

Tabela Permbledhese

Abstract (english)	2
Abstrakt (shqip)	3
1.1 Hyrje	4
1.1 Studimi 1.....	5
1.2 Studimi 2.....	6
1.3 Studimi 3.....	7
1.3.1 Fazat e projektit te eksperimentit	7
1.4 Studimi 4.....	8
1.5 Studimi 5.....	9
1.6 Studimi 6.....	9
1.7 Objektivi.....	10
1.8 Studime të tjera të rëndësishme.....	11
1.9 Parametrat	12
Kreu II	14
Machine Learning	14
2.1 Machine Learning & IoT	14
2.1.1 Fazat e zhvillimit të ML për kujdesin shëndetësor.	15
2.2 Llojet e teknikave të mësimit të makinerisë	15
2.2.1 Unsupervised Learning.....	15
2.2.2 Supervised Learning	16
2.2.3 Semi Supervised Learning	16
2.3 Aplikimet e Machine Learning	16
2.3.1 Prognoza	16
2.3.2 Diagnoza	17
2.4. Algoritmet e perdorura	18
2.4.1 Algoritme të Reduktimit të Dimensionalitetit (DRA).....	18
2.4.2 Analiza Diskriminuese	19
2.4.3 Regresioni Linear.....	19
2.4.4 Regresioni Logjistik	19
2.4.5. Makinë me Vektor Mbështetjeje.....	19
2.4.6 K Nearest Neighbors(KNN)	20
2.4.7.Pema E Vendimit.....	21
2.4.8. Random Forest Classifier (RFC)	21

2.4.9 Naive Bayes (NB).....	21
2.4.10 Rrjetet Neurale Konvolucionale (CNN).....	21
2.4.11 Rrjeta Neurale Artificiale (ANN).....	22
2.4.12 Mësimi I Makinës Së Përforcimit.....	22
2.4.13 Përpunimi I Gjuhës Natyrore (NLP)	22
2.4.14 Automatika Kognitive	23
KREU III.....	24
H-IOT	24
3.1 H-IOT	24
3.2 ARKITEKTURA E H-IOT	25
3.3 Teknologjitë H-IoT.....	26
KREU IV	28
ECG DIGITALIZATION	28
4.1 KODI	29
4.2 DIGITIZING.....	40
KREU V.....	45
APLIKACIONI.....	45
5.1 Algoritmi KNeighborsClassifier	49
5.1.2 Matrica e Konfuzionit	51
5.2 Algoritmi RandomForestClassifier	51
REFERENCAT	55

Abstract (english)

With the global population exceeding 7 billion, the number of deaths from various diseases is increasing. According to the WHO report, 17.9 million people die each year from cardiovascular diseases, accounting for 31% of all global deaths. The lack of specialists can be partly separated by sharing the information of special people, specialists and artificial intelligence management techniques in the cloud to provide information at an early stage, which can increase the ability of life and others. reduce the risk of death. In this thesis we discuss the operation and design of an IoT (Internet of Things).

In this thesis, we present an Internet of Things (IoT) chain and ElectroCardioGram (EKG) sensors within the context of eHealth. The scheme aims to facilitate timely monitoring of needs and care for patients with cardiovascular (CV) disease, thus increasing care for such patients.

Care plays a role in our daily lives. Diseases can be diagonalized and prevented in the right early stages.

Artificial intelligence can be an organizer to take from points of reference and information, without doing so explicitly. Instead of commemorating the code, supporting data on calculation conventions and reasoning is provided in the data. Artificial intelligence is drilled into web view, spam analysis, ad position, stock exchange, etc.

Abstrakt (shqip)

Me popullsinë globale që i kalon 7 miliardë, numri i vdekjeve nga sëmundje të ndryshme po rritet. Sipas raportit të OBSH-së, 17.9 milionë njerëz vdesin çdo vit nga sëmundjet kardiovaskulare, që përbën 31% të të gjitha vdekjeve globale. Mungesa e specialistëve mund të ndahet pjesërisht duke shkëmbyer informacionin e njerëzve të veçantë, specialistët dhe teknikat e menaxhimit të inteligjencës artificiale në cloud për të ofruar informacion në një fazë të hershme, gjë që mund të rrisë aftësinë e jetës dhe të tjerëve. zvogëlojnë rrezikun e vdekjes. Në këtë tezë diskutojmë funksionimin dhe dizajnimin e një sistemi monitorimi dhe matjeje të lidhur me IoT (Internet of Things) me Arduino.

Në Shqipëri janë të regjistruar rreth 80 000 persona me sëmundje të zemrës. Këtu nuk llogariten rreth 250 000 persona me hipertension (ISKSH). Operacionet by-pass për rivendosjen e qarkullimit të gjakut në muskulin e zemrës, të kryera nga QSUT-ja, pothuaj janë 10-fishuar në 15 vitet e fundit. Numri i tyre nga rreth 30 në vitin 2000, sot është pothuaj 300 në vit (QSUT).[1]

Në këtë tezë, ne paraqesim një skemë të sistemit të kujdesit shëndetësor të Internetit të Gjërave (IoT) duke përfshirë sensorët ElectroCardioGram (EKG) brenda kontekstit të Shëndetit elektronik. Skema synon të lehtësojë monitorimin në kohë të nevojave dhe kujdesin për pacientët me sëmundje kardiovaskulare (CV), duke rritur kështu kujdesin për pacientë të tillë.

Kujdesi luan një rol në jetën tonë të përditshme. Sëmundjet mund të diagonalizohen dhe parandalohen në fazat e duhura të hershme.

Inteligjenca artificiale është dega kryesore për t'u marrë nga pikat e referencës dhe informacioni për ta bërë këtë në mënyrë sa më të qartë dhe të plotë. Qëllimet e inteligjencës artificiale përfshijnë të mësuarit , arsyetimin dhe perceptimin.

1.1 Hyrje

Interneti i Gjërave (IoT) ka qenë subjekti kryesor në komunitetin e teknologjisë së kujdesit shëndetësor gjatë viteve të fundit. Fusha e kujdesit shëndetësor ka një rëndësi të madhe praktike dhe IoT hap një spektër të gjerë mundësish për ta përmirësuar atë. Pajisje dhe sensorë të shumtë bashkëkohorë mjekësorë mund të lidhen përmes rrjeteve të ndryshme, gjë që siguron akses në informacione të rëndësishme për gjendjen e pacientëve.

Ky informacion mund të përdoret më pas për qëllime të shumta, si për shembull monitorimi i pacientëve nga distanca, parashikimi i sëmundjes dhe rikuperimi përmes një pasqyre më të madhe të simptomave dhe përgjithësisht përmirësimi i procesit të diagnostikimit dhe trajtimit nëpërmjet rritjes së automatizimit dhe transportueshmërisë.

Përdorimi i sistemeve të bazuara në ML ka përparësi të shumta. Ata mund të trajnohen duke përdorur vëllime të mëdha të dhënash, të quajtura si të dhëna trajnimi, dhe më pas, nëpërmjet përfundimit induktiv, ata mund të ndihmojnë praktikën klinike në vlerësimin e rrezikut dhe hartimin e trajtimit. Këto sisteme mund të zvogëlojnë gabimin duke eliminuar elementët njerëzorë nga sistemi dhe mund të kryejnë punë të përsëritura, duke përmirësuar kështu efikasitetin në krahasim me përpjekjet manuale.

Kontributet kryesore të kësaj pune janë renditur më poshtë:

- a) Është dhënë një përmbledhje e algoritmeve të ndryshme të spikatura ML me fokus të veçantë në aplikimet e tyre dhe rastet e përdorimit në industrinë e kujdesit shëndetësor-IoT.
- b) Janë paraqitur studime të detajuara të aplikimeve të rëndësishme të këtyre algoritmeve në diagnostikimin e pacientëve me sëmundje të zakonshme si çrregullime kardiovaskulare.
- c) Është anketuar roli i algoritmeve ML me arkitekturën IoT në parashikimin e fazave të sëmundjeve të zemrës.
- d) Janë studiuar teknologjitë futuriste të bazuara në ML-IoT që premtojnë hapa të mëdhenj në bërjen e sistemeve të monitorimit shëndetësor më të aksesueshëm dhe efikas, si dhe teknologji që përmirëson procesin e përgjithshëm të marrjes së kujdesit shëndetësor.

1.1 Studimi 1

Algoritmet ML janë të dobishme për parashikimin e fatkeqësive dhe për të ndihmuar në detyrat e menaxhimit të fatkeqësive, të tilla si përcaktimi i rrugëve të evakuimit të turmës, analizimi i postimeve në mediat sociale dhe trajtimi i situatës pas fatkeqësisë. Algoritmet ML gjejnë aplikim të madh edhe në skenarët e menaxhimit të pandemisë, të tilla si parashikimi i pandemive, monitorimi i përhapjes së pandemisë, diagnostikimi i sëmundjeve, etj.

Vitet e fundit kanë qenë dëshmitarë të adoptimit të gjerë të teknikave të mësimit të makinës (ML)/të mësuarit të thellë (DL) për shkak të performancës së tyre superiore për një sërë aplikacionesh të kujdesit shëndetësor duke filluar nga parashikimi i arrestit kardiak nga sinjalet njëdimensionale të zemrës deri te diagnoza me ndihmën e kompjuterit (CADx).) duke përdorur imazhe mjekësore shumëdimensionale. Pavarësisht performancës mbresëlënëse të ML/DL, ka ende dyshime të vazhdueshme në lidhje me qëndrueshmërinë e ML/DL në mjediset e kujdesit shëndetësor (i cili tradicionalisht konsiderohet mjaft sfidues për shkak të një morie çështjesh të sigurisë dhe privatisë të përfshira), veçanërisht në dritën e rezultateve të fundit që kanë tregoi se ML/DL janë të prekshme ndaj sulmeve kundërshtare.

Në ditët e sotme, sëmundjet e zemrës janë shkaku kryesor i vdekjeve në mbarë botën. Parashikimi i sëmundjeve të zemrës është një detyrë komplekse pasi kërkon përvojë së bashku me njohuri të avancuara. Teknologjia e Internetit të Gjërave (IoT) është miratuar kohët e fundit në sistemet e kujdesit shëndetësor për të mbledhur vlerat e sensorëve për diagnostikimin dhe parashikimin e sëmundjeve të zemrës. Shumë studiues janë fokusuar në diagnostikimin e sëmundjeve të zemrës, megjithatë saktësia e rezultateve të diagnozës është e ulët. [2]

Megjithatë, integrimi i IoT në sistemet e kujdesit shëndetësor ka treguar një mënyrë të jashtëzakonshme për të monitoruar shëndetin e pacientëve dhe për të diagnostikuar anomalitë. Sensorët më të përdorur për EKG janë presioni e gjakut, puls i zemrës dhe temperatura e trupit.

Sinjalet e EKG përfaqësojnë aktivitetin elektrik të zemrës në qetësi. Mund të përdoret për të nxjerrë konkluzione rreth ritmit dhe pulsit të zemrës dhe mund të jetë i dobishëm për diagnostikimin e zmadhimit të zemrës për shkak të presionit të lartë të gjakut, rritjes së rrahjeve të zemrës dhe aritmise ose sulmeve në zemër. Figura 1 tregon disa nga sensorët më të përdorur në lidhje me diagnozën kardiovaskulare.

