intrare ca și caracteristici și încearcă să obțină niște reguli de mapare cât mai robuste. Există anumite date de intrare care sunt neliniare sau care oferă un grad de complexitate ridicat pentru maparea lor spre rezultatul dorit. Pentru soluționarea acestei dificultăți au fost create funcțiile de activare. Acestea sunt folosite pentru a limita amplitudinea valorilor de ieșire într-un număr întreg.[11]

Fiecare strat al rețelei neuronale artificiale are propria funcție de activare. Ea poate să difere în funcție de strat sau poate să rămână aceeași. Este vital să se înțeleagă faptul că se poate impune un prag de activare. Cu alte cuvinte, dacă datele de intrare pentru funcția de activare respectă pragul impus, neuronul este considerat activ.[11]

Modul de funcționare într-o rețea neuronală artificială este următorul: se iau intrările și se obține rezultatul sumei lor alături de greutatea lor. Sumei i se alocă o funcție de activare anterior aleasă, generând ieșirea corespunzătoare stratului implicit. Ieșirea este apoi distribuită mai departe spre următorul strat care așteaptă date.[11]

Câteva exemple de funcții de activare utilizate pentru diferite scopuri ar fi: Sigmoid, Tanh, ReLU, Leaky ReLU, ReLU parametrizat, Binary step function, funcția liniară.[11]

### A. Funcția pasului binar

Este cea mai simplă funcție de activare atât ca și implementare în mediul de dezvoltare Python, dar și ca mod de funcționare. Dezavantajul pe care îl are este că nu oferă rezultate bune pentru problemele de clasificare pe mai multe clase. În figura 2.4 se află atât formula matematică a funcției, cât și reprezentarea grafică. [11]

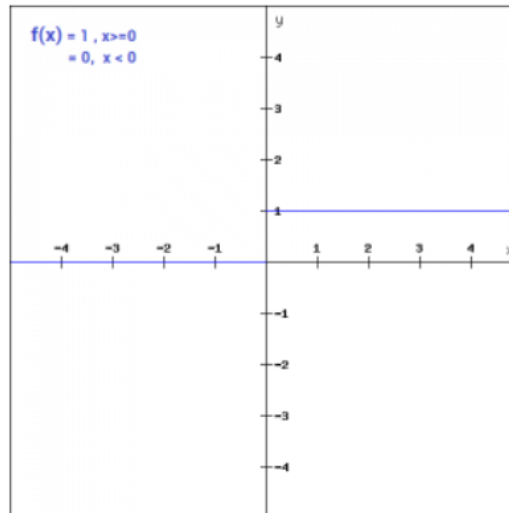


Figura 2.4 Pasul binar

### B. Funcția de activare sigmoid

Este una dintre cele mai utilizate pentru problemele de clasificare. Convertește rezultatul final în valori de 0 și 1. Semnele neuronilor vor fi consistente. În următoarea imagine este o reprezentare explicită a funcției. [11] Are o utilitate mărită în cazul problemelor de clasificare binară.

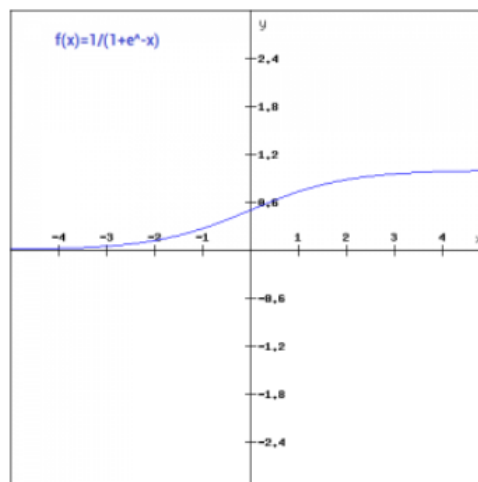


Figura 2.5 Funcția Sigmoid

### C. Funcția liniară

Are o formă foarte ușoară. Aceasta este direct proporțională cu intrarea. Aceasta este ilustrată în figura 2.6. [11]

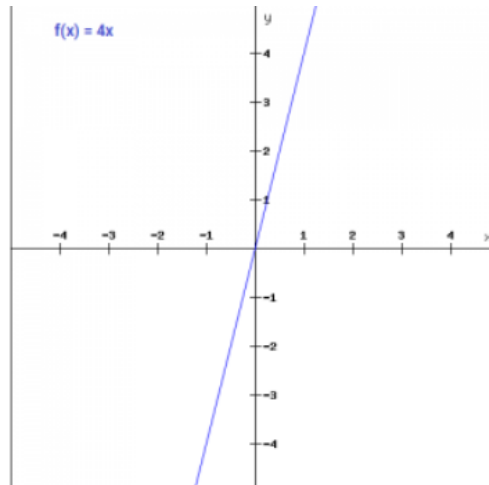


Figura 2.6. Funcția liniară

#### D. Funcția de activare ReLU

Se folosește în general în rețelele neuronale artificiale. Cel mai mare avantaj pe care îl are această funcție de activare este că neuronii nu sunt activați toți în același timp. Acest lucru contribuie foarte mult la creșterea eficienței în producerea de rezultate. O reprezentare grafică a fost atașată mai jos pentru a oferi o înțelegere mai profundă. [11]

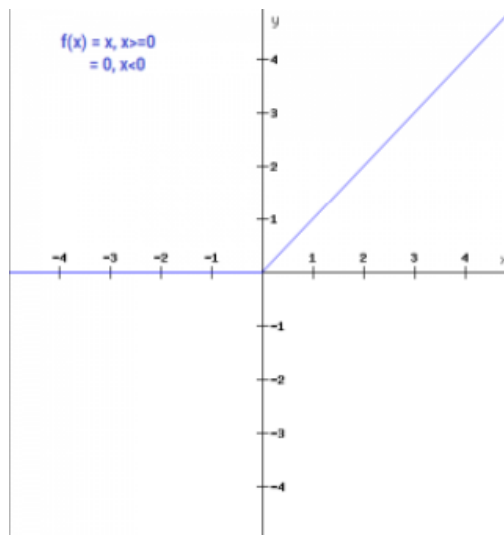


Figura 2.7. Funcția ReLU

#### E. Funcția Leaky ReLU

Este funcția relu construită pentru momentele în care  $x$  deține o valoare negativă, rezultând că o să i se atribue o valoare foarte mică. În momentul în care se folosește ReLU, dacă  $x$  dispunea de o valoare care are ca și caracteristică semnul minus, atunci funcției i

se atribuia valoarea 0. S-a atașat o imagine prin care se poate observa și asemănarea izbitoare cu funcția ReLU. [11]

O comparație descrisă de grafic între funcția de activare ReLU și Leaky ReLU este descrisă prin înclinația dreptei din punctul de origine. Se observă că în cazul celei de a doua funcții se obține un unghi mai mare cu axa absciselor.

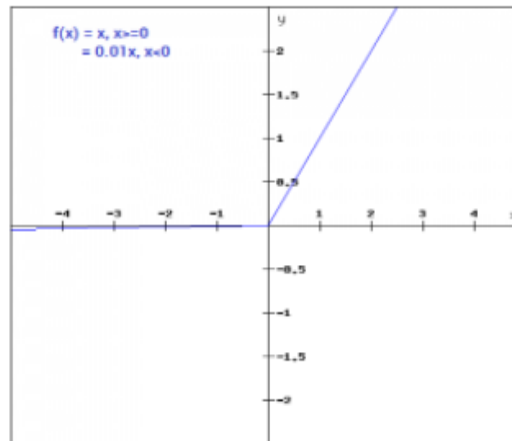


Figura 2.8. Funcția Leaky ReLU

#### F. Funcția Tahn

Diferența majoră între Tahn și Sigmoid, este subliniată prin faptul că această funcție de activare nu menține același semn pentru neuroni. Amplitudinea maximă și minimă a ieșirilor este cuprinsă în intervalul 1 și -1. În figura 2.9 este exemplificată modelul grafic al acestei funcții.

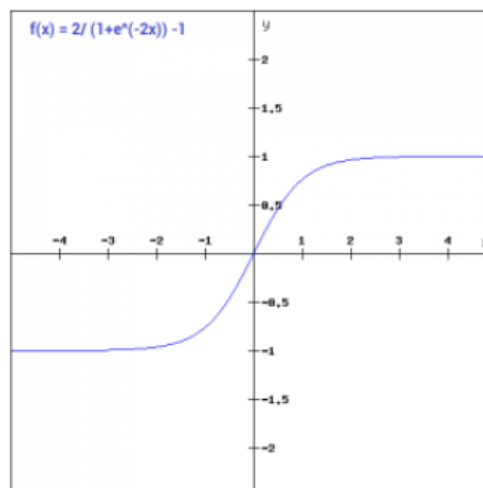


Figura 2.9. Funcția Tahn

Nu există anumite direcții clare și stricte care să impună folosirea anumitor funcții de activare. Acest lucru depinde de datele de intrare, de ce se dorește să se obțină ca și rezultat, dar și de arhitectura rețelei. Ca și numărul de straturi sau numărul de epoci, neuroni de pe straturi, căutarea funcției de activare potrivită necesită o serie de teste



asupra modelului. Totuși, se cunoaște faptul că funcția de activare Sigmoid oferă rezultate foarte bune pentru situațiile în care se dorește o clasificare, dar alături de Tahn, nu este recomandat să se folosească pe straturile din mijlocul rețelei. Ca și alternativă care oferă performanțe foarte bune, pentru straturile ascunse, este ReLu, iar Leaky ReLu în momentul în care există posibilitatea de neuroni morți. Acestea sunt numai niște constatări care au fost obținute în urma mai multor teste. [11] Toate aceste informații susțin faptul că funcția de activare are un rol semnificativ în rezultatele care vor fi înregistrate.

Partea de clasificare se poate împărți în trei categorii distincte: binară, multclasă și multi-etichetă. Cea binară oferă posibilitatea grupării rezultatului grupării în două opțiuni, 1 sau 0. Metoda mai multor clase acordă posibilitatea împărțirii ieșirii în mai mult de două clase, iar metoda mai multor etichete furnizează o sub clasificare a mai multor clase.

Pentru a putea constata dacă rețea oferă niște predicții corecte și apropiate de adevăr, pe lângă partea vizuală în care se crează un grafic cu datele obținute prin predicție și datele reale păstrate pentru partea de testare, se utilizează diferite metrice de măsurare. Dacă modelul poate să furnizeze o acuratețe cât mai mare atunci el poate să fie utilizat în domeniul pe care îl vizează, astfel ar reuși să înlocuiască metodele tradiționale de tratare. În mod firesc, există metrice de măsurare pentru probleme de clasificare, dar și speciale pentru soluționarea aspectelor caracteristice regresiei liniare. Metricile de măsurare oferă o imagine de ansamblu asupra modului de lucru a rețelelor, cât de bine reușesc acestea să funcționeze.

În lucrarea au fost folosite ca și metrice de măsurare a acurateței următoarele: eroarea medie pătratică (MSE), coeficientul de determinare ( $R^2$ ), eroarea medie absolută (MAE). Prin intermediul lor s-a putut urmări evoluția și modul în care anumiți hiperparametri aduc modificări asupra performanței rețelelor neuronale artificiale.

Eroarea medie pătratică face parte din categoria numită media erorilor la pătrat. Din formula care definește acest tip, se observă prezența ridicării la pătrat a diferenței între predicție și realitate. Această clasă este influențată mult mai ușor de valorile numerelor și se folosește în momentul în care se dorește o privire de ansamblu a modului în care evoluează erorile obținute prin predicție pe parcursul testărilor efectuate. Prin utilizarea acestui tip de erori se face o medie între ce se așteaptă să se obțină și există în realitate, valorile ideale. [13] Practic se poate afla cât mai este de înbunătățit modelul pentru a ajunge la momentul în care acesta să poată pune la dispoziție un mod de lucru suficient de robust pentru a putea să fie integrat ca modalitate de rezolvare a diferitelor probleme.

Coeficientul de determinare ( $R^2$ ) are valori cuprinse în intervalul 0 și 1. El subliniază cât de bine are modelul creat, capacitatea de funcționare precisă. Cu cât este mai aproape de 1, cu atât modelul oferă oamenilor, o predicție cu o acuratețe crescută, astfel obținând și o caracteristică convingătoare. [13]

Față de cele două metrice prezentate anterior care fac parte din categoria destinată pătratului diferenței, s-a folosit și o metrică din clasa erorilor absolute. Cea utilizată a fost

eroarea medie absolută. Ea se calculează ca fiind valoarea absolută a sumei pentru diferențele dintre ce e în realitate și ce s-a obținut cu ajutorul predicției. Asupra acestei sume pentru a dobândi valoarea finală a eroarei medie absolute, se aplică raportul  $1/N$ . Prin această înmulțire cu raportul precedent se asigură valoarea corectă a erorii medie absolute. Prezența modului în cadrul determinării sumei, scoate în evidență clasificarea acestui tip de eroare în tiparul erorilor absolute.[13]

În tabelul 1 sunt trecute formulele de la cele trei metrici utilizate în cadrul proiectului. Deosibirile dintre acești indicatori de performanță sunt vizibile în special în modul de calcul. [13]

*Tabel 2.1. Metrici de acuratețe*

Nr crt	Nume	Prescurtare	Formulă
1	Eroare medie pătratică	MSE	$\frac{1}{N} \sum_{n=1}^N (p_n - \hat{p}_n)^2$
2	Coeficient de determinare	$R^2$	$1 - \frac{\sum_{n=1}^N (p_n - \hat{p}_n)^2}{\sum_{n=1}^N (p_n - \bar{p})^2}$
3	Eroarea medie pătratică	MAE	$\frac{1}{N} \sum_{n=1}^N  p_n - \hat{p}_n $

Toate aceste informații au fost studiate în prealabil pentru a putea să se creeze două modele de rețele neuronale artificiale menite să ajute la soluționarea problemelor din domeniul medical, axate pe cancerul din zona mamară.

## 3 Analiză, proiectare, implementare

Proiectul are ca și scop final oferirea de rezultate care au menirea de a ajuta spitalele, mai precis medicii oncologi, prin calcularea unei predicții a ariei, dar și clasificarea caracteristicilor tumorale. Acest lucru a subliniat importanța creării unui model care să pună la dispoziție o eroare cât mai mică, aproape de ideal, prin acest lucru crescând acuratețea de predicție și clasificare.

Capitolul curent înglobează o descriere amănunțită asupra bazei de date care a ajutat la antrenarea rețelelor, mediului de dezvoltare folosit pentru partea de implementare, părți de cod relevante pentru a putea fi exemplificat concis modul de construire a rețelelor neuronale artificiale, dar oferă și o prezentare detaliată existentă în subcapitolul destinat testării.

Prin includerea acestei componente în documentație, se poate asigura o privire de ansamblu a modului de lucru abordat pentru soluționarea problemelor de regresie liniară și de clasificare binară a tumorii mamare. Rețele au putut să fie aduse într-o formă prin care să ofere o funcționare suficient de bună în combinație cu setul de date utilizat ca suport.

### 3.1 Mediu de dezvoltare

În scopul întocmirii acestui proiect a fost necesară alegerea cu atenție a mediului de dezvoltare prielnic implementării, dar și limbajul de programare a influențat tehnica de lucru. Mediu de dezvoltare care a servit ca sprijin în realizarea acestei lucrări și găsirea unui algoritm potrivit, este reprezentat de Pycharm Community Edition 2023. El pune la dispoziție posibilitatea navigării cu o dificultate scăzută prin proiect sau chiar oferă sugestii la ce ar trebui programatorul să folosească în momentul scrierii codului.

Limbajul de programare folosit în contextul acestui proiect, este Python cu versiunea de 3.9. Alegerea acestui program a fost bazat pe motivul că în ultima vreme a reușit să câștige foarte mulți oameni dornici să-l utilizeze pentru simplitatea cu care se poate scrie cod. Acesta oferă și o lizibilitate crescută și odată cu ea și înțelegerea codului devine ușoară. Este foarte ușor de învățat, astfel a fost o alternativă fezabilă în momentul deciderii limbajului de programare. Prin simplitatea codului, Python în combinație cu învățarea automată, face mult mai ușor procesul de rezolvare a problemelor expuse în cadrul acestei lucrări.

Un alt aspect care a contribuit la alegerea limbajului, este dimensiunea largă a bibliotecilor prin care se realizează importul diferitelor funcții. Ele sunt reprezentate și de biblioteci specifice funcțiilor de activare a neuronilor, de creare a straturilor sau de calculare a unor metrice menite să ofere informații referitoare la performanțe.

Prin aceste caracteristici valoroase aduse atât de mediul de dezvoltare, cât și de limbajul de programare ales, s-a putut realiza o implementare facilă pentru programator. Nu este de neglijat, faptul că prin toate proprietățile s-a putut organiza și executa codul, astfel ajungându-se la o rezolvare solidă menită să vină în sprijinul medicilor care se ocupă cu tratarea acestei neoplazii în rândul oamenilor de sex feminin.

## **3.2 Caracteristicile setului de date**

La baza învățării automate se găsesc datele folosite în cadrul algorimilor aleși. Acestea reprezintă informația pe baza căreia algoritmul o să înceapă să observe diferite reguli care ulterior o să ajute la predicția rezultatelor dorite. Atât cantitatea, cât și forma datelor sunt aspecte foarte importante care trebuie să fie tratate înainte de a începe scrierea codului. Primul pas în lucru cu date este înțelegerea aprofundată a bazei de date care urmează să fie utilizată, astfel programatorul putând să aleagă corect segmentarea datelor în date de intrare, respectiv ieșire, dar și tehnici de preprocesare potrivite problemei cu care se confruntă. În general, se aleg seturi de date cât mai voluminoase cu putință, astfel algoritmul să poată să învețe pe mai multe cazuri ce trebuie să prezică.

S-au folosit două seturi de date inițial: Breast Cancer Wisconsin (Diagnostic) [3] și Breast Cancer Wisconsin (Prognostic) [14]. În final s-a utilizat prima opțiune din motivul că cea de a doua pune la dispoziție un set de date mult prea mic. Ea îngloba 198 de instanțe. Datorită numărului scăzut de instanțe pe care le înregistra, baza de date Breast Cancer Wisconsin (Prognostic) nu a putut să ofere destule detalii pentru ca rețeaua neuronală să pună la dispoziție niste erori foarte mici. Pentru a se putea exemplifica și mai bine importanța datelor, au fost adăugate și câteva imagini pentru partea de regresie, în care este evidențiată diferențele oferite de colecția de date și cât de mult influențează modul de învățare a rețelei. Cu un set de date mai mic, eroarea medie pătratică a putut să fie scăzută la o valoare de 0.03, iar cu ajutorul bazei de date mult mai ample s-a putut ajunge la o eroare medie pătratică de 0.00001. Odată cu scăderea erorii, s-a putut observa în schimb o creștere semnificativă a valorii coeficientului de determinare. Parametrul de determinare oferă informații despre cât de bine funcționează algoritmul, cu cât este mai aproape de 1 cu atât modelul lucrează mai bine. Importanța urmăririi metricilor de performanță a reprezentat una dintre cele mai eficiente moduri de urmărire și constatare a eficienței de învățare a modelului. Aceste lucruri sunt detaliate mai amănunțit în subcapitolul care înglobează părțile de testare.

Setul de valori asupra căruia s-au obținut cele mai bune rezultate în lucrarea prezentată, este găsit sub numele de Breast Cancer Wisconsin (Diagnostic), furnizând un set amplu de măsurători. Baza de date a fost descărcată în laptop, dar se putea utiliza anumite linii de cod care realizau importul de pe pagina UI machine learning repository. Un dezavantaj al acestei abordări a fost necesitatea continuă a conexiunii la internet pentru a putea să se apeleze baza de date, astfel s-a preferat descărcarea datelor.

Pentru a putea să se înțeleagă setul de date, a trebuit să se pună accent pe proveniența bazei de date. Înregistrările existente în tabel provin de la imagini digitalizate a unei mase mamare, oferind detalii importante referitoare la celulele care sunt implicate în construcția tumorii. Tabelul este format din 33 de coloane care conțin informații

referitoare la diferite aspecte importante, de exemplu: simetrie, arie, perimetru, puncte concave, textură, diagnostic. Un detaliu fundamental este faptul că nicio coloană nu oferă date lipsă.[3] Absența datelor conducea la un procedeu mult mai complicat de specificare prin cod, astfel existența a tuturor datelor a reprezentat un avatânj major. În consecință, setul de date este suficient de amplu, oferă destul de multe date care pot să fie utilizate ca intrări în algoritm facilitând procesul de învățare. De asemenea, un alt aspect pe care îl conține este că oferă o coloană destinată tipului de diagnostic, dar și una în care se află valori despre aria tumorii. Acest lucru a reprezentat și el un avantaj, deoarece în cadrul acestei lucrări s-a dorit rezolvarea a două posibile probleme, una care să ofere predicția ariei și una care se bazează pe clasificare. Prin intermediul acestei baze de date, ambele situații au putut să fie tratate, neimplicând studiul și căutarea unei noi colecții de date.

Pentru problema de regresie care are ca și scop final predicția cu o eroare cât mai scăzută a ariei tumorii, ieșirea datelor a fost sub formă numerică și nu a presupun nicio modificare de format. Pe de altă parte, pentru problema de clasificare s-a folosit coloana numită 'diagnosis' care conținea litere, B însemnând benignă și M reprezentând malignă. Pentru a putea lucra cu acestea a fost nevoie de utilizarea unei conversii, din text în cifre. Aceste litere au fost convertite în numere, B fiind înlocuit de valoarea 0, respectiv M fiind asociat cifrei 1. Cu ajutorul acestei mapări s-au creat două clase posibile, astfel construindu-se algoritmul de clasificare binară. Aceste modificări au trebuit să aibă loc înainte de momentul începerii creării funcției destinată construcției rețelei.

Rețeaua neuronală artificială (ANN) necesită o divizare riguroasă a cantității. Ea impune o segmentarea a informațiilor pe care o să le utilizeze pentru procesul de antrenare și pentru procesul de testare. Vital este ca datele, atât de intrare cât și de ieșire, pe care se realizează antrenarea, să cuprindă cât mai multe variante care sunt posibil de întâlnit. Modelul devine mai eficient în momentul în care dispune de mai multe date prin care poate să creeze legături, neuronii reușind să elaboreze relații puternice de înțelegere. Validarea este obligatoriu să se realizeze pe date noi, nu pe cele pe care s-a realizat instruirea, astfel se observă comportarea modelului, putând să se demonstreze acuratețea reală într-un mediu necunoscut anterior. Această etapă asigură folosirea algoritmilor în cadrul rezolvării unor probleme din viața cotidiană a oamenilor. Metoda de divizare oferă mai multă credibilitatea asupra rezultatelor obținute și este capabilă să sublinieze diferitele erori pe care modelul ar putea să le întâmpine. În contextul prezentat, s-a utilizat un procent de 80% destinată părții de antrenare și restul de 20% a fost alocate testării.

Preprocesarea datelor a constat și ea în utilizarea unor tehnici de normalizare sau standardizare. Modelul furnizat a oferit rezultate cu o îmbunătățire semnificativă cu ajutorul normalizării. Algoritmii au reușit prin intermediul acestei metode să își mărească performanțele și acuratețea substanțial. S-au oferit niște valori mult mai fiabile, astfel obținându-se o eroare medie pătratică mult mai scăzută și un mod de lucru care oferă o capacitate de predicție ridicată. Acest aspect scoate în evidență importanța și modul de evoluție a tehnicilor folosite pentru obținerea rezultatelor finale, dar și necesitatea înțelegerii în profunzime a datelor cu care se lucrează. Fiecare cerință pentru care este construită rețeaua neuronală artificială, necesită o anumită tehnică specifică de utilizare a seturilor de date.

Pentru obținerea unei eficiențe sporite, primele aspecte de care s-a ținut cont sunt: atenția la alegerea setului de date, segmentarea într-o porțiune pentru antrenare și una pentru validare și utilizarea unor tehnici de preprocesare utile implementării. Aceștia sunt primii pași pe care programatorul îi realizează în momentul în care se hotărăște să utilizeze învățarea automată pentru predicția anumitor aspecte. Datele trebuie să fie pregătite, iar în momentul în care se începe procesul de construire a algoritmului, datele vor facilita un proces de învățare fără erori.

### **3.3 Rețeaua neuronală artificială destinată predicției ariei tumorii mamare**

O proprietate a tumorii care reprezintă o zonă cu puternic interes pentru medicii aflați pe secția de oncologie, se referă la modul de evoluție a ariei tumorii. Mai precis, felul în care o să se producă expansiunea afecțiunii. Acest lucru este foarte important, în momentul în care tumoare are o caracteristică malignă, tratamentul poate să fie influențat de mărimea ei. De exemplu, câteva variante de tratamente care pot să fie influențate de acest aspect sunt: chimioterapie, chirurgie sau radioterapie.

Soluția aleasă pentru soluționarea scopului, a presupus crearea unui model asemănător biologiei creierului uman. S-a utilizat învățarea automată împreună cu algoritmul pentru rețea neuronală artificială.