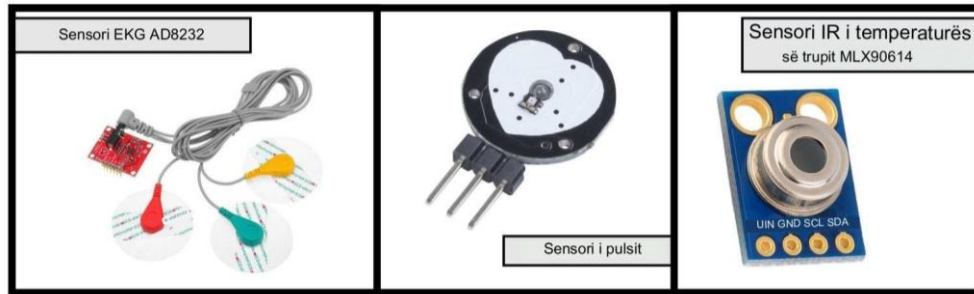


Figura 1. Sensorët më të përdorur për diagnozën kardiovaskulare

1.2 Studimi 2

Për të adresuar këtë çështje, propozohet një kornizë IoT për të vlerësuar më saktë sëmundjet e zemrës duke përdorur një Rrjet Neural Convolutional Modified Deep Convolutional (MDCNN) [2]. Ora inteligjente dhe pajisja e monitorit të zemrës që i është bashkangjitur pacientit monitoron presionin e gjakut dhe elektrokardiogramin (EKG). MDCNN përdoret për klasifikimin e të dhënave të sensorit të marrë në normale dhe jonormale.

Performanca e sistemit analizohet duke krahasuar MDCNN-in e propozuar me rrjetet nervore ekzistuese të mësimit të thellë dhe regresionin logjistik. Rezultatet tregojnë se sistemi i propozuar i parashikimit të sëmundjeve të zemrës me bazë MDCNN funksionon më mirë se metodat e tjera. Metoda e propozuar tregon se për numrin maksimal të regjistrimeve, MDCNN arrin një saktësi prej 98.2 e cila është më e mirë se klasifikuesit ekzistues.

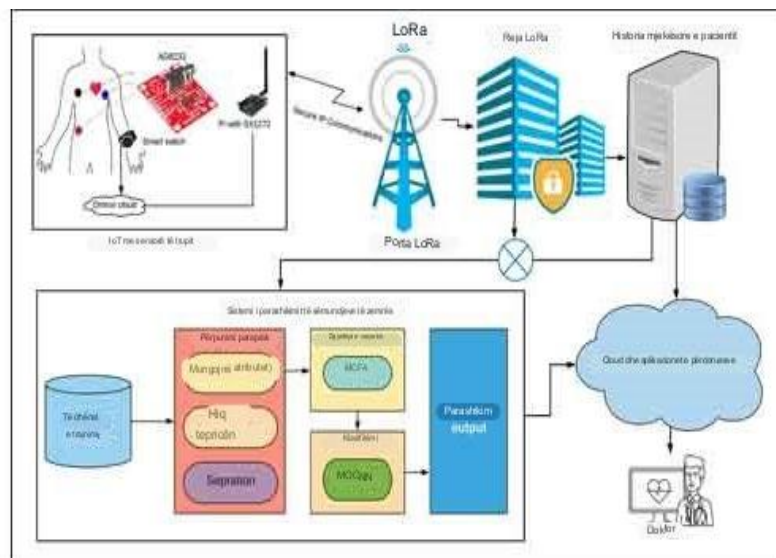


Fig.2. Kuadri për Parashikimin e Sëmundjeve të Zemrës bazuar në Klasifikuesin MDCNN[2].

1.3 Studimi 3

Diagnostikimi I sëmundjeve të zemrës në kohë reale

Gupta. [3] propozoi përdorimin e një modeli të bazuar në ML për të diagnostikuar sëmundjet e zemrës duke monitoruar disa parametra në kohë reale, duke përdorur teknologjinë IoT të veshur. Qëllimi i studimit ishte përdorimi i monitorimit të EKG-së, pulsit dhe temperaturës në kohë reale si të dhëna për një model parashikimi të trajnuar për të parashikuar nëse përdoruesi është në rrezik për ndonjë sëmundje të zemrës ose aritmi.

1.3.1 Fazat e projektit të eksperimentit

Projekti krye në tre faza:

- parapërpunimi i të dhënave
- faza e trajnimit të modelit
- dhe prototipi për parashikimin e drejtpërdrejtë.

1. Në fazën e parë, grupi i të dhënave elektronike të të dhënave shëndetësore (EHR) u konsolidua dhe u izoluan parametrat e konsideruar të rëndësishëm për parashikuesin. Vlerat që mungonin në grupin e të dhënave u zëvendësuan me vlerën e unazës që ndodh më shpesh (dmth. modaliteti) për çdo veçori. Një korrelacion vlerësimi është bërë për të përcaktuar parametrat e ndryshëm duke ndjekur tendenca shumë të ngjashme. U hoqën tendenca të tilla që nuk kontribuan në mësimin e modelit dhe thjesht do të shtonin kompleksitetin e programit (zvogëlimi i dimensioneve).
2. Në fazën e dytë, grupi i të dhënave të përpunuar paraprakisht u nda në grupe trajnimi dhe testimi, të ndjekura nga trajnimi model mbi bazën e të dhënave të trajnimit. Algoritme të ndryshme klasifikimi, përkatësisht KNN, SVM, NB, pema e vendimit dhe pylli i rastësishëm u vlerësuan për saktësinë, shkallën e goditjes, etj.
3. Faza e tretë përfshin zhvillimin e prototipit të harduerit për mbledhjen e të dhënave dhe lidhjen me cloud, dhe më e rëndësishmja, parashikimin në kohë reale. Duke përdorur bordin e zhvillimit të mbështetur nga ESP8266 WiFi, sensorin EKG AD8232, sensorin e pulsit HW01 dhe sensorin e temperaturës MAX30205, nyja u lidh me një laptop që ushqente të dhëna të vazhdueshme në modelin e trajnuar, së bashku me të dhëna të tjera statike si mosha, gjinia dhe nivelet e sheqerit.

Pas kësaj, një laptop përdori modelin duke dhënë një gjykim të saktë 88.9% duke përdorur KNN për të identifikuar nëse pacienti ishte në rrezik për probleme me zemrën. Edhe pse kjo ishte një metodë mjaft e thjeshtë, duhet të bëhen më shumë kërkime për të përmirësuar saktësinë. Prandaj,

kjo teknologji duhet të zhvillohet më tej me trajnim më të mirë me kalimin e kohës dhe instrumente më të mira për një grup të dhënash më të gjerë në kohë reale.

Taÿtan [4] raportoi studime të ngjashme duke përdorur vetëm një qasje monitorimi në kohë reale ndërsa Davis *et al.* [5] gjithashtu llogaritet për të dhënat e kaluara shëndetësore të pacientit.

Khan [6] propozoi një sistem të kujdesit shëndetësor IoT për vlerësimin e sëmundjeve të zemrës duke përdorur CNN të modifikuar të thellë (MDCNN). Të dhënat e përdorura për trajnim dhe testim u mbledhën nga Framingham, depoja e UCI ML dhe të dhënat e shëndetit publik. Një orë inteligjente dhe një pajisje monitoruese e zemrës u përdorën për të mbledhur informacione rreth presionit të gjakut dhe EKG-së, të cilat më pas u transmetuan në server duke përdorur LoRa.

1.4 Studimi 4

MDCNN

Bazuar në këtë informacion të transmetuar, modeli MDCNN klasifikoi pacientin si me një gjendje normale ose anormale të zemrës dhe njoftoi mjekun në rast të një anomalie.

Kjo u krye gjithashtu në tre faza:

- para-përpunimi,
- përzgjedhja e veçorive
- dhe klasifikimi.

*Faza e parë përbëhet nga shtimi i attributeve që mungojnë bazuar në moshën e pacientit, presionin e gjakut dhe kolesterolin, heqjen e tepricave dhe ndarjen e pacientëve në bazë të llojit të dhimbjes së gjoksit që kishin: anginë tipike, anginë atipike, dhimbje jo anginale dhe asimptomatike. Karakteristikat më të rëndësishme për vlerësimin e sëmundjeve të zemrës u zgjodhën duke përdorur algoritmin e optimizimit të sepjeve të bazuara në harta (MCFA) [7]. Së fundi, për fazën e klasifikimit, vlerat e peshës që do të përdoren në model u optimizuan duke përdorur algoritmin përshtatës të optimizimit të tufës së elefantëve. Për prototipin aktual, një Omron HeartGuid-bp8000m u përdor për të matur presionin e gjakut në pushim. Megjithatë, për shkak të mungesës së pajisjeve të veshura për nivelet e kolesterolit dhe glukozës në serum, pseudo numra u krijuan në një gamë fikse. Së fundi, një sensor AD8232 u përdor për matjen e të dhënave të EKG-së. Një Raspberry Pi u përdor për ndërfaqe, i cili dërgoi të dhënat në cloud duke përdorur një transmetues SX1272 900 MHz LoRa. Rezultatet nga ky studim janë dhënë më poshtë në Tabelën 1.

Grupi I të dhënave	Të dhënat	Karakteristikat totale	Vecoritë e zgjedhura	Saktësia
UCI	303	16	7	93.3%
Inkaudrimi	4000	16	7	98.2%
Shëndeti publik	1025	14	8	97.6%
Të dhënat e sensorit	900	16	6	96.3%

Tabela 1. Rezultatet e marra nga studimi per MDCNN[4].

1.5 Studimi 5

Diagnoza e EKG-s

Modeli i zhvilluar mund të përdoret me sensorë të veshur për diagnozën e EKG-së, të cilat lejojnë monitorim të vazhdueshëm gjatë gjithë orës. Sistemi i propozuar kishte 4 hapa:

- zbulimin e rrahjeve të zemrës,
- segmentimin,
- nxjerrjen e veçorive
- dhe klasifikimin. Baza e të dhënave MIT-BIH për aritmitë[8].

1.6 Studimi 6

Baza e të dhënave MIT-BIH

Baza e të dhënave MIT-BIH për aritmitë u përdor për zhvillimin e klasifikuesit. U përdor transformimi i valëzuar diskrete për nxjerrjen e veçorive ndërsa klasifikimi është bërë duke përdorur SVM. Algoritmi më vonë u kodua dhe u zbatua për bordin Galileo IoT të Intel. Sinjali i EKG-së u lexua dhe u digjitalizua me një shpejtësi kampionimi prej 360 mostrash në sekondë ndërsa analiza u bë për çdo 3000 mostra. Saktësia mesatare ishte 97%. Wang *et al.* propozoi një model që kombinon regresionin logjistik dhe ANN për parashikimin e sëmundjeve ose rreziqeve kronike duke mbuluar një rast studimi të hipertensionit bazuar në vlerësimin e rrezikut shëndetësor (HRA). Modeli i integruar përdori të dhëna të disponueshme për të gjithë nga sistemi i mbikëqyrjes së faktorëve të rrezikut të sjelljes (BRFSS) të qendrave të kontrollit dhe parandalimit të sëmundjeve (CDC). Së pari, regresioni logjistik binar u përdor për të zgjedhur elementët e rrezikut që kanë vlera p-pasuese. Gjatë përzgjedhjes së elementëve të rëndësishëm të rrezikut, në të dhënat e eksperimentimit u përdor modeli i regresionit logjistik me shumë faktorë i integruar me vlerësimin e gjasave maksimale të pjeshme (PML) dhe analizën e

regresionit hap përpara. Si rezultat, njëmbëdhjetë faktorë të tillë si mosha, stërvitja, martesja, diabeti, të ardhurat, indeksi i masës trupore, gjinia, zakonet e pirjes së duhanit dhe të pirit u zgjedhën si faktorë të rëndësishëm rreziku. Më pas, u ndërtua dhe u trajnuar rrjeti nervor perceptron me shumë shtresa (MLP) me algoritmin e përhapjes së prapme (BP) për faktorët specifikë për të diagnostikuar nëse një individ po përballonte hipertensionin apo jo. Rritja e numrit të shtresave të fshehura në këtë model MLP-NN mund të përshkallëzojë saktësinë e parashikimeve. Megjithatë, numri i shtresave të fshehura duhet të jetë më shumë se dy të tretat e faktorëve hyrës. Të dhënat u ndanë në grupe trajnimi dhe testimi me një raport 7:3. Saktësia variante nga 71,91% në 72,12% kur numri i shtresave të fshehura u rrit nga 8 në 11. Fillimisht, pragu i paracaktuar u supozua të ishte 0,5, i cili ndryshohej me çdo përsëritje trajnimi. Rezultatet eksperimentale kanë treguar se modeli i integruar i ANN dhe regresionit logjistik është më i mirë se cilido prej të dyve që veprojnë individualisht.