Primul pas în construirea rețelei a fost reprezentat de pregătirea corectă a colecției de date care urma să joace rolul șablonului după care se vor realiza predicțiile. În figura 3.1. este ilustrată o partea de definire a intrărilor, dar și a ieșirilor. Setul de date conținea 33 de coloane. Pentru a realiza procedeul de segmentare a coloanelor a trebui să se înțeleagă inițial ce reprezintă fiecare valoare din acel tabel, astfel putând să se aleagă corect coloanele care o să servească ca și date de intrare, respectiv pentru ieșire. După o documentare atentă privind fiecare serie de valori, ieșirea a fost programată să fie descrisă de coloana numită „area\_mean”. În ceea ce privește datele de intrare a modelului sunt definite de toate coloanele caracteristice acestui set de date, exceptând coloana de ieșire definită anterior, coloana numită „ID” și s-a mai eliminat și cea care conținea informații referitoare la tipul tumorii, malignă sau benignă. Coloana de ID a fost eliminată, deoarece reprezenta un număr unic pentru fiecare pacient care a luat parte la recoltarea de date, astfel nu aducea un beneficiu în calculul ariei. În cazul în care se dorea păstrarea acestei coloane, trebuia să se specifice faptul că nu este o valoare a unui parametru. Cu ajutorul mențiunii, rețelei neuronale artificiale i se scotea în evidență să nu folosească aceea valoare în procesul de antrenare, ea înțelegând faptul că este numai un identificator. Pentru simplitatea atât a codului, cât și a modelului, s-a optat pentru eliminarea completă a acestei coloane.

De asemenea, s-a recurs și la scoaterea coloanei ce conținea informații referitoare la diagnosticul tumorii, dacă aceasta prezintă caracteristica unei neoplazii maligne sau benigne. Detaliile acestea nu jucau un rol semnificativ în calculul ariei tumorii mamare, astfel neutilizarea acestui tip de componentă din setul de date de intrare, nu a influențat performanța de predicție a suprafeței tumorale.

Pentru a se putea face o segmentare clară și precisă a fost nevoie să se specifice eliminarea coloanei care a fost utilizată pentru ce se dorea să se obțină la ieșire. Prin urmare, dacă nu se utiliza tehnica de eliminare modelul utiliza datele de ieșire și ca valori de intrare, astfel acest lucru conducând la o predicție incorectă. Algoritmul primea ca și intrări, date pe care trebuia să le prezică. Performanțele rețelei erau eronate și nu ofereau o soluție utilă în rezolvarea problemei de regresie liniară. Aceste procedee au avut loc înaintea începerii procesului de învățare.

```
output = dataset[['area_mean']].values
#
# Eliminarea coloanelor 'area_mean', 'id', și 'diagnosis' din input
input = dataset.drop(columns=['area_mean', 'id', 'diagnosis']).values
X_train, X_test, Y_train, Y_test = train_test_split(*arrays: input, output, test_size=0.2, random_state=42)
```

Figura 3.1 Împărțirea setului de date

Un alt aspect vital în utilizarea acestui algoritm specific învățării automate, este momentul specificării procentului de date care este rezervat pentru partea de testare și antrenare. Setul de date chiar dacă conține 569 de valori, nu este considerat suficient de mare. În general se lucrează cu seturi de date care conțin mii de valori diferite, astfel ducând la o învățare profundă a modelului. Cu toate acestea, baza de date folosită în proiect, este găsită ca fiind parte din multe studii și articole științifice existente deja. Această informație a jucat un factor decisiv în alegerea acestui set.

Rezultatele care au presupun o eroare medie pătratică și un coeficient de determinare cât mai bune, au fost obținute în momentul în care setul de date de antrenare ocupa valoare de 80%, iar cel de validare restul de 20%.

Notațiile utilizate în cadrul codului prezentate în figura de mai sus, sunt:

- X\_train semnifică datele de intrare utilizate în cadrul procesului de antrenare a rețelei neuronale artificiale
- Y\_train semnifică datele de ieșire folosite în cadrul procesului de antrenare a rețelei neuronale artificiale
- X\_test semnifică datele de intrare utilizate în cadrul procesului de validare a rețelei neuronale artificiale
- Y\_test semnifică datele de ieșire folosite în cadrul procesului de validare a rețelei neuronale artificiale

Validarea implementată pe un set de date diferit față de cel pe care se efectuează procedeul de învățare a rețelei, conferă mai multă robustețe și credibilitate modului de predicție a modelului.

În continuare, s-a trecut la etapa de preprocesare a datelor. Acest aspect a presupun găsirea de tehnici care se pretează pe datele cu care s-a lucrat. Acest lucru a fost descoperit prin mai multe teste cu diferite tehnici. Modalitatea optimă a fost normalizarea, prin care s-au obținut cele mai bune soluții în modelul de lucru cu datele pentru problema curentă. Tehnica de preprocesare a fost aplicată pentru datele de intrare, dar și pentru datele de ieșire, pentru că ambele seturi de date au fost utilizate sub forma de valori continue și nu reprezintă clase sau etichete. Normalizarea a efectuat anumite transformări prin care s-au convertit toate numerele în valori care aparțin

intervalului 0 și 1. După efectuarea segmentării atente și utilizării corecte a tehnicii de normalizare, datele sunt pregătite să ajute rețeaua să învețe și mai apoi să testeze ce a învățat.

În figura 3.2 sunt prezentate liniile de cod care se ocupă cu procesul de normalizare aplicat valorilor. Utilizarea acestei metode, presupune și importarea bibliotecii specifice. Din biblioteca sklearn s-a importat tipul de sclare dorit spre folosire. Acest import a oferit posibilitatea unei utilizări simple și rapide a tehnicii de normalizare. Din liniile de cod se observă aplicarea procedurii atât pe datele destinate procesului de antrenare, dar și pentru cele care se vor utiliza pentru validare. După realizarea acestui pas, toate datele au aceeași caracteristică și mai important este faptul că o să facă parte din același interval de valori. Prin urmare, această tehnică este des întâlnită în momentul în care setul de date există diferențe considerabile între valorile prezente în cadrul tabelului. Se poate spune că se oferă o unifromizare a valorilor.

```
from sklearn.preprocessing import MinMaxScaler
# Normalizarea caracteristicilor de intrare
sc_X = MinMaxScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
# Normalizarea caracteristicilor de iesire
sc_Y = MinMaxScaler()
Y_train = sc_Y.fit_transform(Y_train)
Y_test = sc_Y.transform(Y_test)
```

Figura 3.2 Normalizarea datelor

Al doilea pas a fost construcția unei funcții în care se definește rețeaua neuronală artificială alături de toate caracteristicile ei, în figura 3.3. sunt ilustrate linile de cod care au servit la rezolvarea sarcinii. S-a început cu crearea unui model gol de tip secvențial, pentru că acesta este cea mai comună și utilizată tehnică întâlnită. Crearea unui model secvențial presupune introducerea straturilor pe rând în interiorul modelului. Un avantaj foarte mare în utilizarea acestui tip, este simplitatea dobândită și modul de lucru optim pentru probleme de regresie și clasificare. După definirea modelului gol s-a început adăugarea straturilor cu ajutorul funcției ,add'.

Perfomanțele optime s-au obținut cu ajutorul a două straturi ascunse care au ca și rol realizarea de legături între intrare și ieșire. Se observă că fiecare strat este caracterizat de un număr de neuroni diferit. Primul stat ascuns are ca și proprietăți un număr de 15 neuroni și aceștia sunt activați prin intermediul funcție de activare ReLU. Al doilea strat ascuns a fost alcătuit dintr-un număr de 8 neuroni asupra cărora s-a folosit tot funcția de activare ReLU, ca și în cazul primului strat.

Un alt aspect care a trebuit să fie specificat este dimensiunea intrării modelului. Din motive prezentate mai sus, adică eliminarea a celor 3 coloane din datele de intrare, dimensiunea acestuia a ajuns să fie descrisă de valoarea numărului 29.

Al doilea strat ascuns conține un număr de 8 neuroni, mult mai puțini decât primul. O asemănare între cele două straturi este funcția de activare folosită. S-a putut observa că modelul furnizează cele mai bune valori în care se utilizează ReLU pentru neuronii care



se află pe straturile din mijlocul rețelei. Aflându-se în interiorul în rețelei, starturile de mijloc au ca și scop creare de legături între neuroni și de a putea conduce de la intrare informațiile spre o predicție cât mai corectă la ieșire.

Ultimul strat adăugat determină stratul de ieșire. Acesta poate să conțină un singur neuron, care reprezintă aria tumorii. El este format dintr-un singur neuron care urmează să fie activat cu ajutorul funcției de activare de formă liniară.

```
model = Sequential()
model.add(Dense(15, activation='relu', input_dim=29))
# model.add(Dropout(0.001))
model.add(Dense(8, activation='relu'))
model.add(Dense(1, activation='linear'))#strat_iesire
# Creare optimizer cu o rată de învățare specificată
optimizer_arie = Adam(learning_rate=0.001)
# Compilarea modelului cu optimizerul definit
model.compile(optimizer=optimizer_arie, loss='mse')
model.summary()
history = model.fit(X_train, Y_train, epochs=200, validation_split=0.2)
```

Figura 3.3 Definirea rețelei

Se poate observa și folosirea algoritmului de optimizare. Pe baza documentării anterioare și pe anumite teste efectuate, s-a hotărât utilizarea algoritmului de optimizare numit Adam cu o rată de învățare de 0.001. Valoarea ratei de învățare pentru care s-a ajuns la cele mai bune soluții este chiar cea cu care algoritmul Adam vine predefinit. Totuși, s-a ales abordarea prin care rata de învățare poate să fie scrisă de programator, din cauza testelor care s-au efectuat pentru a vedea felul în care valoarea ratei v-a influența evoluția predicției, dar și eroarea în cadrul modelului.

În final după specificarea fiecărui strat și a algoritmului de optimizare dorit, a urmat partea de compilare unde se specifică ca valoarea erorii să fie definită de valorile pe care le obține eroarea medie pătratică. Acest lucru ajută la monitorizarea modului de dezvoltare pe care o să-l dețină modelul. Antrenarea modelului este caracterizată de numărul de 200 de epoci și o împărțire de validare cu valoarea de 0.2. Motivul utilizării valori de 0.2 este segmentarea datelor de antrenament în seturi mai mici de antrenare și testare, cu ajutorul lui se monitorizează evoluția modelului și modifică hiperparametri pentru obținerea rezultatelor bune. Inițial s-a folosit valoarea de 0.1, dar urmărind modul de lucru prezentat în diferite articole științifice pentru găsirea unei metode de îmbunătățirea a performanțelor, s-a observat preferarea valorii de 0.2. Acest aspect a scăzut puțin eroarea medie pătratică.

Linile de cod prezentate în figura anterioară, ilustrează felul în care s-a abordat rezolvarea problemei de creare și antrenare a unei rețele neuronale artificiale capabilă să ofere o predicție asupra ariei afecțiunii întâlnite la femei în zona mamară.

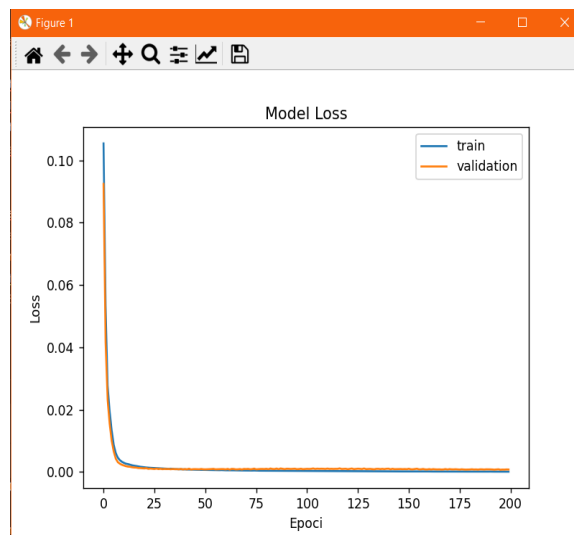
Al treilea punct important în implemnetarea codului, menit să ofere soluții pentru problema de regresie, a fost calculul erorilor. Acest pas are ca și scop dobândirea de credibilitate a modelului. Metricile utilizate în cadrul problemei de regresie au fost: eroarea medie pătratică, eroarea medie absolută și coeficientul de determinare. Cu ajutorul mediului de dezvoltare și a limbajului de programare, acest aspect privind zona de calcul a fost exprimată în trei rânduri de cod ușor de înțeles. Prin apăsarea funcțiilor

specifice, programul utilizează formulele matematice în momentul în care ajunge la partea această, astfel persoana responsabilă cu scrierea codului, nu trebuie să își definească propriile funcții și să se axeze pe matematică. În figura 3.4 este prezentat codul care se ocupă cu obținerea de valori pentru metricile care ajută la urmărirea evoluției performanței modelului. Ambele erori utilizate, eroarea medie pătratică și eroare medie absolută se calculează prin intermediul valorile de ieșire păstrate pentru procesul de testare și valorile oferite prin predicție. O analiză amănunțită a modului de calcul a acestor erori se poate observa în tabelul 2.1 din capitolul doi.

```
# Calcularea erorilor
mse_value = np.mean(np.square(np.subtract(Y_test, yhat)))
mae_value = np.min(np.square(np.subtract(Y_test, yhat)))
r2 = r2_score(Y_test, yhat)
print("Eroarea medie patratice (MSE):", mse_value)
print("Eroarea minima (MAE):", mae_value)
print("Coeficientul de determinare (R²):", r2)
```

Figură 3.4 Calculul metricilor de performanță

În final pentru a fi mult mai ușor de vizualizat și de interpretat varianțele aduse de către parametri în cadrul rețelei, s-au utilizat graficele. S-a importat din librăria matplotlib funcțiile specifice care permit utilizarea și crearea de grafice în cadrul linilor de cod. S-au realizat două grafice, unu pentru ilustrarea suprapunerii dintre datele obținute prin antrenarea modelului și unu pentru observarea graficului erorii pe setul de antrenare comparat cu cea de pe validare.

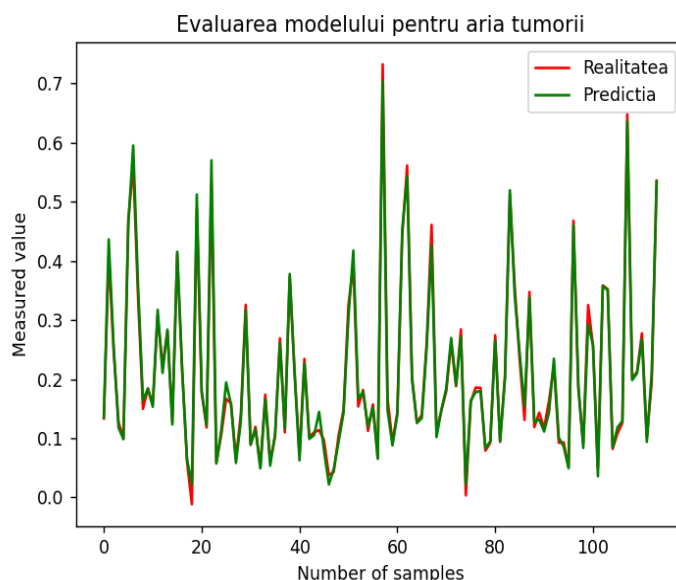


Figură 3.5 Eroarea medie pătratică pe setul de antrenare vs validare

În figura 3.5 se află graficul pentru compararea eroriilor medii pătratice pe setul de antrenare, culoarea albastră și pe cel de testare definit de culoarea roșie. Ambele erori pornesc de la valori sub 1 și ajung să atingă valori apropiate de  $1.1833 \times 10^{-4}$ . În cazul erorii care caracterizează setul de testare se observă că pornește cu o valoare mult mai joasă față de cea de antrenare. Erorile scad suficient de mult și de repede, iar un aspect

important de observat este faptul că nu oferă fluctuații. Acestea pe parcursul a celor 200 de epoci se urmăresc perfect, ajungând și să se suprapună.

În figura 3.6 este prezentată suprapunerea aproape perfectă, dintre datele oferite de rețeaua neuronală artificială, culoarea verde, și datele care fac parte din realitate, care sunt date provenite de la pacienți, culoarea roșie. Cele două grafice se urmăresc aproape perfect, existând foarte multe zone în care suprapunerea e perfectă, astfel că nuanța de roșu nu mai e vizibilă.



Figură 3.6 Grafic cu datele obținute prin predicție comparete cu setul de date reale

Pentru graficele din figurile 3.5 și 3.6 s-au obținut următoarele valorile pentru metricile de performanță a modelului.

- Eroarea medie pătratică (MSE): 0.00016270231905955376
- Eroarea minimă absolută (MAE): 7.406363084523584e-09
- Coeficientul de determinare ( $R^2$ ): 0.9923359776195436

Atât eroarea medie pătratică, cât și eroarea minimă sunt definite de niște numere foarte mici. Ambele sunt cu mult sub valoarea 1, astfel modelul este capabil să învețe, pe baza datelor puse la dispoziție, să ofere predicție corecte, asupra suprafeței prin utilizarea rețelei neuronale artificiale anterior construită. Un punct important de menționat este valoarea coeficientului de determinare care oferă detalii despre cât de bine funcționează modelul. Valoarea primită este foarte aproape de 1, idealul.

În concluzie, rețeaua neuronală creată este suficient de robustă pentru a putea să ofere rezultate cât mai apropiate de realitate. Metricile de performanță susțin această afirmație, iar graficele reușesc să ofere confirmare și să crească credibilitatea modelului de lucru. Prin aceasta analiză se constată că modelul a reușit să îndeplinească cerințele inițiale pentru care a fost creat.

### **3.4 Rețeaua neuronală artificială pentru clasificarea tumorii mamare**

A doua problemă care a făcut parte din obiectivele acestei lucrări, a implicat rezolvarea problemei de clasificarea a tumorii în două clase posibile: malignă sau benignă. Motivația găsirii unui algoritm fiabil pentru soluționarea acestei dificultăți este pentru că tipul tumorii influențează tratamentul care urmează să fie administrat persoanei care prezintă această neoplazie.

Setul de date conține 598 de instanțe provenite de la diferite paciente la care li s-au înregistrat anumite valori importate despre tumoare. Fiecare coloană oferă informații care o să ajute algoritmul să învețe cum să ofere o predicție bună. În cazul de față, tipul tumorii poate să fie de două tipuri, acest aspect fiind specific clasificărilor binare.

Metoda folosită a fost utilizarea tot a unei rețele neuronale capabilă să aibă o putere de acuratețe mărită. Pașii pentru această lucrare au respectat aceași ordine ca în cazul precedent, soluționarea problemei de regresie liniară. În schimb, au apărut unele detalii modificate care vor fi prezentate în continuare.

Partea de preprocesare a datelor a fost un detaliu semnificativ în modul de lucru a modelului. Tehnica care a fost utilizată și în cazul clasificării, a fost normalizarea. Această metodă a oferit cele mai bune rezultate și o comportare adecvată a rețelei. Cu toate acestea, în cazul de față s-a utilizat numai pentru datele care vor ocupa rolul de date de intrare în rețeaua neuronală artificială.

Datele de ieșire au avut o formă mai specială, acestea erau de tip text și au necesitat o convertire specifică. Pentru a putea să se transforme în date numerice, favorizând un mediu mult ușor de lucru, a fost folosită tehnica prin care se ofereau etichete. Coloana care urma să fie folosită pentru ieșire, conținea două tipuri de valori de tip text, mai exact litere, B și M. Cum s-a specificat mai sus, este o problemă de clasificare binară, astfel valorile de pe coloana asupra căreia se dorește o predicție au trebuit să fie transformate în cifra 0 sau 1. Setul de date de ieșire nu a mai trebui să beneficieze de o preprocesare ulterioară, din cauza faptului că acesta conținea valori fixe de 0 sau 1. Normalizarea a efectuat anumite modificări prin care s-au convertit toate numerele în valori care fac parte din intervalul 0 și 1. După efectuarea segmentării atente și utilizării corecte a tehnicii de normalizare, datele sunt pregătite să ajute rețeaua să învețe și mai apoi să testeze ce a învățat.

Datele de intrare au presupus modificări la ce coloane avea incluse. S-au eliminat coloanele care conțineau datele de ieșire, respectiv ID ul pacientului. Numărul de identificare a fiecărei paciente nu prezenta importanță și nu oferea detalii prin care algoritmul să poată să ajungă la o acuratețe mai mare față de cea obținută, astfel eliminarea lui a fost soluția aleasă în cazul rezolvării problemei. Totuși, asupra acestor date a fost necesară aplicarea unei metode de preprocesare, astfel s-a folosit aceeași metodă ca și în cazul rețelei neuronale anterioare.

Împărțirea datelor pentru rezolvarea problemei de clasificare este și el un pas necesar, astfel el a fost fragmentat într-un procent de 80% și 20%. Procentul mai mare a fost folosit pentru datele de intrare și ieșire destinate procesului de antrenare. Datele de validare au ocupat 20% din cantitatea totală de date.

Paul următor a constatat în crearea funcției numită clasificare, exemplificat în imaginea 3.7, având ca și parametri datele de antrenare, dimensiunea intrării, numărul de epoci și împărțirea de validare care a avut aceeași valoare și în cazul regresiei. Dimensiunea intrării s-a mărit, datorată faptului că am eliminat numai coloanele ce erau reprezentate de ID și de diagnosticat, s-a ajuns astfel la numărul de 30, dintr-un total de 33.

Rețeaua neuronală este construită tot din 2 straturi ascunse caracterizate de funcția de activare ReLU și prezentând 15, respectiv 8 neuroni pe strat. Diferență dintre cele două rețele neuronale artificiale construite este vizibilă în cadrul stratului de ieșire asupra căruia de această dată s-a utilizat funcția de activare Sigmoid. Cu toate acestea, pe stratul de ieșire este în continuare prezent un singur neuron.

```
def clasiificare(x2_train, y2_train, input_dim=30, epochs=200, validation_split=0.2):
    model2 = Sequential()
    model2.add(Dense(15, activation='relu', input_dim=input_dim))
    model2.add(Dense(8, activation='relu'))
    model2.add(Dense(1, activation='sigmoid'))
    optimizer_clasificare = Adam(learning_rate=0.001)

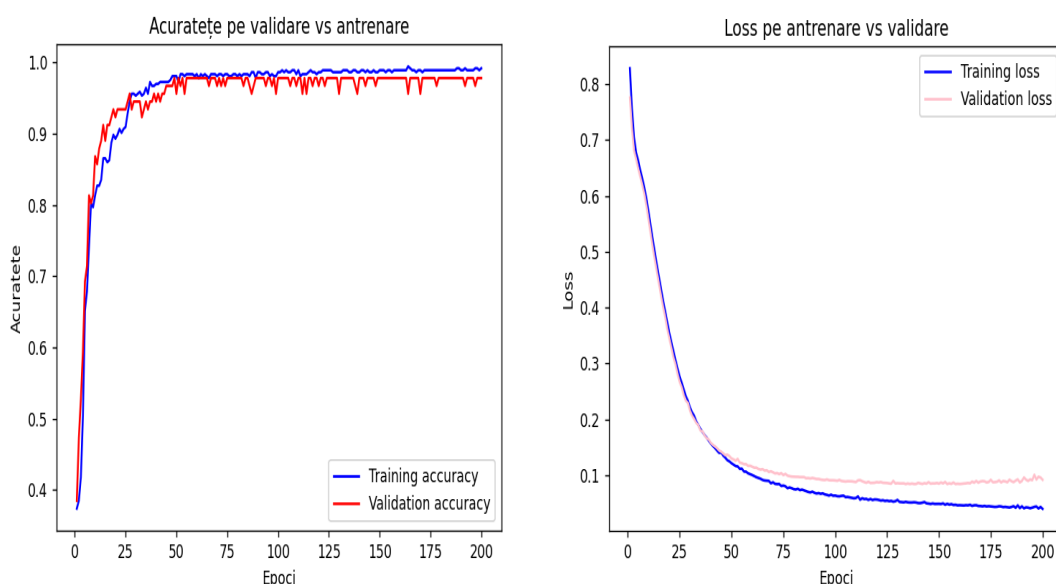
    # Compilarea modelului
    model2.compile(optimizer=optimizer_clasificare, loss='binary_crossentropy', metrics=['accuracy'])
    model2.summary()

    # Antrenarea modelului
    istorie = model2.fit(x2_train, y2_train, epochs=epochs, validation_split=validation_split)
    return model2, istorie

model2, istorie = clasiificare(x2_train, y2_train, input_dim=30, epochs=200, validation_split=0.2)
```

Figură 3.7 Funcția pentru antrenarea rețelei neuronale destinată clasificării tumorii

Algoritm de optimizare este asemănător ca și în cazul regresiei, folosindu-se tot algoritmul Adam cu o rată de învățare de 0.001. Această abordare a oferit capacitatea de a obține cele mai bune performanțe livrate de către modelul implementat.



Figură 3.8 Grafice de comparație pentru acuratețe și erorile pe parcursul antrenării

De asemenea, pentru a putea să se verifice dacă modelul poate să fie folosit pentru scopul pentru care a fost gândit, s-au utilizat anumite metrice destinate testării

performanțelor modelului. În acest caz s-au folosit valoarea coeficientului de determinare, dar și valorile obținute pentru acuratețe.

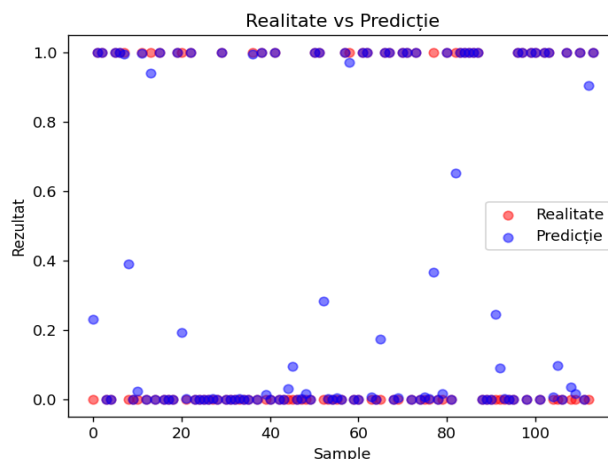
În figura prezentată mai sus, 3.8, se poate observa o comparație între cele două metrici importante: pierderile și acuratețea obținută pe parcursul proiectului. În graficul din partea stângă se ilustrează evoluția acurateței pe parcursul procesului de antrenare a modelului. Culoarea roșie este folosită pentru acuratețea pe validare, respectiv cea albastră este utilizată pentru acuratețea pe antrenare. Se observă faptul că linia albastră este mult mai stabilă, robustă decât cea roșie. O asemănare între cele două este că amândouă prezintă o creștere substanțială până în 25 de epoci, astfel modelul își crește acuratețea foarte mult prin valorile de la epoci. De asemenea, se observă că atât pe antrenare cât și pe testare acestea ajung să depășească valori de 0.95 (95%) și se stabilizează aproximativ în jurul acestei valori.