1.7 Objektivi

Qëllimet e ardhshme të këtij studimi janë përfshirja e më shumë faktorëve të rrezikut dhe trajnimi i të dhënave mbi më shumë përsëritje për rezultate më të mira parashikimi. Sindroma akute koronare (ACS) është një gjendje e rëndë mjekësore për shkak të çekuilibrit të krijuar midis nevojave metabolike në trup. Kjo gjendje karakterizohet nga dhimbje gjoksi që rrezaton në qafë dhe krahun e majtë. Profesionistët mjekësorë përshkruajnë teste të ndryshme laboratorike dhe EKG bazuar në kushtet e pacientit. Berikol *etj.* [9] studioi modele ndërtimi për diagnostikimin e ACS duke përdorur metodën ML për SVM me disa attribute të pacientit si mosha, faktorët e rrezikut, seksi dhe prania e enzimave kardiake.

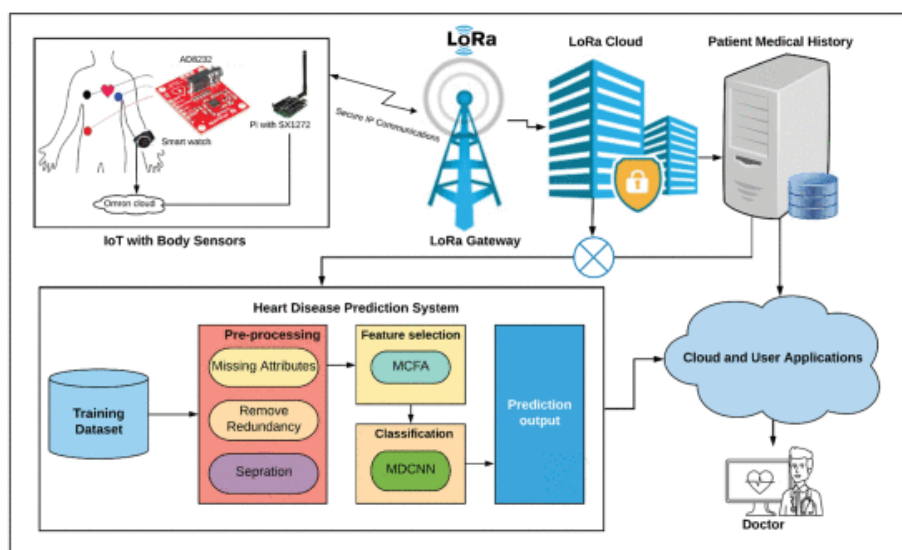


Figura 3. Kuadri i propozuar i sistemit të parashikimit të sëmundjeve të zemrës.

1.8 Studime të tjera të rëndësishme

Berikol *etj.* [9] përdori të dhënat e 228 pacientëve me historinë e tyre mjekësore, klinike, laboratorike dhe informacione imazherike për të ndërtuar një model ML. Në këtë studim u testuan katër metoda duke filluar me SVM, NB, ANN dhe regresionin logjistik. Rezultati tregoi se SVM kishte saktësinë më të lartë në grupin e pacientëve duke filluar nga 19 vjeç në 91 vjeç. U konstatua se nga 228 pacientë, 99 ishin të listuar si me ACS dhe 129 si pa ACS. Modeli SVM dha 99.13% parashikime të sakta në diagnostikimin e ACS. Diagnostifikimi është një fushë e gjerë dhe metodat moderne shkencore janë gjithmonë në zhvillim. Ky studim tregoi sistemi mund të integrohet me departamentet mjekësore për vendimmarrje në të ardhmen. Megjithatë, studimi u krye në historinë e pacientit dhe gjetjet e EKG. Faktorët e rrezikut të pacientëve nuk u përfshinë në studim që është një kufizim i madh. Matja e dhimbjes mund të ketë nevojë për një qasje më praktike sesa analiza teorike. Së fundi, kërkohen më shumë të dhëna në kohë reale për trajnimin e modelit dhe marrjen e vendimeve në kohë reale.

Sistemet IoT janë ende në zhvillim dhe janë të detyruar të sjellin më shumë ndryshime teknologjike. Më shumë të dhëna, metoda kërkimore dhe teknika të avancuara do të ndihmojnë në përmirësimin e vlerësimeve për saktësi më të mirë në të ardhmen. Përveç analizës së EKG-së e cila është duke u hulumtuar në mënyrë aktive, një sfidë tjetër është zhvillimi i pajisjeve të veshura të cilat mund të përdoren vazhdimisht dhe janë financiarisht të qëndrueshme. Zgjidhjet moderne të veshjes mund të ndihmojnë për të përmirësuar më tej procesin dhe për të sjellë një sistem më të sofistikuar të kujdesit shëndetësor IoT për vendimmarrje të përmirësuar.

Yadav *etj.* [10] zhvilloi një sistem për të parashikuar fillimin e sëmundjeve të zemrës tek pacientët në fazat e tyre të hershme duke përdorur algoritmin NB. Ndërfaqja grafike e përdoruesit e lehtë për t'u përdorur dhe kërkesa për një numër të vogël regjistrimesh për trajnimin e klasifikuesit NB e bënë këtë sistem veçanërisht të dobishëm. Të dhënat u morën nga spitali i Cleveland-it dhe përbëheshin nga 303 pacientë me të dhënat e tyre të moshës, gjinisë, raporteve të EKG-së, presionit të gjakut, etj. Të dhënat hyrëse u përdorën në dy forma, fillimisht për të trajnuar modelin dhe më pas për të testuar të dhënat e mbetura për të gjetur efikasitetin e sistemit. Rezultatet janë marrë në formë binare ku 1 parashikon praninë e sëmundjeve të zemrës dhe 2 parashikon mungesën e sëmundjeve të zemrës. Megjithëse modelet treguan efikasitet të lartë prej 85%, kërkohet trajnim i mëtejshëm për grupe të dhënash më të mëdha përpara se të miratohen në skenarët e jetës reale.

Choi *et al.* [11] përdori NLP për të marrë fjalitë e pacientit (sekuenca të renditura të kodeve të ndryshme mjekësore) si vektorë të shpërndarë duke përdorur metodën skip-gram dhe metodat e klasifikimit si regresioni logjistik, rrjeti nervor, SVM dhe KNN. Duke përdorur ngulitjen me skip-gram, krijohen përfaqësime të konceptit dhe takimit nga të cilat më pas rrjedhin vektorët e pacientit. Këto përfaqësonin historinë mjekësore të pacientit dhe përdoren për të parashikuar dështimin e zemrës. Përmirësim i dukshëm në parashikimin e dështimit të zemrës u vu re nga

autorët. Jo vetëm që modeli duke përdorur metoda ML ishte më i saktë, por edhe koha e nevojshme për të trajnuar këtë model ishte më e vogël në krahasim me modelet e tjera.

Tabela *etj.* [12] paraqiti një qasje pa pranga për matjen e vazhdueshme të presionit të gjakut duke përdorur kamerat e smartfonëve për të regjistruar sinjalet e fotopletizmograme (PPG) nga majat e gishtave të një personi dhe duke i konvertuar ato në vlerat e BP-së duke përdorur një marrëdhënie të thjeshtuar lineare.

1.9 Parametrat

Parametrat në këtë marrëdhënie janë specifike për personin, BP e të cilit llogaritet dhe gjenden duke përdorur regresionin linear. Modeli i regresionit linear është trajnuar duke përdorur metodën LOOCV. Modeli i trajnuar testohet dhe BP e vlerësuar krahasohet me BP të matur me një monitor oscilometrik të presionit të gjakut. Tabei *etj.* [13] mati si presionin diastolik të gjakut (DBP) ashtu edhe presionin sistolik të gjakut (SBP), grafikët e shpërndarjes dhe korrelacionet. Janë tërhequr linjat e regresionit midis vlerave të vlerësuar dhe faktike për të dyja, gjë që tregoi se ekziston një lidhje lineare pozitive midis tyre. MAE dhe devijimi standard (SD) I gjetur për DBP ishin përkatësisht 2.98 dhe 2.49 mm Hg, ndërsa ato për SBP ishin përkatësisht 3.28 dhe 2.46 mm Hg. Këto vlera qëndrojnë nën standardin universal [14] të MAE prej 5 mm Hg dhe SD prej 8 mm Hg, duke e bërë metodën e propozuar një qasje praktikisht të përdorshme.

TABELA 3. Përdorni rastet e algoritmeve ML për prognozën dhe kontrollin e përhapjes.

Viti	Rasti i përdorimit	Algoritmi	Përparësitë	Të metat
2018	Parashikimi i fillimit të sëmundjeve të zemrës në fazat e hershme	NB	GUI i lehtë për t'u përdorur, kërkohet më pak trajnim	Vendosja në botën reale kërkon përfshirjen e faktorëve të ndryshëm dytësor që nuk përfshihen në model.
2018	Zbulimi i virusit të gripit	NLP	Koha më e shkurtër e trajnimit, nuk kërkon parapërpunim për vlerat që mungojnë	Të dhënat e përdorur kanë ardhur nga një sistem i vetëm shëndetësor.
2019	Klasifikimi i modeleve të rrezikut të epilepsisë	KNN	Dendësi spektrale të fuqisë së reduktuar në dimensione u përdorën për vlera më të mira të prodhimit.	Numri i lartë i alarmeve fallco
2019	Parashikimi i rekuperimit të shokut hemoragjik	Regresioni logjistik	E tejkalon klasifikuesin bazë në protokollet e shumta të testimit	Të dhënat e testit të përdorura ishin të vogla
2020	Identifikimi i COVID-19	CNN	Dallon me sakësi shtamet SARS-Cov-2 nga viruset e tjera	Eshtë testuar duke përdorur një numër të vogël sekuecash gjenomike
2021	Korniza cloud e bazuar në IoT për kontrollin e përhapjes së Eboles	Pema e vendimit	Zbatim i lehtë në shkallë të gjerë duke përdorur RFID dhe sensorë të veshur	Modeli i vlerësimit të të dhënave që mungojnë nuk është zhvilluar

Ganesan dhe Sivakumar [15] propozuan mbledhjen e informacionit në kohë reale për pacientët që përdorin pajisje IoT të veshur. Kjo përfshin presionin e gjakut, sheqerin në gjak dhe EKG. Për të trajnuar modelin, u përdor baza e të dhënave UCI e cila përfshin regjistrat e kaluar të të dhënave mjekësore të mbledhura nga institucionet mjekësore. Pas trajnimit, modeli testohet për të dhënat hyrëse të pacientëve.

Klasifikuesit e përdorur ishin J48, regresioni logjistik, MLP dhe SVM. J48 dha saktësi më të mirë se regresioni logjistik, megjithëse regresioni logjistik performoi më mirë se MLP. Një çështje e hapur në këtë zone është ndërtimi i një modeli për të trajnuar mbi lloje të ndryshme të dhënash.

Kjo është për shkak se statusi i pacientit merret në forma të ndryshme si imazhe, regjistrime testesh, grafikë, etj. Që sistemi të funksionojë siç duhet, modeli duhet të identifikojë llojin e hyrjes, ta klasifikojë atë dhe ta normalizojë atë me të gjitha formatet e disponueshme të hyrje.

Kreu II

Machine Learning

Kohët e fundit, kujdesi shëndetësor po përdor gradualisht teknologjitë në zhvillim si Interneti i Gjërave (IoT), Mësimi i Makinerisë (ML) dhe kompjuteri cloud për ofrimin e sistemeve inteligjente të kujdesit shëndetësor për të përmirësuar diagnostikimin dhe trajtimin efektiv të shëndetit. Qasjet ML në sistemin e kujdesit shëndetësor ofron shërbime inteligjente për monitorimin e kujdesit shëndetësor dhe automatizimin mjekësor në perspektiva dhe mjedise të ndryshme (spitale, zyra, shtëpi, etj.).