În ceea ce privește partea de erori, cu ajutorul culorii albastre este descrisă eroarea pe parcursul antrenării, iar cu ajutorul nuanței de roz s-a pus în evidență eroare pe validare. Valorile acestora cunosc o scădere considerabilă de la valoarea de 0.8, ajungând la aproximativ 0.1 pe validare și mai puțin de atât pe partea de antrenare. Ambele grafice nu prezintă fluctuații puternice reușind să se suprapună până în epoca cu numărul 50.

Problema de regresie a putut să fie adusă la niște rezultate mult mai optime, iar acest lucru poate să fie vizibil după valorile pe care le preiau erorile. În cazul acesta de clasificare, metricile de performanță au preluat următoarele valori:

- R2 Score pentru clasificare = 0.9177495389572502
- Acuratețea pe testare pentru clasificare: 0.9824561476707458

Prin analiza acestor valori se demonstrează că modelul are un coeficient ridicat și apropiat de ideal. Acuratețea modelului este de aproximativ 98% astfel acesta tinde spre perfecțiune. Prin aceste valori se poate spune că modelul a fost implementat suficient de bine. O posibilă modificare care ar putea influența această acuratețe este poate lipsa de date, există posibilitatea ca modelul pentru această problemă să aibă nevoie de o cantitate mult mai mare de date pentru o clasificare mai precisă. Legăturile dintre neuroni și regulile pe care și le formează ar putea să prezinte o îmbunătățire în momentul în care se adaugă date. Acest lucru ar putea să influențeze chiar și partea de predicție de date.



Figură 3.9 Rezultatele oferite de ANN în cadrul problemei de clasificare

În figura 3.9 este un grafic care oferă o imagine vizuală a modului în care rețeaua neuronală oferă răspunsuri pentru problema de clasificarea binară a tumorii prezente în zona mamară a femeilor.

Bulinele cu albastru reprezintă datele care provin din predicție, iar cele cu roșu sunt datele reale provenite de la pacientele supuse acestei înregistrări de date. Din cod s-au setat culorile în așa fel încât să se poată observa suprapuneri într-un mod ușor. Momentele în care datele reale se suprapun cu datele obținute, se formează culoarea mov. Creșterea acurateței și mai mult ar implica o intersecție și mai bună.

Se observă o multitudine de cazuri în care datele oferite prin intermediul rețelei neuronale se potrivesc perfect cu cele care erau oferite spre predicție. Prin intermediul unui set de date mult mai voluminos, modelul ar putea să capteze și mai multe informații folositoare.

### **3.5 Teste realizate pentru a ajunge la forma optimă**

Subcapitolul curent înglobează o scurtă istorisire asupra testelor care s-au realizat pe parcursul implementării codului. Procesul s-a bazat foarte mult pe testat și găsirea celor mai buni parametri pentru rețelele neuronale artificiale construite. Cele mai bune rezultate atinse au fost prezentate în subcapitolul care îngloba și modul de implementare. În partea în care se explică liniile de cod, au fost incluse și graficele care pun la dispoziție o reprezentare vizuală și mult mai ușor de urmărit a acurateței modelului.

În continuare, pentru fiecare rețea s-au descris și exemplificat grafic o parte semnificativă a testelor care s-au realizat de-a lungul acestei lucrări. S-au prezentat testele care au adus un aport important în dezvoltare, fiind cele mai relevante și din care s-a putut ajunge la o concluzie sau la o idee care poate să fie premiza unei decizii. Având în vedere complexitatea și numărul ridicat de încercări care au luat parte pentru a se ajunge în punctul final, s-a ales o organizare cât mai simplă și organizată, astfel în fiecare tabel s-au structurat într-un mod ușor de înțeles și urmărit.

Importanța acestui subcapitol este evidentă, pentru a putea ajunge la cel mai bun rezultat s-au încercat diferite combinații. Partea de testare ocupă un loc primordial în cazul oricărui proiect care este menit să ofere soluții pentru probleme provenite din oricare sferă. În cea ce privește o rețea neuronală artificială există o multitudine de cazuri de teste care trebuie realizate pentru a putea asigura o funcționare corectă. Este imposibil să fie încercate toate, dar pe parcursul implementării codului pentru ambele probleme pentru care s-a dorit rezolvare, s-a încercat exploatarea cât mai multor posibilități. De-a lungul acestui subcapitol o să se observe cât de mult influențează fiecare valoare nouă a unui parametru, întreaga serie de performanțe obținute.

Acest subcapitol a fost împărțit la rândul lui în două componente pentru o mai bună structurare. Prima parte este alcătuită dintr-o descriere a testelor realizate în cadrul soluționării problemei de predicție a regresiei. Pentru o mai bună exemplificare s-au utilizat tabele și imagini reprezentative, care pot să sublinieze cu ușurință alegerile făcute în cadrul valorilor alese. A doua parte este o scurtă prezentare a metodelor și valorilor care au fost utilizate pentru a ajunge la forma finală a modelului de rețea neuronală

artificială care se bazează pe rezolvarea problemei de clasificare a tumorii în cele două clase, malignă sau benignă. Asemănător celui alt caz, și în acesta s-au folosit tabele, dar și imagini care au exemplificat cele mai relevante teste care au fost puse în aplicare.

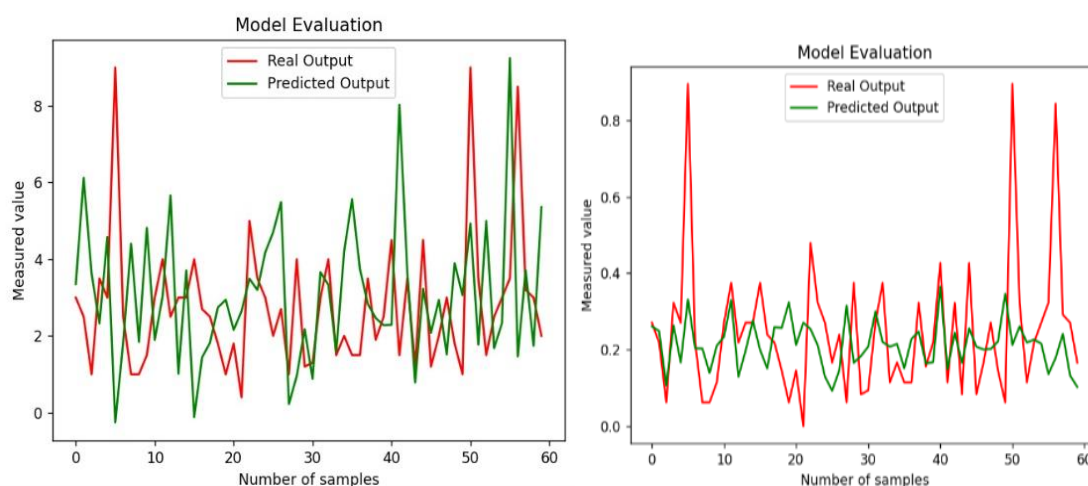
În concluzie, fiecare parte a necesitat o serie de teste specifice pentru problemele pe care le rezolvau. Cu ajutorul lor, se poate ține evidența și decide care variantă oferă cele mai mici erori și respectiv cele mai bune performanțe. Astfel prin acest pas, programatorul se asigură că oferă spre client cele mai bună variantă a muncii sale.

### 3.5.1 Teste realizate pentru rețeauă neuronală artificială destinată predicției ariei

Atingerea unor rezultate care să ofere o credibilitate crescută și o funcționare corectă a modelului pe datele oferite, a presupun o serie de mai multe încercări în baza cărora s-au încercat mai multe variante. S-au adăugat imagini pentru a putea să se facă o exemplificare mult mai ușoară de înțeles și totodată, de urmărit de către programator. Fiecare imagine adăugată conține o explicație succintă despre interpretare și de ce s-a ales să se opteze pentru o altă abordare.

Inițial s-a pornit de la utilizarea unui set mai mic de date, Breast Cancer Wisconsin (Prognostic)[14], notat în tabelul 3.1 ca fiind setul de date 1. Acesta include un set redus de date care a fost insuficient pentru model să poată învăța și să ofere niște performanțe care puteau să fie luate în considerare ca și soluții. Rezultatele cele mai relevante care au fost atinse cu ajutorul acestui set de date sunt reprezentate în tabelul 3.1, ca fiind primele două rânduri. În tabel s-au adăugat și coloanele care conțin valorile pentru coeficientul de determinare și eroarea medie pătratică, acestea au jucat un rol important în urmărirea evoluției modului de lucru a rețelei neuronale.

În figura 3.10, s-a prezentat o imagine de ansamblu a modului în care arătau datele pe care le oferea rețeaua neuronală în comparație cu cele reale. Această imagine conține și graficul pentru standardizare dar și pentru normalizare.



Figură 3.10 Standardizare versus Normalizare pe setul unui de date

S-a observat faptul că prin ajutorul standardizării se creștea amplitudinea datelor, dar nu urmărea aproape deloc datele provenite din realitate. Normalizarea în schimb,



punea la dispoziție o urmărire cu o suprapunere promițătoare în anumite locuri, dar amplitudinea graficului rămânea scăzută. Standardizarea oferea ca și avantaj o urmărire în ceea ce privește partea de amplitudine a datelor, dar se poate observa că nu se suprapuneau aproape deloc cu datele provenite din realitate. Pe acest set de date și implicit cu metodele de preprocesare, s-au încercat diferite modificări în ceea ce privesc numărul de neuroni, numărul de straturi și funcții de activare, dar acestea au fost erorile medii pătratice cele mai mici obținute. Eroarea medie pătratică atinge un număr exagerat de mare, în cazul în care se utiliza standardizare, aproximativ cifra 6. Acesta a fost principalul motiv pentru care standardizarea nu a mai reprezentat o metodă care să fie considerată fezabilă în contextul prezentat.

S-a ales continuarea cu normalizare ca fiind metoda de preprocesare care a oferit cele mai bune rezultate, iar după o serie de modificări asupra parametrilor rețelei care nu influențau cu nimic eroarea medie pătratică, nescăzând sub valoarea de 0.03, s-a încercat mărirea setului de antrenare. Mărirea setului de date a presupus schimbarea procentului care conținea date de antrenare, ajungând să reprezinte 80% din setul de date. Eroarea medie pătratică ajunsese la o valoare de aproximativ 0.01, dar suprapunerile de pe grafic nu ofereau suficientă credibilitate.

Prin urmare, s-a pus problema aflării dacă setul de date nu oferă suficiente detalii sau înregistrări prin care modelul să poată să-și scadă eroarea medie pătratică. Acesta a fost momentul în care s-a hotărât găsirea și utilizarea unui set de date mult mai amplu. Acest set de date mai mic provenea dintr-un studiu care conținea mult mai multe înregistrări de la pacienți, astfel s-a ales cel care oferea cele mai multe înregistrări. În continuare, testele prezentate au folosit setul de date numit Breast Cancer Wisconsin (Diagnosis).[3] Coloana nouă care urma să reprezinte ieșirea dorită, are date mult mai multe și mai exacte. În cazul setului anterior utilizat, datele de ieșire erau numere întregi. Setul nou de date oferă niște date de ieșire mult mai exacte, acestea făcând parte din mulțimea numerelor reale pozitive.

De asemenea, influența mai multor date s-a putut observa în cadrul erorii și la modul de înfățișarea a graficelor. Cu ajutorul unui set de date cu valori multiple, modelul a putut să reușească să ofere o predicție superioară. Pentru a putea să se mărească numărul de date de antrenare, în momentul împărțirii datelor pentru antrenare și validare, s-a preferat varianta în care se folosesc 80% din date pentru procesul destinat învățării rețelei.

Importanța datelor este crucială în buna funcționare a modelului, deoarece la baza oricărui algoritm caracteristic învățării automate, se află regulile care pot să fie identificate între datele furnizate. Acestea sunt primele care intră în interiorul rețelei și care pun la dispoziție detalii despre anumite aspecte importante care o să conducă spre rezultatul final.

În tabelul 3.1 sunt ilustrate organizat toate valorile obținute în cadrul testelor de-a lungul procedurii de implementare a rețelei neuronale, care oferă sprijin în predicția supraprefetei ariei tumorii. Pentru toate testele prezente în tabelul 3.1 s-a utilizat funcțiile de activare în următorul mod: straturi ascunse folosesc ReLU, iar cel de ieșire folosește o funcție liniară.