2.1 Machine Learning & IoT

Sektorët e kujdesit shëndetësor po pranojnë gradualisht teknologjinë që ofron lehtësira të ndryshme të kujdesit shëndetësor në distancë, parashikimin e sëmundjeve dhe aftësitë e diagnostikimit në shtëpi, të cilat kombinojnë Machine Learning (ML) dhe Internetin e Gjërave (IoT) të përmbledhura në 7 pika kryesore[16][17]:

1. Kujdesi shëndetësor po përdor teknologjitë në zhvillim si IoT, ML dhe kompjuteri cloud për të përmirësuar diagnostikimin dhe trajtimin efektiv të shëndetit.
2. Qasjet ML në sistemin e kujdesit shëndetësor ofrojnë shërbime inteligjente për monitorimin e kujdesit shëndetësor dhe automatizimin mjekësor në perspektiva dhe mjedise të ndryshme.
3. Arkitekturat e IoT dhe cloud janë shfrytëzuar për të bërë sisteme inteligjente të kujdesit shëndetësor të afta për të lidhur aplikacione në kohë reale.
4. Teknikat e bazuara në AI dhe ML kryejnë automatizim inteligjent, duke analizuar dhe parashikuar informacione kuptimplote dhe janë në gjendje të mësojnë nga të dhënat e tyre të mbledhura, pa nevojën e qenieve njerëzore.
5. Integrimi i IoT dhe ML kishte siguruar një sistem klinik të zgjuar dhe inteligjent për mbështetjen e vendimeve, i cili ndihmon në zbulimin e sëmundjeve në një fazë të hershme duke përmirësuar shkallën e mbijetesës së pacientit.
6. Teknikat e ML kanë dhënë një kontribut kyç në diagnostikimin e sëmundjeve në kujdesin shëndetësor, duke lejuar kështu një reduktim të ndjeshëm të kostove të vizitës tek mjeku dhe një përmirësim të përgjithshëm të cilësisë së kujdesit ndaj pacientit.
7. Integrimi i fuqisë së IoT dhe ML për kujdesin shëndet

2.1.1 Fazat e zhvillimit të ML për kujdesin shëndetësor.

ML ka sjellë revolucion dhe po miratohet gjerësisht në kujdesin shëndetësor për shkak të performancës së tij ekstra të zakonshme në detyra të ndryshme të gjithanshme si p.sh njohja e organeve të trupit duke përdorur imazhe [18], klasifikimi dhe zbulimi i sëmundjes [19-20] dhe segmentimi dhe rindërtimi i imazheve mjekësore [21-23]. Teknika përdoret në një sërë aplikacionesh të kujdesit shëndetësor, duke filluar nga sinjalet njëdimensionale të zemrës deri te imazhet mjekësore shumëdimensionale për diagnozën me ndihmën e kompjuterit. Besohet se në të ardhmen e afërt, aplikacionet e ekspertëve do të ndihmojnë mjekët për ekzaminimin e pacientëve. Modeli ML ka arritur inteligjencën duke konkurruar me inteligjencën njerëzore në disa fusha klinike si radiologjia [24] dhe dermatologjia [25]. Teknologjitë më të fundit të avancuara si të dhënat e mëdha, kompjuteri në renë kompjuterike dhe shkenca e të dhënave [26] i kanë shtuar një avantazh për të prodhuar rezultate më të sakta. Këto teknologji së bashku kanë luajtur një rol vendimtar në mundësimin e shërbimeve shëndetësore në zonat e thella rurale.

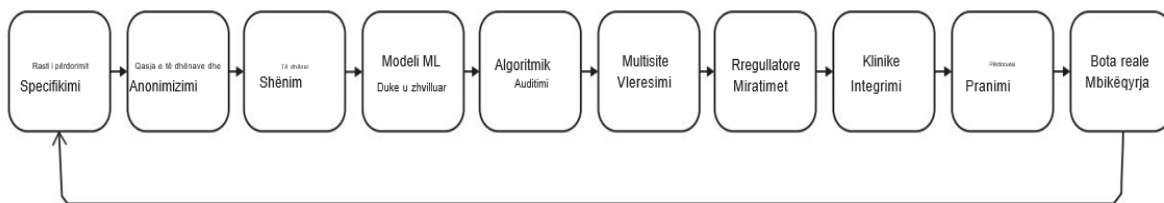


Figura 6.1 Fazat kryesore për zhvillimin e sistemeve të kujdesit shëndetësor të bazuar në PP

2.2 Llojet e teknikave të mësimit të makinerisë

2.2.1 Unsupervised Learning

Mësimi i pambikëqyrrur është një teknikë ML e cila trajnon makinën me të dhëna të paetiketuara dhe kështu quhet edhe harta vetë-organizuese. Ritmet algo sipas kësaj metode grumbullojnë të dhënat në bazë të attributeve të tyre të ngjashme, tendencës, modeleve, etj. Metoda nxjerr rregullsitë nga grupi i të dhënave për reduktimin e të dhënave me dimensione të larta në dimensione të ulëta. Metoda mund të zbatohet në të gjitha llojet e të dhënave, por disavantazhi kryesor i kësaj metode është se rezultati është i paparashikueshëm.

Grumbullimi dhe analiza e komponentit kryesor (PCA) është dy teknika të rëndësishme të të mësuarit të pambrojtur që përdoren në kujdesin shëndetësor. Një nga shembujt e përdorimit të mësimit të pambikëqyrrur është parashikimi i

sëmundjeve të zemrës bazuar në teknikat e grupimit [28] dhe PCA po përdoret për parashikimin e hepatitit [29].

2.2.2 Supervised Learning

Mësimi i mbikëqyrur është një teknikë ML e cila trajnon makinën me të dhëna të etiketuara. Gjatë trajnimit, makina pajiset me të dhëna hyrëse së bashku me rezultatin e pritur pasi të përfundojë trajnimit, makina pritët të parashikojë etiketën e saktë për të dhënat e reja të futura si në hyrje.

Dy teknikat më të përdorura të të mësuarit të mbikëqyrur klasifikohen në bazë të rezultateve të gjeneruara. Nëse prodhimi është diskret, atëherë ai klasifikohet si klasifikim; megjithatë, nëse prodhimi është i vazhdueshëm, atëherë ai klasifikohet si regresion. Një nga shembujt në kujdesin shëndetësor duke përdorur mësimin e mbikëqyrur është analizimi i imazheve mjekësore dhe njohja e organeve të ndryshme të trupit [30] dhe klasifikimi i sëmundjeve të mushkërive [31].

2.2.3 Semi Supervised Learning

Semi Supervised Learning është një teknikë ML e cila trajnon makinën me të dhëna të etiketuara dhe të paetiketuara; zakonisht, të dhënat e etiketuara janë shumë të vogla, ndërsa të dhënat e paetiketuara janë në sasi të mëdha. Me këtë algoritëm, fillimisht, duke përdorur metodën e mësimit të pambikëqyrur, të dhënat e paetiketuara grumbullohen dhe më pas ato etiketohen duke përdorur të dhënat ekzistuese të etiketuara. Pak nga aplikacionet e kujdesit shëndetësor që përdorin mësimin gjysmë të mbikëqyrur janë analiza e të folurit dhe klasifikimi i sekuencës së proteinave.

2.3 Aplikimet e Machine Learning

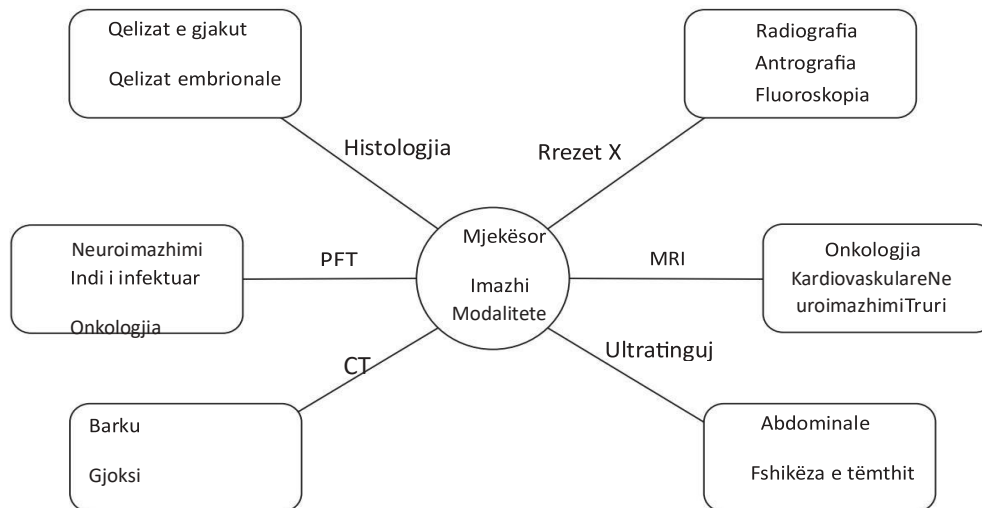
ML po mundëson analizën e të dhënave të mëdha heterogjene që gjenerohen në kujdesi shëndetësor në baza ditore nga disa burime siç tregohet në figure.

ML përfiton katër aplikime kryesore të kujdesit shëndetësor, përkatësisht, prognozën, hundën e diagnostikimit, trajtimin dhe rrjedhën klinike të punës.

2.3.1 Prognoza

Është terminologjia mjekësore që përdoret për parashikimin e zhvillimit të sëmundjes, komplikimeve të krijuara nga sëmundja në shëndet, ndikimin e saj në jetë, parashikimi i kohëzgjatjes së sa kohë do të marrë sëmundja për t'u kuruar, dhe nëse simptomat do të mbeten të qëndrueshme ose do të përkeqësohen ose përmirësohen me fazën e kohës,

ndikimi i tij në stilin e jetës nëse një pacient do të jetë në gjendje të kryejë aktivitetet e tij të përditshme; dhe ndikimi i saj në jetëgjatësinë.



2.3.2 Diagnoza

Regjistrimet Elektronike të Shëndetit (EHR)

Një sasi e madhe të dhënash të strukturuar dhe të pastrukturuar mblidhen si

raporte mjekësore të pacientëve nga disa burime që formojnë EHR në rutinën e përditshme. Për diagnozën, të dhënat klinike nxirren nga këto të dhëna duke përdorur metodat ML[32].

Analiza e imazhit mjekësor

Përqëllime terapeutike dhe diagnostike, imazhet mjekësore mblidhen nga disa modalitete si radiografia, angiografia dhe Tomografia e Kompjuterizuar (CT). Një tipologji e modaliteteve të imazherisë mjekësore të përdorura zakonisht është ultratingulli, rrezet X, MRI, CT, PET, etj. [33], siç tregohet në figurë.

Trajtimi

Pasi imazhi të ekzaminohet, raporti për trajtimin duhet të gjenerohet nga mjeku ose radiologu, i cili është një detyrë sfiduese dhe kërkon kohë, kështu që studiuesit po përpiqen të hartojnë kornizën duke bashkëpunuar NLP (Natural language Processing) dhe ML; pak nga kornizat diskutohen në [34-36].

Rrjedha klinike ML

Është pranuar gjerësisht në kujdesin shëndetësor për potencialin e saj drejt zhvillimit dhe prashikimit të sëmundjes, në mënyrë që ka provuar të sigurojë një efikasitet në parashikimin e rrezikut në sëmundjet kardiovaskulare trajtimi në kohë të jetë i mundur. Për shembull, ML [37], kancerin dhe disa sëmundje të tjera, përkatësisht, mëlçinë yndyrore dhe diabetin [38].

2.4. Algoritmevet e përdorura

Ky seksion ofron një përmbledhje të shkurtër të algoritmeve të përdorura në H-IoT dhe disa shembuj të aplikacioneve të tyre H-IoT. Këto algoritme mund të ndahen në algoritme mësimore të mbikëqyrura dhe të pambikëqyrura, në varësi të faktit nëse etiketat e dëshiruara të klasifikimit janë dhënë në grupin e të dhënave hyrëse (të mbikëqyrura) apo jo (të pambikëqyrura). Algoritmet e mësimimit të makinerive në përgjithësi preferojnë të dhëna të etiketuara, ndërsa algoritmet e mësimimit të thellë janë më të aftë në shfrytëzimin e të dhënave të paetiketuara.

2.4.1 Algoritme të Reduktimit të Dimensionalitetit (DRA)

DRA janë një grup algoritmesh të tilla si analiza diskriminuese lineare (LDA) dhe analiza e komponentit kryesor (PCA) të cilat marrin grupe të mëdha të dhënash si hyrje, identifikojnë korrelacionet dhe modelet në to dhe ofrojnë një grup shumë më të vogël të dhënash (përsa i përket numri i dimensioneve) si rezultat pa humbur asnjë informacion kritik të dhënë më parë. Kjo heq mospërputhjet, të dhënat e tepërta dhe veçoritë shumë të ndërlydhura të të dhënave. Reduktimi i dimensionalitetit kryhet nëpërmjet hapave të mëposhtëm: [39]

- Standardizimi: çdo pikë e të dhënave është shkallëzuar si:
$$\text{Variabla e re} = (\text{ndryshore origjinale} - \text{mesatarja e të gjitha variablave të veçorisë}) / (\text{devijimi standard i të gjitha variablave të veçorisë}).$$
- Llogaritja e matricës së kovariancës: $[\text{veçori}] [\text{veçori}]^T$.
- Llogaritja e eigenvectors dhe eigenvalues.
- Llogaritja e komponentëve kryesorë.

Premtime të rëndësishme janë bërë për diagnostikimin e sëmundjes së Parkinsonit dhe kancerit të gjirit duke kombinuar IoT dhe DRA si LDA [40]. Rëndësia e një kombinimi të IoT dhe DRA për të rritur aftësitë diagnostikuese është diskutuar mirë në literaturën e disponueshme [41][42].

2.4.2 Analiza Diskriminuese

Analiza diskriminuese, e cila është një lloj i rëndësishëm i DRA, projekton pikat e të dhënave në një hapësirë me dimensione më të ulëta, në mënyrë që klasat të ndahen në mënyrë të përshtatshme në grupe që nuk mbivendosen. Një klasifikim i tillë është i ngjashëm me regresionin e shumëfishtë kur përfshihen vetëm dy grupe, por rezulton se bëhet më i ndërlikuar me rritjen e numrit të grupeve. Në kujdesin shëndetësor, analiza diskriminuese gjen aplikime kryesisht në matjen e prognozës dhe ashpërsisë së sëmundjes tek një pacient. LDA klasifikohet duke gjetur një kombinim linear të veçorive, ndërsa një analizë më e përgjithshme diskriminuese e shumëfishtë shtrihet e njëjta në një hapësirë jolineare.

2.4.3 Regresioni Linear

Regresioni linear është një teknikë modelimi e cila përdor një qasje lineare për të gjetur një marrëdhënie midis një ndryshoreje të varur dhe një ose më shumë variabla të pavarur. Preferohet në rastet kurekzistojnë vetëm variabla të pavarura të vazhdueshme.

Ekzistojnë shumë teknika për përgatitjen e një modeli të regresionit linear, nga të cilat më së shpeshti përdoren katrorët më të vegjël të zakonshëm dhe metoda e zbritjes së gradientit. E para përpiqet të minimizojë drejtpërdrejt shumën e vlerave të gabimit në katror për të gjetur koeficientët, ndërsa e dyta përdor një qasje përsëritëse për të minimizuar shumën e mbetjeve në katror.

2.4.4 Regresioni Logjistik

Regresioni logjistik është një algoritëm i bazuar në probabilitet në të cilin funksioni i tij sigmoid vepron si funksion i kostos dhe merr një vlerë midis zeros dhe një. Regresioni logjistik mund të jetë i dy llojeve. Kur vëzhgimet duhet të klasifikohen në dy klasa, përdoret regresioni logjistik binar, ndërsa më shumë se dy klasifikime të klasave kërkojnë përdorimin e regresionit logjistik shumënomial. Përdorimi i regresionit logjistik është veçanërisht i favorshëm kur problemi i regresionit ka një ndryshore të varur dikotomike.

2.4.5. Makinë me Vektor Mbështetjeje

Ky algoritëm përdor konceptin e një hiperplani klasifikues.

Qëllimi është të identifikohet një plan që ndan grupin e të dhënave në dy grupe në mënyrë që hendeku midis pikave të të dhënave në dy grupe të maksimizohet. Ky hiper-avion thuhet se ka rrezen maksimale. Pikat e të dhënave që bien në anët e ndryshme të hiperplanit u caktohen grupeve të ndryshme.

Dimensioni i hiperplanit varet nga numri i veçorive. Për veçori më pak se ose të barabartë me 2 në numër, hiperplani është thjesht një vijë. Kthehet në një aeroplan 2-D për 3 veçori, ndërsa

imazhi i tij për më shumë se 3 veçori bëhet i vështirë. SVM-të janë të favorshme pasi ato janë jashtëzakonisht elastike ndaj problemeve të tepërta. SVM-të jo vetëm që mund të funksionojnë si klasifikues linearë, por gjithashtu mund të përdorin bërthama jolineare për të klasifikuar grupet e të dhënave duke përdorur funksione jolineare.

Ginantra *etj.* [43] propozoi një model që demonstroi një klasifikues SVM i cili ishte më i mirë se klasifikuesit e tjerë për identifikimin nëse një person vuan nga sëmundje të ngjashme me gripin (ILI) (dmth, infeksione akute të frymëmarrjes). SVM është gjetur të jetë më e sakta në verifikimin e vendndodhjes pa kërkesën e të dhënave të karakteristikave të kanalit për të funksionuar.

SVM-të u përdorën gjithashtu për të zhvilluar metoda për zgjidhjen e detyrës së klasifikimit të materialeve implantare mjekësore [44].

2.4.6 K Nearest Neighbors(KNN)

KNN është një metodë e mbikqyrur e të mësuarit ku objektet klasifikohen në bazë të ngjashmërisë së tyre me veçori të caktuara të objekteve të tjera, kategoria e të cilave është e paracaktuar. Distanca, kryesisht Euklidiane, llogariten për objektin kategoria e të cilit është

e vendosur për k fqinjët e saj më të afërt. Ky është ndryshimi midis veçorive të fqinjëve është marrë dhe përmbledhur për të gjetur distancën. Distanca Euklidiane jepet nga barazimi. 2. Më pas bëhet votimi për të përcaktuar kategorinë në të cilën bëjnë pjesë shumica e k objekteve më të afërta. Vlera e k përcaktohet përmes procesit të akordimit të parametrave. Zakonisht zgjidhet të jetë rreth rrënjës katroretë numrit të përgjithshëm të objekteve dhe në përgjithësi është një numër tek për të shmangur mundësinë që kategori të shumta të marrin vota të barabarta. KNN është i dobishëm për klasifikimin e të dhënave të etiketuara edhe kur grupi i trajnimit është shumë i vogël dhe përdoret gjerësisht në aplikacione të ndryshme. Ahmed [45] përdori KNN në të dhënat e mbledhura nga pajisjet IoT për të parashikuar sulmet në zemër. Ai u përdor në të dhënat e mbledhura nga 20 sensore Kinect për të matur pozicionet e ndryshme në tru.

Këto u kaluan më pas përmes klasifikuesve KNN të përdorur me distancat Euklidiane dhe Minkowski për të parashikuar aktivitetin aktual të përdoruesit. Kjo ka një potencial të madh për t'u përdorur në matjen e fitnesit pasi të zhvillohet klasifikimi për një sërë aktivitete.

Azimi *etj.* [46] propozoi përdorimin e imputimeve të shumta të KNN për të vlerësuar pikat e humbura ose të munguara të të dhënave të mbledhura për të monitoruar gratë shtatzëna. Ky aplikacion mund të përdoret me besueshmëri, si një shërbim mjekësor B2C ose si një mjet për kërkime mbi shëndetin e nënës. Hossain *et al.* [47] shfaqti një aplikacion tjetër për matjen e aktivitetit duke përdorur sensorë të punës me rrjet të gjerë me rreze të gjatë (LoRaWAN) dhe akselerometër për zbulim duke përdorur KNN me një saktësi prej 80%.

2.4.7. Pema E Vendimit

Një pemë vendimi përbëhet nga tre komponentë: nyjet e brendshme, degët dhe nyjet e gjetheve të cilat përfaqësojnë përkatësisht veçoritë, rregullat e vendimit dhe rezultatet. Indeksi Gini dhe entropia janë dy nga metodat më të përdorura për klasifikimin e të dhënave.

Cho [48] përdori pemë vendimi për të gjurmuar vendndodhjen e personave gjatë një pandemie. Xie *et al.* [49] përdori pemët e vendimit për të zhvilluar një algoritëm klasifikimi të rrahjeve të zemrës që identifikon tkurrjen e parakohshme ventrikulare (PVC) për të diagnostikuar mia të aritit duke përdorur amplituda dhe intervale të rrahjeve të zemrës si veçori.

2.4.8. Random Forest Classifier (RFC)

Pemët e vendimit përshtaten sipas të dhënave të veçanta të përdorura për trajnimin e tyre. Rezultatet e marra duke përdorur një pemë vendimi ndryshojnë në mënyrë drastike nëse të dhënat e trajnimit ndryshohen. Ky algoritëm është llogaritës i shtrenjtë. Një optimë lokale në përgjithësi llogaritet sepse kthimi prapa nuk është i mundur pasi të ndodhë ndarja. Metoda random forest i adreson këto kufizime. Në këtë model, disa pemë vendimi trajnohen njëkohësisht për të prodhuar një rezultat të vetëm. Një bashkim i tillë i pemës së vendimit quhet thasë. Si shembull, Al Hossain *et al.* [50] demonstroi aplikimin e një modeli të rastësishëm pyjor që i tejkaloi modelet e tjera me 95% saktësi në parashikimin e numrit të njerëzve të infektuar nga gripi në vende publike. Ai tregon një saktësi të lartë për shkak të aftësisë së tij për të kombinuar rezultatet e të gjitha pemëve të vendimit. Gupta *etj.* [51] paraqiti një klasifikues të rastësishëm pyjor që ia kalonte KNN, SVM dhe pemës së vendimit me 77.8% saktësinë zbulimin e lëvizjes anormale të turmës

2.4.9 Naive Bayes (NB)

Klasifikimi NB bazohet konceptualisht në theori të Bayes. 'Naive' i referohet faktit që të gjitha tiparet supozohen të jenë të pavarura nga njëra-tjetra. Të dhënat ndahen në një matricë tipare dhe një vektor përgjigjeje. Rreshtat e matricës së veçorive ofrojnë të gjithë mbledhjen e të dhënave në terma të vektorëve, secili prej të cilëve përfaqëson llojin e

variablit relativ. Nga ana tjetër, çdo rresht i vektorit të përgjigjes përfaqëson një klasë rezultati.

Sadhukhan *et al.* [52] dhe Assery *et al.* [53] përmendi situatat ku NB ia kalonte të gjithë klasifikuesve të tjerë për të klasifikuar tweet-et që mund të ndihmojnë në menaxhimin e çështjeve të rrjeteve sociale në periudha fatkeqësish ose pandemie.