În prima coloană a tabelului a fost specificată tehnica de preprocesare a datelor aleasă. Au fost trei tehnici care au fost analizate: Standaridizare, Normalizare, RobustScaler. O reprezentare grafică a diferențelor aduse de aceste tehnici în cadrul modelului, se poate observa în imaginea 3.11. Prin analiza acestora s-a văzut felul în care se schimbă datele și modelul odată cu ele. Primele două rânduri sunt calculate pe setul mai mic de date. Acestea au fost adăugate pentru a putea susține ideea necesității unui baze de date suficient de mare.

Testele au fost realizate cu ajutorul optimizatorului Adam sau Rmstop. S-au încercat diferite combinații cu aceste două tipuri de algoritmi de optimizare. De asemenea, s-a abordat și problema schimbării ratei de învățare pe care aceștia o folosesc. Variantele care au fost utilizate în partea de testare pentru cât de repede să învețe au fost următoarele: 0.1, 0.001, 0.0001. De-a lungul fiecărei modificări s-a notat ce influență este adusă asupra modelului. Pentru a putea să fie urmărite cu exactitate performanțele obținute, în tabelul 3.1 au fost adăugate și coloane în care au fost trecute valorile de la eroarea medie pătratică calculată și de la coeficientul de determinare.

Metricile de performanță au acordat o privire pe ansamblu referitoare la capacitatea modelului de a ajuta la predicția ariei tumorii. De asemenea, prin valorile atribuite acestor metrici, se poate atribui modelului o credibilitate suficient de crescută, astfel primește o șansă prin care ar putea să fie o soluție bună pentru soluționarea problemei pentru care a fost construit.

De altfel, un hiperparametru care a jucat un rol important în performanțele obținute de către rețea, a fost reprezentat de numărul de straturi ascunse și de numărul de neuroni de pe acestea. Având dimensiunea intrării de 29, s-a aproximativ că numărul total de neuroni care trebuie să fie în model este între 20-23. Inițial s-a încercat cu un singur strat ascuns, dar nu s-au obținut niște rezultate care să furnizeze niște valori decente de erori. Singura varietate care a avut o eroare medie pătratică mică a fost cea în care s-au pus 22 de neuroni și s-a utilizat algoritmul de optimizare Rmstop. Totuși, după o documentare mai atentă a diferitelor articole, s-a observat că majoritatea susțineau ideea că ar trebui folosite minim 2 straturi. Ambele informații combinate, referitoare la numărul de straturi și la numărul de neuroni s-a ajuns la o împărțire de 15 neuroni pe stratul numărul unu și nouă pe stratul doi. În concluzie, pentru a putea respecta regula cu minim două straturi, s-a continuat testare pentru o rețea care să conțină combinația anterioară menționată.

De altfel, o valoare care a trebuit să fie modificată a fost numărul de epoci. Aspectul acesta a necesitat o atenție deosebită, deoarece au fost multe cazuri în care prin anumite combinații cu o rată de învățare ridicată sau diferiți factori și cu un număr prea mare de epoci, rețeaua intră în supra-antrenare. Numărul de epoci optim asupra cărui s-au obținut cele mai bune grafice este 200. Fiecare valoare poate să ofere diferite schimbări în rețea. În momentul modificării unui număr trebuie din nou început să se testeze cu celelalte. De exemplu, pentru un anumit număr de neuroni și epoci, modelul să se slab antrenat, iar în momentul în care se cresc neuroni, modelul să fie supra-antrenat. Găsirea hiperparametrilor optimi necesită o abordare foarte organizată și diferite teste pentru a putea afla combinația perfectă.

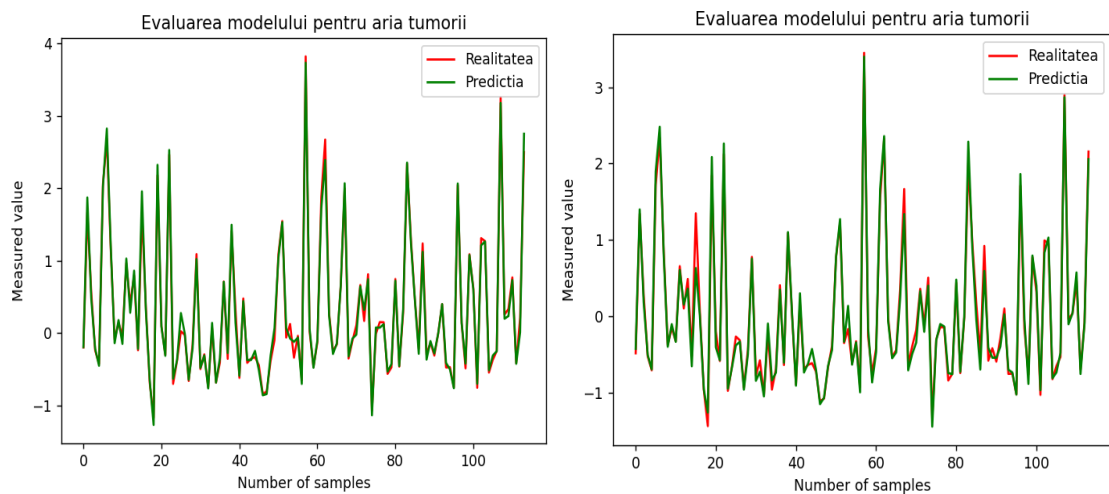
Tabel 3.1 Valori obținute în urma testelor

Prep.	Nr stratur i	Nr neu. strat1	Număr neu strat 2	Opt.	Rată de învățar e	Epoci	MSE	$R^2$
S(set 1)	1	23	-	Adam	0.001	200	~6	-
N(set de date 1)	1	22	-	Adam	0.0001	200	0.03	-
S	1	15	-	Adam	0.001	200	0.011	0.986
S	1	23	-	Adam	0.001	200	0.0179	0.980
S	1	23	-	Adam	0.0001	200	0.0463	0.94
S	2	15	8	Adam	0.0001	200	0.06	0.934
RobustS caller	2	15	8	Adam	0.001	200	0.0116	0.9878
S	2	15	8	Adam	0.001	200	0.0240	0.97
<b>N</b>	<b>2</b>	<b>15</b>	<b>8</b>	<b>Adam</b>	<b>0.001</b>	<b>200</b>	<b>0.0001 6</b>	<b>0.9923</b>
N	2	15	8	Adam	0.0001	200	0.0019	0.90
N- overfitti ng	2	15	8	Adam	0.01	200	0.0005	0.99
N	2	15	8	Rmstop	0.001	200	0.0003	0.98
N	1	22	-	Adam	0.001	200	0.0002	0.989
N	2	15	8	Adam	0.001	150	0.0002	0.98
N	2	15	8	Adam	0.001	300	0.0002	0.993

În tabelul de mai sus , valorile pentru care a fost găsit cel mai bun model au fost îngroșate. Rezultatele acestor teste au condus la o rețea neuronală cu două staturi aflate în mijloc, 15 neuroni pe primul și 8 pe al doilea. Algorimul de învățare care s-a pretat cel mai bine este Adam cu o rată de învățare de 0.001, la un număr de epoci de 200.

În figura 3.11 au fost adăugate două grafice în care se poate observa diferența felului în care se prezintă graficul cu Robust Scaller și cu standardizare. Valorile hiperparametrilor au fost lăsați aceiași ca și în cazul modelului cel mai bun. Erorile medii pătratice au fost următoarele: pentru standardizare s-a obținut 0.02, iar pentru cealaltă metodă 0.01. Eroarea medie pătratică pentru modelul optim este 0.0001, iar tot ce diferă între cele trei varinate este metoda de preprocesare folosită. Fiecare detaliu influențează

semnificativ performanțele obținute, din acest motiv partea de testare a jucat un rol important în procesul de aflare a celui mai optim model.



Figură 3.11 Standardizare versus RobustScaler pentru aceeasi parametrii pentru care s-a obținut cel mai bun model

De asemenea, cu ajutorul graficelor puse la dispoziție se poate remarca faptul că standardizarea urmărește cel mai prost datele reale. În jurul epocii 20 nu se atinge amplitudinea dorită, dar și în continuare se observă o suprapunere mult mai proastă.

S-au efectuat teste și pentru funcțiile de activare care ar trebui să fie utilizate. În tabelul 3.2 se poate observa evoluția rezultatelor pentru modelul optim găsit, în momentul în care se modifică funcțiile care sunt folosite pentru activarea neuronilor de pe straturi. Prin aceste teste s-a pus în evidență și impactul funcțiilor folosite asupra metricilor de performanță.

Tabel 3.2 Performanțe bazate pe funcția de activare folosită

Funcție de activare strat1	Funcție de activare strat 2	Funcție de activare ieșire	MSE	$R^2$
Relu	relu	Relu	0.0001	0.98
Linear	Linear	linear	0.00015	0.98
<b>relu</b>	<b>relu</b>	<b>linear</b>	<b>0.00001</b>	<b>0.997</b>
Leaky relu	Leaky relu	Leaky relu	0.0001	0.992
Leaky relu	Relu	Linear	0.0001	0.995
Relu	Leaky relu	Linear	0.00009	0.995
Leaky relu	Leaky relu	Linear	0.00008	0.996

Setul de combinații pentru funcții de activare pentru care s-a găsit cel mai bun rezultat, a fost îngroșat în tabelul 3.2. S-a decis că cele mai bune valori au fost procurate

prin aplicarea funcției ReLu pe straturi aflate în mijlocul rețelei, iar pe stratul de ieșire o funcție liniară. În modul acesta, eroarea medie pătratică a avut valoare de 0.00001 și s-a atins un coeficient de determinare de 0.997.

În concluzie, cea mai mică eroare și cel mai mare coeficient de determinare, s-au obținut pentru un model cu 2 straturi ascunse a câte 15, respectiv 8 neuroni. Straturile de mijloc folosesc funcția ReLu, iar cel aflat în exterior utilizează funcția liniară. Numărul de epoci optim pentru care modelul a furnizat cel mai bun mod de predicție este 200, iar setul de date fiind împărțit în 80% date de antrenare și 20% date folosite în procesul de validare. Rețeaua neuronală artificială, putând să atingă această performanță, poate să fie luată în calcul ca fiind considerată o soluție suficient de bună pentru a putea să rezolve problema de predicție a suprafeței ariei neoplaziei. De asemenea, aceste performanțe îi oferă un caracter convingător în rândul medicilor pentru a putea să fie pusă în aplicare în spitalele de oncologie. Scopul a fost atins, construirea unei rețele neuronale artificiale care să aibă să dețină capacitatea de a pune la dispoziție o predicție cât mai apropiată de realitate. Prin intermediul, acestui cod, medicii au posibilitatea să observe ritmul de creștere a ariei tumorii.

Prin urmare, se poate afirma faptul că în cazul problemei de predicție, s-a pus în aplicare o implementare riguroasă prin care s-au obținut performanțele așteptate. Partea de testare a venit ca și o completare pentru a susține veridicitatea rezultatelor explicate.

### **3.5.2 Teste realizate pentru rețeaua neuronală artificială destinată clasificării tumorii**

Partea a doua a subcapitolului destinat testării, se axează pe modul de validare a modelului care se ocupă cu clasificarea binară. Se vorbește despre o clasificare binară din motivul posibilității ieșirii de a face parte numai din două clase. Acest lucru a influențat notarea clasei maligne cu 1 și cea benignă cu 0.

În ceea ce privește partea de clasificare a tumorii a presupun o abordare destul de asemănătoare în cea ce privește arhitectura rețelei alese, dar a prezentat modificări în rândul parametrilor modelului.

Setul de date asupra cărui s-au efectuat testele este Breast Cancer Wisconsin (Diagnosis)[3]. Nu s-a mai ales încercarea pe setul cu valori mai puține, deoarece rețeaua neuronală care are ca și scop clasificarea a fost construită ulterior celei pentru arie, nu se dorea utilizarea a două seturi de date diferite.

În tabelul 3.3 au fost adăugate cele mai relevante teste care au condus la niște concluzii. Inițial modelul a conținut un singur strat ascuns și după o documentare mai atentă realizată cu ajutorul diferitelor articole, s-a observat tendința de utilizare a minim 2 straturi. Numărul de neuroni s-a aproximativ la număr cuprins între 21-23 de neuroni prin intermediul regulii de 2/3. Cele mai bune rezultate au fost înregistrate în momentul în care erau utilizați 23 de neuroni în interiorul rețelei.

Se observă schimbarea coloanei destinate valorilor erorii medii pătratice , cu cea pentru acuratețe. Acest fapt se datorează motivului că în cadrul problemelor care au ca și scop împărțirea rezultatului final în diferite clase, nu se utilizează metrica erorii pătratice.

Modul de calcul pentru coeficientul de determinare se poate găsi în tabelul 2.1 aflat în capitoul lucrării numit studiu bibliografic.