2.4.10 Rrjetet Neurale Konvolucionale (CNN)

CNN është një rrjet feed-forward që përdoret në problemet e klasifikimit [54]. Ai zbërthen hyrjen në përbërës dhe më pas i kalon ato në një shtresë konvolucioni, e cila i vendos këto pjesë në kombinime të ndryshme derisa të dalin disa modele prej tyre (përbërje). Një shtresë e njësisë lineare të korrigjuar (ReLU) më pas harton imazhet hyrëse kundrejt këtyre modeleve për të formuar një shtresë tipare të korrigjuar dhe i kalon ato në një shtresë bashkimi. Shtresa e bashkimit redukton hartën për të dhënë një hartë të veçorive të

bashkuara, e cila më pas rrafshohet për të formuar një vektor linear dhe futet në një rrjet plotësisht të lidhur për të kategorizuar hyrjen. CNN-të përdoren gjerësisht në zonat që kërkojnë interpretim vizual të imazheve me një topologji të ngjashme me rrjetin.

2.4.11 Rrjeta Neurale Artificiale (ANN)

Një rrjet nervor artificial (ANN) është një model ML që imiton procesin e të mësuarit të trurit të njeriut: ekziston një shtresë hyrëse që merr të dhënat që do të përpunohen, disa shtresa që përpunojnë të dhënat dhe një shtresë dalëse që jep rezultatin. Në ANN, shtresat e fshehura marrin hyrje të ndërmjetme, caktojnë një peshë të rastësishme dhe paragjykim për secilën prej hyrjeve dhe llogaritin disa shuma të ponderuara, të cilat më pas kalohen nëpër shtresa me pesha dhe shuma, derisa të arrijnë në shtresën e fundit, e cila përdor një funksion aktivizimi për të përcaktuar rezultatin. Kur rezultatet janë të pasakta, ato kthehen në shtresat e mëparshme (përhapja e pasme) në përputhje me një funksion të kostos për të ndryshuar peshat derisa përgjigjet të merren me saktësi të mjaftueshme.

2.4.12 Mësimi I Makinës Së Përforsimit

Të mësuarit përforsues bazohet në metodologjinë me të cilën foshnjat mësojnë të interpretojnë botën përreth tyre. Kjo përfshin një agjent, ose nxënësin, i cili fillon nga një gjendje specifike në mjedis, le të themi S_0 . Agjenti më pas ndërmerr një veprim, A_0 , dhe gjendja përditësohet në S_1 . Nëse veprimi ishte në drejtimin e duhur, mjedisi e shpërblen agjentin me R_1 . Kjo përsëritet derisa shpërblimi të maksimizohet. Ai përbëhet nga një grup algoritmesh si Monte Carlo dhe Q Learning.

Park *et al.* [55] diskutoi diagnostifikimin e automatizuar të vëllimeve të larta të pacientëve përmes pajisjeve të veshshme IoT duke përdorur mësimin Q. Zhao *et al.* [56] ofroi një metodë të re për të drejtuar turmat në një qytet të zgjuar duke përdorur mësimin përforsues, një teknologji që mund të jetë thelbësore për heqjen e bllokimeve gjatë COVID-19 duke mbajtur në përputhje me normat e distancimit shoqëror.

2.4.13 Përpunimi I Gjuhës Natyrore (NLP)

NLP i referohet aplikimit të algoritmeve ML për kompjuterët për të kuptuar dhe interpretuar gjuhën natyrore njerëzore, fjalimin dhe tekstin. Karakteristikat e tij gjuhësore të nxjerrjes e bëjnë jashtëzakonisht të lehtë përpunimin dhe përcaktimin sasior të informacionit të pastrukturuar. Algoritmi përdor qasjen e mëposhtme:

- Rrjedhimi dhe simbolizimi: zbërthimi i fjalëve në fjalë rrënjësore dhe kategorizimi i tyre.
- Analiza semantike dhe zbërthimi: nxjerrja e kuptimit të mirëfilltë të fjalëve të përdorura dhe zbërthimi i tyre në kontekst dhe qëllim.
- Modelimi i temës: kuptimi i fushës së cilës biseda është pjesë e NLP

2.4.14 Automatika Kognitive

Automatizimi kognitiv është një nën-element i AI. Ai përdor teknologji të avancuara si nxjerrja e të dhënave, njohja e emocioneve, NLP dhe arsyetimi kognitiv me objektivin e imitimit të inteligjencës njerëzore. Automatizimi kognitiv përpiket të imitojë inteligjencën njerëzore duke përdorur teknologjinë për të zgjidhur problemet. Ai vepron si një mekanizëm katalizator pas përgjigjeve efikase dhe të përmirësuara të krijuara nga një pajisje AI.

Automatizimi kognitiv, duke ofruar një qasje më kolektive ndaj H-IoT, ndihmon në aplikacionet ku kërkohet një përdorim i sinkronizuar i sistemeve fiziologjike dhe psikologjike për trajtimin e urgjencave mjekësore. Muhamedi *etj.* [57] zhvilloi një sistem monitorimi kognitiv të kujdesit shëndetësor 5G që mund të revolucionarizojë sistemet e kujdesit shëndetësor, veçanërisht në qytetet inteligjente duke ekzekutuar një të dhëna dhe një motor njohës të burimeve njëkohësisht. Alhussein *et al.* [58] studioi kornizat njohëse të IoT të zhvilluarapër monitorimin dhe diagnostikimin e epilepsisë

KREU III

H-IOT

Hyrje

IoT dhe algoritmet e avancuara të Machine Learning janë të kombinuara në sistemet e shëndetit për të përmirësuar diagnozën, trajtimin dhe procesin e marrjes së vendimeve. Aplikimi i IoT në fushën biomedicinale ka shkaktuar interes në studime mbi Internetin e Gjithçkaje (IoE), duke përfshirë trajtimin simptomatik dhe vëzhgimin e pacientëve. Përpunimi i informacionit në Internetin e Gjëra të Kujdesit Shëndetësor (H-IoT) përfshin marrjen, ruajtjen, analizën dhe shpërndarjen e të dhënave të gjeneruara nga pajisjet IoT. Këto të dhëna mund të përfshijnë shenja vitale si presioni i gjakut, ritmi i zemrës dhe nivelin e oksigjenit, si dhe nivelet e aktivitetit të pacientit, shembujt e gjumit dhe përdorimin e medikamenteve.

3.1 H-IOT

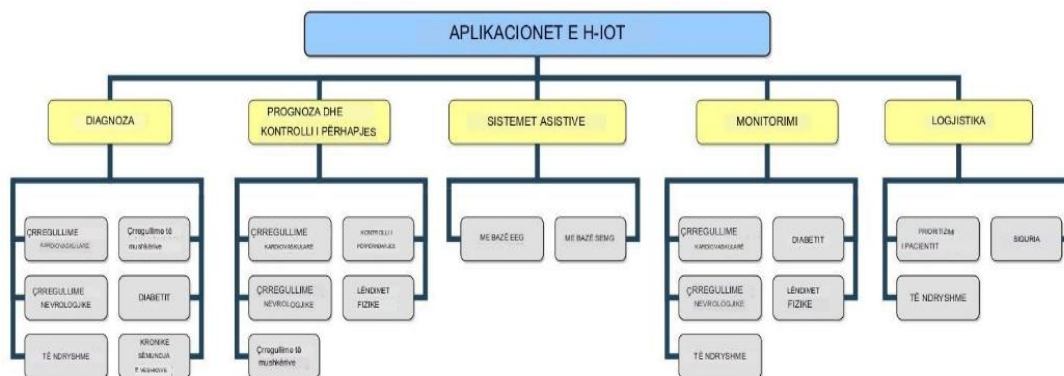


Figura 4. Aplikacionet e H-IOT

Një sistem IoT i kujdesit shëndetësor mund të përkufizohet si një rrjet i të gjitha burimeve të disponueshme shëndetësore të lidhura me njëri-tjetrin për transferimin e shpejtë të të dhënave ndërmjet tyre përmes internetit [59].

Kjo do të thotë që të gjitha burimet e kujdesit shëndetësor si mjekët, spitalet, qendrat e rehabilitimit dhe të gjitha pajisjet mjekësore dhe sensorët së bashku me pacientët ndërlidhen me njëri-tjetrin për transferim të vazhdueshëm të të dhënave në kohë reale. Sensorët e ndryshëm të shoqëruar me aplikacione që interpretojnë leximet e tyre mund të zbulojnë anomali dhe të dërgojnë të dhënat e pacientit tek mjekët/spitalet për diagnostikim dhe analizë, pas së cilës mund të përshkruhen dhe ndërmerren veprime korrigjuese.

Që një kornizë e tillë të ekzistojë dhe të funksionojë pa probleme, duhet të plotësohen tre kërkesa kryesore [60]:

- **Ndërveprueshmëria:** Gama e gjerë e pajisjeve që përdoren në kuadër duhet të jenë në gjendje të bashkëpunojnë me njëra-tjetrën për të mundësuar funksionalitetin e dëshiruar.
- **Vonesa dhe besueshmëria e kufizuar:** Për trajtimin efektiv të emergjencave dhe analizën e sinkronizuar të sasisë së madhe të të dhënave, transmetimi ndërmjet subjekteve në rrjet duhet të jetë i shpejtë dhe i saktë.
- **Privatësia dhe siguria:** Të dhënat personale që transmetohen në një kornizë H-IoT janë të ndjeshme dhe duhet të arrijnë në mënyrë të sigurt

vetëm subjektet përkatëse, gjë që kërkon që vendet të kenë vërtetimin dhe masat e sigurisë. Mekanizma të ndryshëm për autorizimin dhe vërtetimin e pajisjeve IoT janë të disponueshme duke përdorur teknologji të tilla si enkriptimi dhe funksionet fizike të paklonueshme [61].

3.2 ARKITEKTURA E H-IOT

Një sistem H-IoT përbëhet nga një rrjet nga fundi në fund i përbërë zakonisht nga tre shtresa kryesore të funksionimit [62]:

- **Shtresa e mbledhjes së të dhënave:** Kjo shtresë është përgjegjëse për mbledhjen e të dhënave mjekësore nga pajisje të ndryshme sensore të bashkangjitura me pacientin/subjektin e testit që duhet të monitorohet ekzaminohet.
- **Shtresa e ruajtjes së të dhënave:** Kjo shtresë është përgjegjëse për ruajtjen e të dhënave të mëdha të mbledhura nga sensorë të ndryshëm dhe të transmetuara përmes internetit.
- **Shtresa e përpunimit të të dhënave:** Kjo shtresë analizon të dhënat e ruajtura në serverë për të gjeneruar përgjigjen e kërkuar përmes aplikimit të algoritmeve kompjuterike. Gjithashtu, këtu bëhet përpilimi dhe vizualizimi i rezultateve.

Zbatimi i këtyre shtresave mundësohet tregohet në figurë:

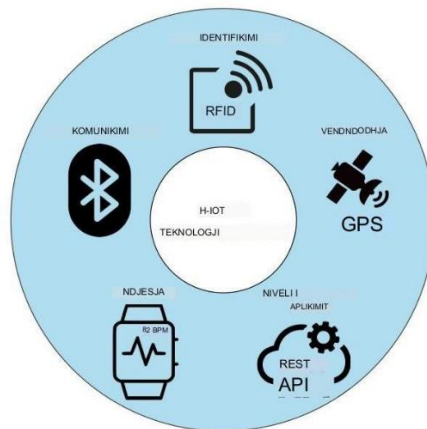


Figura 5. Teknologjitë H-IoT.

3.3 Teknologjitë H-IoT.

- **Teknologjia e identifikimit:** Që nyjet në punën e rrjetit të një kuadri H-IoT të kenë akses në informacion dhe të komunikojnë me njëri-tjetrin në mënyrë të sigurt, secila nyje duhet të identifikohet në mënyrë unike përmes teknologjive të tilla si një identifikues unik (UID) [63].
- **Teknologjia e komunikimit:** Komunikimet në distanca të shkurtra dhe të gjata ndërmjet nyjeve në rrjetin H-IoT kërkojnë shtigje. Komunikimi në distancë kryhet nëpërmjet mjeteve konvencionale si interneti, ndërsa komunikimi në distancë të shkurtër kërkon veçmas teknologji specifike, mundësisht ato që mundësojnë komunikim të shpejtë me valë si Bluetooth, Zigbee, RFID etj.
- **Teknologjia e vendndodhjes:** Sistemi i pozicionimit global (GPS) mundëson nyjet e ndryshme për të gjurmuar me saktësi vendndodhjet gjeografike të njëri-tjetrit, gjë që është jashtëzakonisht e rëndësishme për rastet e caktuara përdorimi të H-IoT. Mund të kërkojnë gjithashtu sisteme të tjera të ndryshme të gjurmimit të vendndodhjes për të kompensuar rastet e lidhjes së dobët GPS [64].

- **Teknologjia e sensorit:** Të dhënat e analizuara për të nxjerrë konkluzione në një sistem H-IoT gjenerohen nga sensorë. Prandaj, mekanizmat e ndjeshmërisë që monitorojnë ndryshimet fiziologjike në kohë reale në trupin e pacientit janë të domosdoshme. Një shumëllojshmëri e madhe sensorësh janë në dispozicion për marrjen e të dhënave të tilla, për shembull, akselerometrat për të ndjerë nxitimin linear, xhiroskopët për matjen e shpejtësisë këndore dhe sensorët e elektrokardiogramit (EKG) për të matur aktivitetin e sigurt elektrik në zemër [65]. Evolucionin në teknologjinë e sensorëve lidhet drejtpërdrejt me evolucionin e kornizave H-IoT duke çuar në parashikime më të sakta dhe kosto më të ulëta.
- **Arkitektura e nivelit të aplikacionit:** Arkitekturat e nivelit të aplikacionit si arkitektura e orientuar drejt shërbimit (SOA) ose transferimi i gjendjes përfaqësuese (REST) lejon që pajisjet e ndryshme në sisteme të funksionojnë në mënyrë të pavarur nga njëra-tjetra. Operacionet e secilës pajisje janë të përcaktuara siç duhet dhe mund të ndryshohen sipas nevojës dhe kur kërkohet pa kompromentuar ndërveprueshmërinë e sistemit.

KREU IV

ECG DIGITALIZATION

Hyrje

Dixhitalizimi i një elektrokardiogramë (EKG) i referohet procesit të konvertimit të sinjalit elektrik të regjistruar gjatë një EKG në një formë dixhitale që mund të ruhet, analizohet dhe shfaqet në një kompjuter ose pajisje tjetër elektronike.

Procesi i dixhitalizimit të EKG-së në përgjithësi përfshin hapat e mëposhtëm:

- 1) Marrja e sinjalit: Një grup elektrodosh vendoset në trupin e pacientit për të kapur aktivitetin elektrik të zemrës. Këto elektroda janë të lidhura me një pajisje për marrjen e EKG-së, siç është një elektrokardiograf, i cili regjistron sinjalin elektrik.
- 2) Amplifikuar dhe filtruar: Sinjali elektrik i kapur përforcohet dhe filtrohet për të eliminuar zhurmën dhe për të nxjerrë në pah karakteristikat përkatëse të EKG-së. Kjo ndihmon në përmirësimin e cilësisë së sinjalit përpara dixhitalizimit.
- 3) Muestreo: Sinjali i përforcuar dhe i filtruar konvertohet në një seri vlerash numerike diskrete përgjatë procesit muestreo. Muestreo nënkupton marrjen e barnave periodike sinjalizuese në intervale kohore të paracaktuara.
- 4) Kuantifikimi: Vlerat e mostrës janë të kuantizuara duke u caktuar atyre një numër specifik në funksion të amplitudës së tyre. Kjo lejon që sinjali analog i vazhdueshëm të përfaqësohet si një seri vlerash dixhitale diskrete.
- 5) Kodimi dhe ruajtja: Vlerat e matura kodohen duke përdorur një format dixhital, siç është standardi i kodimit të EKG (për shembull, formati DICOM), në mënyrë që ato të mund të ruhen në një pajisje ruajtëse, si një kompjuter ose kartë memorie.

Pasi sinjali EKG të jetë dixhitalizuar, ai mund të përdoret për aplikime të ndryshme, të tilla si diagnostikimi i sëmundjeve kardiale, monitorimi i pacientit në distancë, hetimi klinik dhe edukimi mjekësor. Algoritmet dhe programet e specializuara mund të analizojnë dhe shfaqin sinjalin e dixhitalizuar për informacion shtesë në lidhje me rrahjet e zemrës, ritmin e zemrës, kohëzgjatjen e intervalit dhe praninë e anomalive ose modeleve jonormale në EKG. [66]

4.1 KODI

IN:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Pathi I imazhit hyrës
image_path = 'test.jpg'

# Ngarkojmë imazhin hyrës
image = cv2.imread(image_path)

# Kontrollon nëse imazhi është ngarkuar me sukses
if image is None:
    print(f'Error: Failed to load image '{image_path}''')
    exit()

# Konverton imazhin ne shkallë gri
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Aplikojmë Otsu thresholding(është një metodë që gjen vlerën e pragut optimal të sfondit të një imazhi)
ret, thresh = cv2.threshold(gray_image, 120, 255, cv2.THRESH_BINARY +
cv2.THRESH_OTSU)

# Shfaq imazhin e thresholduar
plt.imshow(thresh, cmap='gray')
plt.show()
```

OUT: <matplotlib.image.AxesImage at 0x7fd6381d83d0>

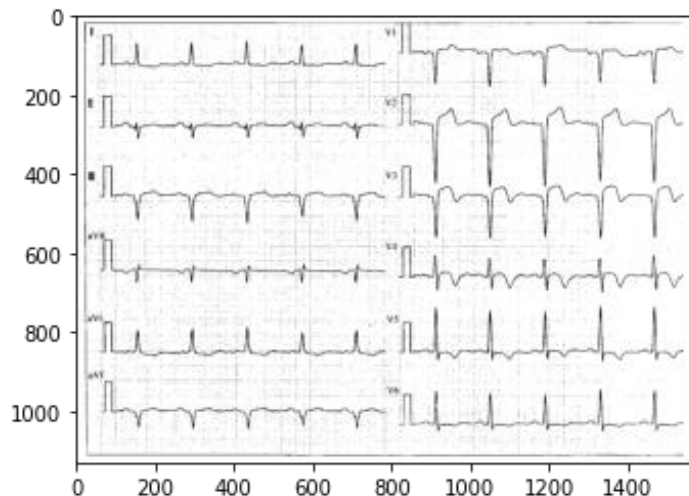


Figura 4.1.1. Outputi 1

Ky kod është një shembull i përdorimit të bibliotekave OpenCV dhe Matplotlib për të ngarkuar një imazh, ta konvertojë atë në ngjyrat e hirit, të aplikojë një treshold Otsu mbi imazhin hyrës, dhe të shfaqë imazhin e tresholduar. Këtu janë hapat kryesorë të këtij kodi: Importohen bibliotekat e nevojshme: cv2 për OpenCV, numpy për manipulimin e imazheve dhe të dhënave, dhe matplotlib.pyplot për shfaqjen e imazheve.

Specifikohet path-i për imazhin hyrës në variablën "image_path".

Ngarkohet imazhi hyrës duke përdorur funksionin cv2.imread() dhe ruhet në variablën "image". Kontrollon nëse ngarkimi i imazhit ishte i suksesshëm. Nëse imazhi nuk u ngarkua me sukses, shfaqet një mesazh gabimi dhe programi mbyllet.

Imazhi konvertohet në shkallë gri duke përdorur funksionin cv2.cvtColor() dhe ruhet në variablën "gray_image".

Aplikohet tresholdi Otsu mbi imazhin në shkallën gri duke përdorur funksionin cv2.threshold(). Rezultati ruhet në variablat "ret" (vlera e kthyer e funksionit) dhe "thresh" (imazhi i tresholduar). Imazhi i tresholduar shfaqet duke përdorur funksionin plt.imshow() nga biblioteka matplotlib.pyplot, me parametrin cmap='gray' për të shfaqur imazhin në skemën e gri. Pastaj, me funksionin plt.show(), imazhi shfaqet në dritaren e imazhit.

Ky kod shërben si një shembull bazik për tresholdimin e imazheve me tresholdin Otsu duke përdorur OpenCV dhe shfaqjen e rezultatit me Matplotlib. Mund të përdoret si bazë për zhvillimin e aplikacioneve më të avancuara të përpunimit të imazheve.

IN:

```
import math
import numpy as np
import matplotlib.pyplot as plt
```

```

from PIL import Image

def hist(img):
    row, col = img.shape
    y = np.zeros(256)
    for i in range(row):
        for j in range(col):
            y[img[i, j]] += 1
    x = np.arange(256)
    plt.bar(x, y, color='b', width=5, align='center', alpha=0.25)
    plt.show()
    return y

def regenerate_img(img, threshold):
    row, col = img.shape
    y = np.zeros((row, col))
    for i in range(row):
        for j in range(col):
            if img[i, j] >= threshold:
                y[i, j] = 255
            else:
                y[i, j] = 0
    return y

def count_pixel(h):
    cnt = 0
    for i in range(len(h)):
        if h[i] > 0:
            cnt += h[i]
    return cnt

def weight(s, e):
    w = 0
    for i in range(s, e):
        w += h[i]
    return w

def mean(s, e):
    m = 0
    w = weight(s, e)
    for i in range(s, e):
        m += h[i] * i
    return m / float(w)

def variance(s, e):
    v = 0

```



```

m = mean(s, e)
w = weight(s, e)
for i in range(s, e):
    v += ((i - m) ** 2) * h[i]
v /= w
return v

def find_threshold(h):
    cnt = count_pixel(h)
    threshold_values = { }
    for i in range(1, len(h)):
        vb = variance(0, i)
        wb = weight(0, i) / float(cnt)
        mb = mean(0, i)

        vf = variance(i, len(h))
        wf = weight(i, len(h)) / float(cnt)
        mf = mean(i, len(h))

        V2w = wb * (vb) * (vb) + wf * (vf) * (vf)
        V2b = wb * wf * (mb - mf) ** 2

        threshold_values[i] = V2w

    min_V2w = min(threshold_values.values())
    optimal_threshold = [k for k, v in threshold_values.items() if v == min_V2w][0]
    print('Optimal threshold:', optimal_threshold)
    return optimal_threshold

image = Image.open('test.jpg').convert("L")
img = np.asarray(image)

h = hist(img)
op_threshold = find_threshold(h)