*Tabel 3.3 Valori obținute în urma testelor*

Nr. straturi	Nr. neuroni strat1	Nr. neuroni strat2	Opt.	Rată de învățare	Epoci	Acuratețe	$R^2$
<b>2</b>	<b>15</b>	<b>8</b>	<b>Adam</b>	<b>0.001</b>	<b>200</b>	<b>0.982456</b>	<b>0.9412</b>
2	15	8	Adam	0.01	200	0.95	0.86
2	15	8	Adam	0.0001	200	0.94	0.79
1	23	-	Adam	0.001	200	0.98245	0.9249
2	15	8	Adam	0.001	150	0.9736	0.91
2	14	9	Adam	0.001	200	0.982456	0.9182
2	15	10	Adam	0.001	200	0.9736	0.9356
2	15	8	Rmsprop	0.01	200	0.982	0.9225
2	15	8	Adam	0.001	300	0.9736	0.916
2	15	8	Adam	0.0001	300	0.9561	0.8102
2	20	3	Adam	0.001	200	0.9824	0.9262
2	15	8	Rmsprop	0.01	300	0.95	0.8334
2	15	8	Rmsprop	0.0001	200	0.93	0.73
2	15	8	Adam	0.001	250	0.973	0.895

De altfel, cele mai bune rezultate au fost obținute în jurul valorii de 200 de epoci și la un număr de neuroni pe stratul unu de 15, iar pe stratul numărul doi de opt neuroni. În momentul în care se crește numărul de epoci prea mult, rețeau neuronală ajungea să intre în supra-antrenare și deja începea să furnizeze performanțe scăzute. Ca și în cazul precedent s-au făcut teste care au implicat și modificarea algoritmului de optimizare. Adam a fost metoda de optimizare care a pus la dispoziție niște rezultate care au putut să fie luate în considerare.

O altă caracteristică importantă a algoritmilor de optimizare este faptul că pentru fiecare se poate impune o nouă rată de învățare. Cu ajutorul testelor realizate, s-a putut observa că modelul începea să nu mai funcționeze suficient de bine când se creștea rata de învățare. Cea mai bună variantă a fost folosirea algoritmului aflat sub numele de Adam în combinație cu valoarea de 0.001 a vitezei de învățare. Numărul 0.001 este considerat scorul prestabilit pentru acest algoritm de optimizare.

Spre deosebire de circumstanța anterioară, în care s-a creat un tabel (3.1) pentru problema de regresie, aici nu s-a mai adăugat o coloană destinată specificării tipului tehnicii de preprocesare ale pentru datele de intrare. Acest lucru a fost datorat modului de lucru cu datele ieșire, fiind convertite în valori de 0 și 1, deoarece se caută o soluție pentru o problemă de clasificare binară. În consecință, dacă datele de ieșire pot să fie valori numai de 1 sau 0, a trebuit utilizarea unei tehnici care convertea datele de intrare într-un interval de numere între 0 și 1. Drept urmare, pentru îndeplinirea condiției care trebuia să fie respectată s-a utilizat normalizarea. În cazul de față, s-a utilizat numai pentru datele de intrare.

De asemenea, în tabelul 3.3 au fost adăugate și două coloane foarte importante în ceea ce privește urmărirea evoluției modelului. Acestea sunt reprezentate de coloanele care oferă detalii despre acuratețea și coeficientul de determinare a modelului. În probleme de clasificare, acuratețea joacă un rol foarte important în ceea ce privește performanțele puse la dispoziție de către algoritmul folosit. De altfel, din tabelul de mai sus se observă că sunt anumite combinații de hiperparametri care au condus la aceeași acuratețe, dar s-au putut observa diferențe mici la coeficientul de determinare.

În consecință, combinația care a condus la cele mai bune performanțe în cazul rețelei neuronale artificiale destinate problemei de clasificare a tumorii în două clase, a fost îngroșată în interiorul tabelului 3.3. Rețeaua neuronală artificială formată din două straturi ascunse, primul strat având 15 neuroni, iar al doilea strat 8 neuroni, utilizând algoritmul de optimizare Adam cu o rată de învățare de 0.001, s-a obținut valoarea acurateței de 98% și un coeficient de determinare de 0.94. Pierderea care a fost calculată cu aceste valori a fost de 0.0647 pentru datele de testare.

O asemănare între cele două rețele create, implică numărul de neuroni și de epoci pentru care s-a primit cea mai bună acuratețe. De asemenea, se observă că și algoritmul de optimizare favorabil este același și implică aceași valoare pentru rata de învățare. Pe de altă parte, diferența majoră este la modul în care se activează neuronii. În cadrul problemei de clasificare, s-a utilizat funcția de activare sigmoid. Aceasta este cel mai des întâlnită ca metodă de activare în cadrul ultimului strat, pentru problemele care implică o clasificare. Acest detaliu, explică de ce în tabelul 3.4 pentru coloana 3, nu s-a schimbat funcția cu rolul de activare.

În tabelul de mai jos pentru a putea stabili care combinație de funcții de activare oferă cea mai bună acuratețe, s-au adăugat două coloane suplimentare care conțin valoarea acurateței și a coeficientului de determinare. Se observă că valoarea procentului care simbolizează cât de bine se suprapune graficul, nu scade sub 96%.

În ceea ce privește coeficientul de determinare, acesta are o valoare stabilă și ridicată. Acest lucru indică o funcționare corectă a modelului, valoarea acestuia tinde spre 1.

În tabel s-au adăugat diferite combinații între funcțiile de activare caracteristice fiecărui strat din rețeaua neuronală artificială. Se observă că valoarea acurateței rămâne destul de constantă, în schimb coeficientul de determinare prezintă valori distincte. Pentru alegerea celei mai bune combinații de funcții de activare, s-a preferat cea care a oferit o acuratețe crescută, dar și un coeficient de determinare crescut.

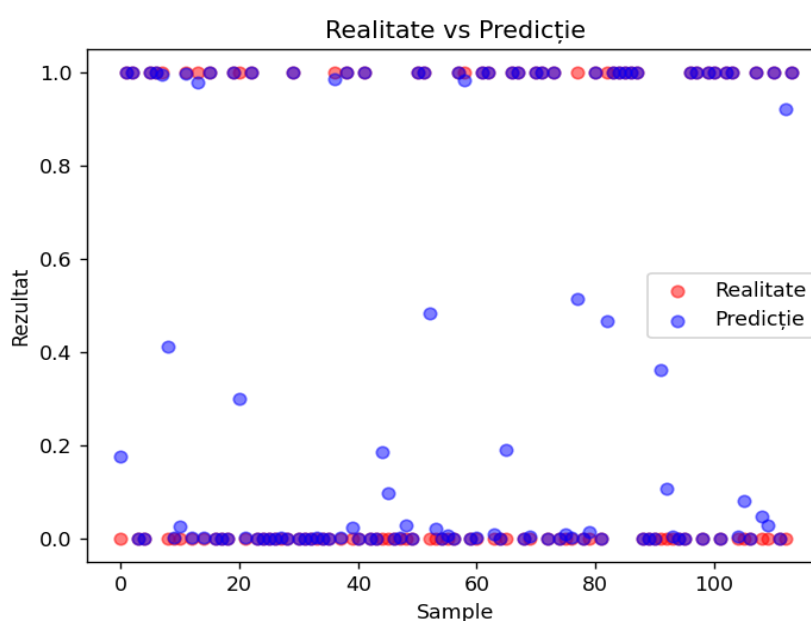
Tabel 3.4 Performanțele obținute bazate pe funcțiile de activare folosite

Funcție activare strat1	Funcție activare strat2	Funcție de activare pe ieșire	Acuratețe	$R^2$
<b>Relu</b>	<b>Relu</b>	<b>Sigmoid</b>	<b>0.9824</b>	<b>0.94</b>
Leaky Relu	Relu	Sigmoid	0.9736	0.92
Sigmoid	Sigmoid	Sigmoid	0.9824	0.9271
Tahn	Tahn	Sigmoid	0.9824	0.9265
Linear	Linear	Sigmoid	0.9824	0.92

Combinăția asupra căreia s-a obținut cea mai bună clasificare a fost îngroșată în tabelul 3.4. Pentru straturile ascunse s-a folosit funcția de activare Relu, iar pentru stratul care se află la ieșire s-a utilizat funcția cu numele sigmoid. Ambele tipuri au fost explicate în capitolul destinat studiului bibliografic.

În acest subcapitol s-au înregistrat numai câteva teste care au pus în evidență concluzia alegerii unei anumite combinații. În realitate, s-au efectuat o multitudine de încercări până să se hotărască care variantă prezintă cel mai bun comportament pentru problema de clasificare.

În figura de mai jos, este ilustrat modul de funcționare a modelului cu toate valorile cu ajutoarea cărora s-au obținut cele mai bune rezultate de-a lungul testelor susținute. Se poate observa că există anumite momente în care nu se oferă o clasificare perfectă și acest lucru se datorează valorii acurateței care nu atinge valoarea ideală.



Figură 3.12 Clasificarea tumorii



Testele care au fost realizate în cadrul acestei probleme au fost mult mai vaste, dar în tabele organizate mai sus s-au adăugat cele mai relevante pentru a pune la dispoziție un mod ușor și eficient de urmărire a modului în care sunt influențate metricele de performanță.

De asemenea, s-a observat o mai bună funcționare în cazul rețelei destinate predicției, pentru acest subiect s-a obținut o eroare mult mai mică și un coeficient de determinare crescut, aproape de ideal. În ceea ce privește rețeaua neuronală ar putea să cunoască posibile dezvoltări, astfel putând să ajungă la o variantă mult mai avansată. Acesta poate să fie considerat un drum care poate să fie exploatat în viitorul dezvoltării implementării.

Capitolul destinat implementării și analizei rețelelor neuronale artificiale asupra căroră s-a realizat aceasta lucrare, este cel mai lung capitol din acesta documentație, conținând detalii despre fiecare lucru care a trebuit să fie folosit de programator pentru a se ajunge la rezultatul final. Aceasta parte conține diferite subcapitole denumite cu nume sugestive pentru o mai bună organizare. S-au prezentat cât mai în detaliu, modulul de lucru cu datele, părți relevante din cod, dar și testele care au urmat. Se poate considera că acest capitol este o însumare a fiecărui pas parcurs în demersul de atingere a scopurilor prezentate în partea de introducere.

În concluzie, ambele rețele neuronale artificiale au beneficiat de o implementare atentă care să respecte anumite reguli. Pentru fiecare s-a încercat găsirea valorilor optime pentru a putea ajunge la soluții capabile să fie introduse ca rezolvare în viața oamenilor. Principalul scop era eliminarea erorii umane din cadrul anumitor situații, din acest motiv s-a urmărit obținerea unei baterii cât mai mici. Partea de teste a fost o secțiune care a necesitat mult timp și o metodă de lucru foarte organizată pentru a nu uita pentru ce valori s-au obținut anumite performanțe. Ambele rețele au putut să ofere rezultatele care erau așteptate și care au fost descrise în subcapitolul lucrării în care erau explicate obiectivele care se doresc să fie atinse.

## 4 Concluzii

Capitolul de față prezintă în detaliu constatările care au fost obținute în urma testelor, dar și rezultatele cele mai bune care au fost obținute în cadrul rețelelor neuronale artificiale dezvoltate de-a lungul acestei lucrări. Obiectivul acestei părți este de a pune la dispoziție o imagine clară și amplă asupra modului de lucru, prezentând concluziile demonstrate anterior în celelalte capitole. Rețelele neuronale artificiale au avut niște scopuri bine definite, acestea oferă suport în cazul afecțiunilor provenite în cadrul infectării cu cancer în zona mamară a sexului feminin.

Rezultatele obținute au fost acordate pentru setul de date Breast Cancer Wisconsin (Diagnosis). Antrenarea modelului și procesul de testare a fost realizate cu ajutorul datelor provenite din baza de date menționată anterior. Aceasta a oferit scheletul de la care s-a pornit în configurarea rețelelor neuronale artificiale.

De asemenea, s-a destinat un subcapitol întreg în care sunt descrise posibile direcții de dezvoltare. Acestea sunt importante și au scopul de a sublinia capacitatea proiectului de dezvoltare în continuare și de a oferi o funcționalitate mult mai mare. Proiectul s-a aflat în stadiul incipient, crearea. Astfel se poate înțelege importanța posibilității de îmbunătățire a proiectului.

### 4.1 Rezultatele obținute

Pe parcursul dezvoltării s-au utilizat diferite combinații pentru a putea să fie testate în procesul de implementare. Aceste teste au putut să ofere o bază puternică și solidă în interpretarea rezultatelor, astfel putând să se ajungă la rezultate favorabile și care pot să fie puse în aplicare în viața cotidiană.

Pentru a putea analiza cele mai bune performanțe și a cunoaște momentul în care au fost atinse, s-au utilizat anumite metrici de performanță care au fost capabile să ofere diferite informații referitoare la modul de funcționare a modelului. Din punctul de vedere menționat, s-a studiat rolul fiecărei metrici pentru a înțelege scopul și ce reprezintă fiecare valoare, astfel s-au ales următorii indicatorii: eroarea medie pătratică, eroarea medie absolută, coeficientul de determinare. Alegerea a fost argumentată în cadrul capitolului destinat studiului bibliografic, dar și în cel pentru partea de implementare. Aceste valori au oferit un fundament esențial în analiza rezultatelor obținute.

De asemenea, în cazul clasificării tumorii în cele două clase în funcție de caracteristicile pe care le prezintă, s-a urmărit și valoarea acurateței pe care rețeaua neuronală o deținea. În acest sens, s-a obținut o acuratețe crescută, care oferă detalii despre cât de bine se separă tumorile maligne de cele benigne. Valoarea care a fost atinsă pentru acuratețe a fost de aproximativ 98%. Coeficientul de determinare în urma testelor asupra parametrilor rețelei neuronale artificiale, a putut să fie crescut până la valoarea de 0.9412. Se poate observa că valoarea acestuia este destul de aproape de numărul 1, care reprezintă capacitatea modelului de a funcționa aproape de ideal.

Totodată, partea destinată predicției ariei tumorii este redată de o problemă de regresie liniară asupra căreia s-a obținut niște performanțe superioare. În cazul acesta s-au urmărit valorile de la eroarea medie pătratică și de la coeficientul de determinare. Scopul este ca eroarea medie pătratică (MSE) să aibă o valoare mai mică de 1 și să se ajungă la un număr cât mai mic. Modelul cu o eroare cât mai scăzută și un coeficient de determinare cât mai crescut, reprezintă cea mai bună variantă de funcționare. În cazul predicției ariei tumorii, eroarea medie cea mai mică a fost 0.00016, iar pentru a doua performanță utilizată a fost de 0.9923. Se poate observa diferența de 0,0077 până când  $R^2$  ajunge la ideal. Suprapunerile dintre graficul cu date oferite de predicției și cel cu datele provenite din realitate sunt putermice. Există numai anumite curbe care nu se urmăresc perfect. Acest lucru accentuează modul optim atins de rețeaua neuronală destinată procesului de predicție a ariei tumorii. Modelul a putut să fie adus la un anumit punct de performanță optim care poate să ofere rețelei un grad crescut de credibilitate și de robustețe. Prin aceste caracteristici, modelul poate să prezinte un interes către posibile direcții noi de dezvoltare. În plus, se poate observa că pentru aceasta situație s-au putut atinge niște performanțe mult mai bune decât în cazul clasificării.

Prin urmare, ambele modele create au putut să fie aduse în punctul în care să rezolve problemele pentru care au fost gândite. Astfel, s-au obținut două rețele neuronale artificiale antrenate pe baza setului de date disponibil pe platforma UI machine learning repository[3]. În ceea ce privește rețeaua care se ocupă cu predicția ariei tumorii, folosindu-se diferite tehnici de îmbunătățire a rețelei s-a putut ajunge la performanțe care pun la dispoziție o predicție foarte apropiate de datele provenite din realitate.

Pe de altă parte, modelul care a fost construit pentru a asigura o clasificare a tumorii, a fost supus anumitor teste pentru a se putea găsi combinația perfectă pentru a obține o acuratețe cât mai mare. Multe din cazurile pe care le oferă realitatea au putut să fie surprinse și cu ajutorul modelului. De asemenea, se observă potențialul mare pe care îl conține aceste rezultate. Modelul oferind și soluții potrivite care vin în sprijinul luptei cu această neoplazie apărută în cazul femeilor în zona mamară.

În concluzie, ambele modele care au avut ca scop simularea realității cât mai bine au putut să își îndeplinească aceasta sarcină, oferind rezultate extraordinare de bune. Erorile sunt de ordin mic, iar valoarea parametrului care măsoară cât de bine funcționează modelele, a ajuns să valoreze un număr apropiat de 1. Astfel, lucrarea începând cu un studiu amănunțit asupra diferitor articole științifice descrise anterior, a putut să pună la dispoziție o descriere în detaliu asupra muncii depuse pentru atingerea rezultatelor optime. Cu siguranță cu ajutorul a unor tehnici mult mai avansate aplicate asupra datelor sau asupra arhitecturii rețelei se pot atinge niște performanțe și mai bune. Acest aspect urmează să fie tratat și îmbunătățit în viitoarele dezvoltări ale proiectului, deoarece este important furnizarea unor soluții cât mai apte care pot să fie descrise de o acuratețe cât mai mare și de o eroare scăzută, aproape neglijabilă. Caracteristicile înregistrate au putut să le ofere credibilitate rețelelor, astfel putându-se să se pună problema de dezvoltare a proiectului în continuare. Proiectul poate să constituie o bază puternică asupra căruia să se poată realiza diferite modificări și actualizări, astfel încât să crească utilitatea modelelor, dar totodată și acuratețea pe care o pune la dispoziție.

## 4.2 Direcții de dezvoltare

Lucrarea prezintă procesul de construire și de învățare pentru un proiect realizat pe parcursul unui an universitar. De-a lungul perioadei de implementare s-au pus în aplicare diferite combinații de tehnici pentru a ajunge la cele mai bune performanțe posibile. Pentru partea de testare, setul de date a fost împărțit într-o parte destinată antrenării și una pentru procesul de învățare. Acest lucru a fost necesar pentru a putea urmări acuratețea de predicție a rețelelor neuronale artificiale, în contextul unor împrejurări noi, fără să cunoască datele deja. O posibilă analiză mai atentă ar presupune testarea rețelei neuronale pe un set complet nou. Cu ajutorul unui set de date complet străin și nou se poate realiza o examinare completă asupra modului de funcționare a modelelor. Acest lucru ar putea să confere o doză în plus de credibilitate și ar oferi o concluzie prin care fără îndoială modelele își ating scopul.

Un alt aspect important ar putea să fie reprezentat și de mărirea setului de date. Mărirea setului de date sau utilizarea uneia mult mai amplu, ar putea să ajute rețelele să învețe mult mai bine. Crearea de legături se bazează foarte mult pe datele care sunt introduse în interiorul rețelei, dar și cantitatea pusă la dispoziție. Volumul datelor o să impacteze performanțele obținute. Totodată, un avantaj adus de setul de date este că poate să creeze mai multe cazuri care pot să fie folosite pentru procesul de învățare, astfel modelul ar fi capabil să surprindă mult mai multe combinații posibile de date provenite de la pacienți.

În plus, lucrarea de față a utilizat date provenite din cadrul unor imagini efectuate asupra zonei mamare a femeilor. Ele au fost specializate numai pentru cancer declanșat în aria sânilor. O direcție care se poate urmări pentru dezvoltarea acestui proiect, ar putea să fie constituit de găsirea unor seturi de date care se concentrează tot pe caracteristicile acestei boli, dar care are proveniență în diferite regiuni ale organismului uman. Prin intermediul acestei adăugări se pot folosi modele într-un mod mult mai generalizat și chiar ar putea să ajute și pacienții care suferă de această neoplazie prezentă în zone distincte. O consecință ar fi mărirea ariei de utilizare a proiectului, astfel el putând să fie adaptat la diferite seturi de date.

De altfel, se poate realiza o analiză mai aprofundată asupra tehnicilor mai avansate pentru arhitectura rețelei, astfel încercându-se să se mărească performanțele în cazul problemei de clasificare. Obținându-se o performanță mai bună, modelul o să fie capabil să producă răspunsuri mult mai apropiate de realitate. Clasificarea corectă a tipului tumorii în combinație cu predicția ariei tumorii pot să constituie un atu important în combaterea acestei afecțiuni. O predicție suficient de aproape de realitate poate să ofere un tratament personalizat pentru fiecare pacientă, crescând șansele de viață considerabil.

De asemenea, o altă posibilă direcție care ar putea să prezinte suficient interes ar putea să fie descris de o interfață. Interfața ar putea să fie făcută cu ajutorul limbajului de programare Python. Prin aceasta să se introducă datele de la pacient, iar pe baza lor, interfața să prezică dacă tumoarea este malignă sau benignă. Interfața ar putea să fie

extinsă și adaptată astfel încât să poată să ofere un mediu prietenos cu utilizatorii care vor fi reprezentați de medicii oncologi. De asemenea, ar putea să fie construită în așa fel încât doctorul să introducă informațiile de la pacient pe care le deține și apoi să-i apară un mesaj prin care să i se comunice o predicție referitoare la modul de dezvoltare a ariei. În plus, se poate integra și partea de clasificare. Ar putea să existe un alt tab al interfeței sau în momentul în care se furnizează datele de intrare a pacientului, să apară un mesaj în care să se ofere predicția referitoare la arie și totodată și tipul tumorii. Prin intermediul acestei aplicații s-ar crea un mediu prietenos și ușor de utilizat în interiorul spitalelor în momentul în care fiecare spital o să dispună de tehnologia necesară.

Prin urmare, este important ca viitoarele dezvoltări să se bazeze pe ce există deja în aceasta lucrare, deoarece proiectul de față poate să servească tip suport pentru o înțelegere mai bună a ce s-a realizat până în momentul actual. Prin continuarea proiectului din punctul în care se află se poate ajunge la creșterea utilității acestuia, dar și la îmbunătățiri importante care vor putea să fie vizibile în cazul performanțelor. Calitatea îngrijirii medicale ar trebui să cunoască un progres semnificativ, impactând la rândul lui experiența pacienților pe parcursul vindecării. Interfața ar putea să aducă un impact pozitiv în integrarea în domeniu medical și ar aduce un avantaj important în crearea unui mediu ușor de utilizat de oamenii din domeniul medicinei.

În concluzie, pe parcursul întregii lucrări s-a ilustrat importanța obținerii unor performanțe cât mai bune, dar și modul în care acestea au fost atinse. Utilitatea proiectului este de nedescris, oferindând o soluție care poate să fie un impact pentru găsirea unui tratament care să aducă o schimbare în probabilitatea de supraviețuire. Întreaga documentație cuprinde și încercările care au ajutat în procesul de atingere a performanțelor optime. În plus, s-a pus la dispoziție o descriere amănunțită asupra soluției alese pentru atingerea scopului. Obiectivele prezentate în partea de introducere, au fost îndeplinite cu succes, astfel se poate concluziona că proiectul a ajuns la forma finală asupra căreia urmând să se caute noi direcții de îmbunătățire.

## 5 Bibliografie

- [1] Yue, Wenbin, Zidong Wang, Hongwei Chen, Annette Payne, and Xiaohui Liu. 2018. "Machine Learning with Applications in Breast Cancer Diagnosis and Prognosis" *Designs* 2, no. 2: 13.
- [2] Patel A. Benign vs Malignant Tumors. *JAMA Oncol.* 2020;6(9):1488.
- [3] Wolberg, William, Mangasarian, Olvi, Street, Nick, and Street, W.. (1995). Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository.
- [4] ABRAHAM, Ajith. Artificial neural networks. Handbook of measuring system design, 2005
- [5] MAHESH, Batta. Machine learning algorithms-a review. International Journal of Science and Research (IJSR).[Internet], 2020, 9.1: 381-386
- [6] Melina Arnold, Eileen Morgan, Harriet Rumgay, Allini Mafra, Deependra Singh, Mathieu Laversanne, Jerome Vignat, Julie R. Gralow, Fatima Cardoso, Sabine Siesling, Isabelle Soerjomataram, Current and future burden of breast cancer: Global statistics for 2020 and 2040
- [7] JABBAR, H.; KHAN, Rafiqul Zaman. Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study). Computer Science, Communication and Instrumentation Devices, 2015, 70.10.3850: 978-981.
- [8] LAVANYA, M.; PARAMESWARI, R. A multiple linear regressions model for crop prediction with adam optimizer and neural network mlraonn. International Journal of Advanced Computer Science and Applications, 2020, 11.4.
- [9] REYAD, Mohamed; SARHAN, Amany M.; ARAFA, Mohammad. A modified Adam algorithm for deep neural network optimization. Neural Computing and Applications, 2023, 35.23: 17095-17112.
- [10] Qifang Bi, Katherine E Goodman, Joshua Kaminsky, Justin Lessler, What is Machine Learning? A Primer for the Epidemiologist, *American Journal of Epidemiology*, Volume 188, Issue 12, December 2019, Pages 2222–2239
- [11] SHARMA, Sagar; SHARMA, Simone; ATHAIYA, Anidhya. Activation functions in neural networks. Towards Data Sci, 2017, 6.12: 310-316.
- [12] ERICKSON, Bradley J.; KITAMURA, Felipe. Magician's corner: 9. Performance metrics for machine learning models. Radiology: Artificial Intelligence, 2021, 3.3: e200126.
- [13] STEURER, Miriam; HILL, Robert J.; PFEIFER, Norbert. Metrics for evaluating the performance of machine learning based automated valuation models. *Journal of Property Research*, 2021, 38.2: 99-129.
- [14] Wolberg, William, Street, W., and Mangasarian, Olvi. (1995). Breast Cancer Wisconsin (Prognostic). UCI Machine Learning Repository