result = regenerate_img(img, op_threshold)
plt.imshow(result, cmap='gray')
plt.savefig("otsu.jpg")
plt.show()

```

OUT:

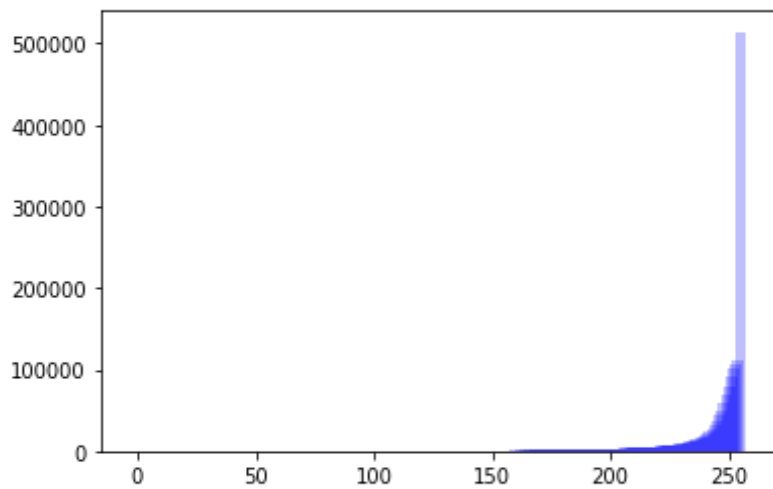


Figura 4.1.2. Outputi 2

optimal threshold 138

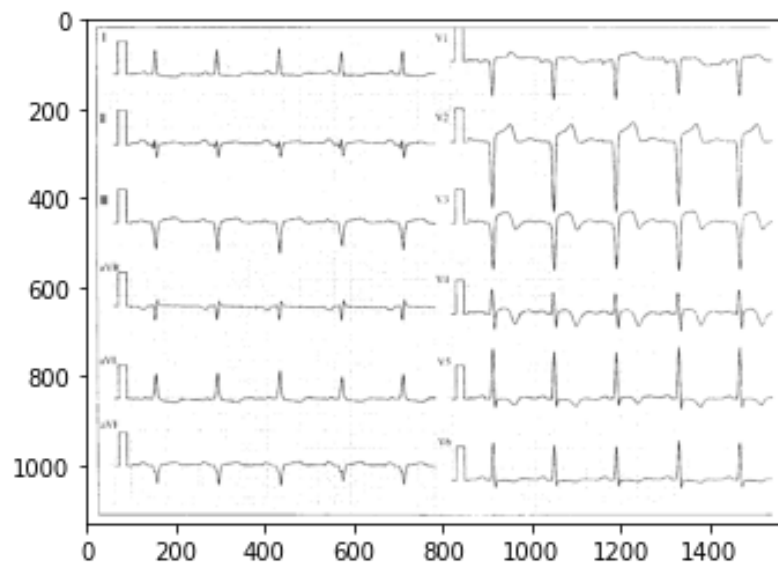


Figura 4.1.3. Outputi 3

Kodi kryen tresholdimin Otsu të një imazhi duke përdorur ndihmën e librave dhe moduleve të ndryshme në Python. Këtu janë disa prej hapat kryesorë të këtij kodi:

Importohen bibliotekat e nevojshme: math për operacione matematikore, numpy për manipulimin e imazheve dhe të dhenave, matplotlib.pyplot për shfaqjen e imazheve dhe PIL (Python Imaging Library) për punën me imazhe.

Definohen disa variabla të nevojshme, siç është "threshold_values" për të ruajtur vlerat e tresholdit, dhe lista "h" me vlerat fillestare.

Përdoret funksioni "Hist" për të llogaritur histogramin e imazhit hyrës. Kjo funksion gjeneron histogramën dhe e shfaq atë në formë grafike përmes funksionit "plt.bar()".

Përdoret funksioni "regenerate_img" për të riprodhuar imazhin e tresholduar. Ky funksion kontrollon vlerën e pikselit në çdo pozicion të imazhit hyrë dhe vendos vlerat e pikselave në imazhin e ri bazuar në tresholdin e dhënë. Përdoren funksionet "countPixel", "weight", "mean" dhe "variance" për të llogaritur vlerat e ndryshme të statistikave të histogramës dhe variancës brenda dhe jashtë klasave të ndarë nga tresholdi.

Përdoret funksioni "threshold" për të gjetur vlerën optimale të tresholdit duke llogaritur variancën e brendshme të klasave për çdo treshold të mundshëm.

Përdoret funksioni "get_optimal_threshold" për të gjetur tresholdin optimal duke zgjedhur tresholdin me variancën më të ulët të brendshme të klasave.

Ngarkohet imazhi hyrës përmes funksionit "Image.open()" nga biblioteka PIL dhe konvertohet në ngjyrat e hirit duke përdorur "convert('L')". Imazhi konvertuar ruhet si një matricë në variablën "img".

Llogaritet histograma e imazhit të konvertuar duke përdorur funksionin "Hist".

Përdoret funksioni "threshold" për të llogaritur tresholdin optimal dhe të ruajë vlerat e tresholdit në "threshold_values".

Përdoret funksioni "get_optimal_threshold" për të gjetur tresholdin optimal nga "threshold_values".

Përdoret funksioni "regenerate_img" për të krijuar imazhin e tresholduar bazuar në tresholdin optimal.

Imazhi i tresholduar shfaqet përmes funksionit "plt.imshow()" dhe ruhet si një imazh të ri në skamën e gri përmes funksionit "plt.savefig()".

IN:

```
import numpy as np
import cv2 as cv
from matplotlib import pyplot as plt

# Supozojmë që 'res' është variabli që mban imazhin që do të shqyrtojmë
img = res

# Konverton imazhin në shkallë gri
gray = cv.cvtColor(img, cv.COLOR_BGR2GRAY)

# Konverton shkallën gri në float64
gray = np.float64(gray)

# Gjenerimi i zhurmës së variancës 25
noise = np.random.randn(*gray.shape) * 10

# Shtojmë zhurmën në imazh
noisy = gray + noise

# Presim vlerat e pikselit midis 0 dhe 255
noisy = np.uint8(np.clip(noisy, 0, 255))
```

```
# Zbaton algoritmin e denoising (Denoising i referohet procesit të heqjes së zhurmës nga një sinjal)
```

```
dst = cv.fastNlMeansDenoising(noisy, None, 3, 7, 21)
```

```
# Shfaq imazhin e denoizuar
```

```
plt.imshow(dst, cmap='gray')
```

```
plt.show()
```

```
plt.imshow(gray, 'gray')
```

OUT:

<matplotlib.image.AxesImage at 0x7fd649a9d5d0>

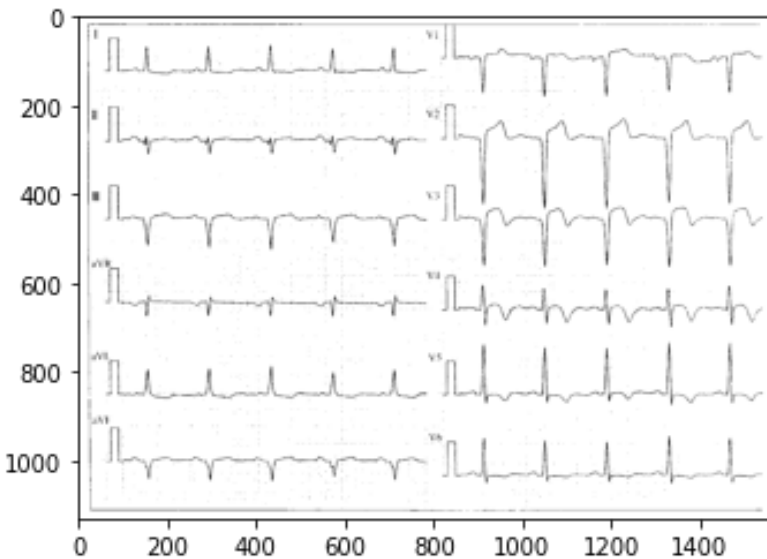


Figura 4.1.4. Outputi 4

IN: plt.imshow(noisy, 'gray')

OUT:

<matplotlib.image.AxesImage at 0x7fd6499cf150>

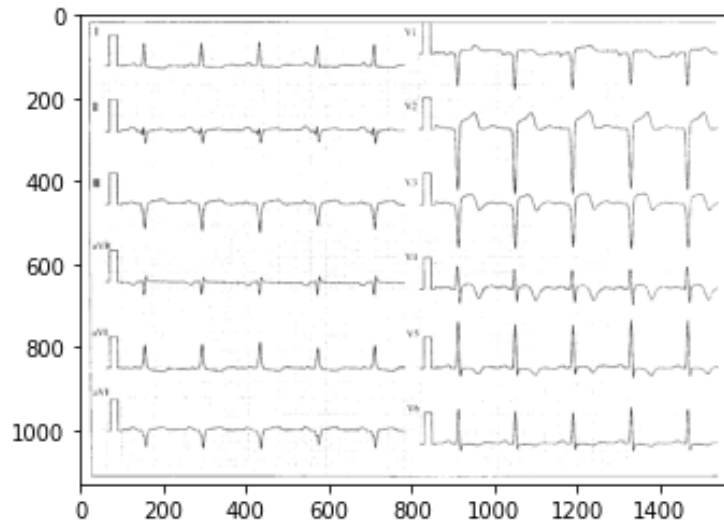


Figura 4.1.5. Outputi 5

IN: plt.imshow(dst,'gray')

OUT:

<matplotlib.image.AxesImage at 0x7fd6367f6c10>

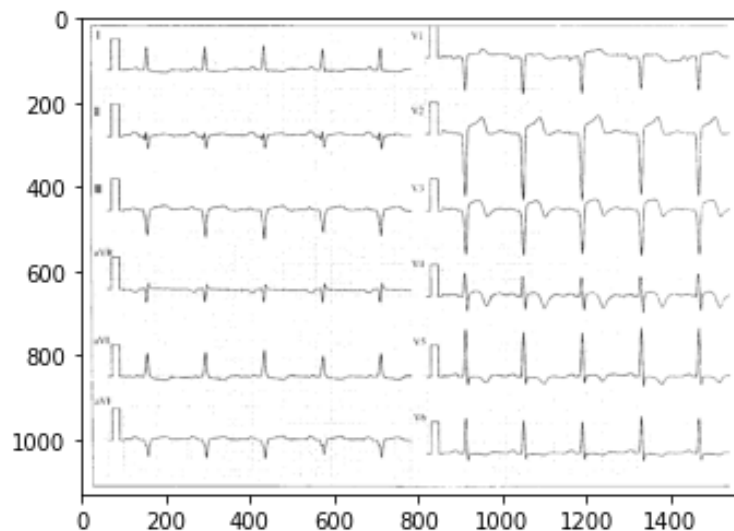


Figura 4.1.6. Outputi 6

Ka disa pjesë të ndryshme të operacioneve të ndërthurura për përpunimin e imazheve. Këtu janë disa nga veprimet që kodi kryen:

Importohen librat e nevojshme: numpy për manipulimin e të dhënave të imazhit, cv2 (OpenCV) për operacionet e përpunimit të imazhit, dhe matplotlib.pyplot për shfaqjen e imazheve.

Vendoset vlera e intensitetit të të gjithë pikselave të një imazhi me madhësi 1130x1566 në vlerën 6. Kjo krijon një imazh tërësisht të bardhë me ndriçim homogjen.

Përdoret funksioni `cv2.cvtColor()` për të konvertuar imazhin në ngjyrat e hirit (BGR) në imazh me ngjyra të bardhë e zi (GRAY).

Përdoret funksioni `cv.fastNlMeansDenoisingColored()` për të zbutur imazhin me ndihmën e denoising algoritmit të OpenCV. Në këtë rast, konvertohet imazhi në BGR për të përdorur funksionin për imazhe me ngjyra.

Përdoret funksioni `np.float64()` për të konvertuar imazhin në tipin e të dhënave `float64`. Kjo është e nevojshme për përpunimin e imazhit duke përdorur ndonjë algoritem që kërkon këtë format të të dhënave.

Gjenerohet një zhurmë me variacion 25 me anë të funksionit `np.random.randn()` dhe shtohet në imazhin fillestar. Ky veprim simulon shtimin e zhurmës në imazh.

Përdoret `np.uint8()` për të konvertuar përsëri imazhin në tipin e të dhënave `uint8`. Kjo bën që vlerat e pixelave të ruhen vetëm në gamën e vlefshmërisë 0-255.

Aplikohet algoritmi i denoising duke përdorur funksionin `cv.fastNlMeansDenoising()`. Këtu, imazhi i zhurmëzuar (noisy) dhe imazhi i rezultuar i konvertuar në BGR (dst) janë hulumtuar për të reduktuar zhurmën duke përdorur algoritmin e denoising të OpenCV.

Nëse pjesa e kodit e zakonshme është hequr nga komentet, do të shfaqen tre subplotë (funksioni `subplot()` në Python) në një figure me Matplotlib. Subploti i parë tregon imazhin origjinal në ngjyrat e hirit (gray), subploti i dytë tregon imazhin me zhurmë (noisy), ndërsa subploti i tretë tregon imazhin e denoised (dst).

IN:

```
# Duke supozuar se "dst" mban imazhin e denoisuar
image = dst

# Crop the image
crop_img = image[15:50+1030, 50:50+1566]

# Shfaq imazhin e prerë
plt.imshow(crop_img, cmap="gray")
plt.show()
```

OUT:

<matplotlib.image.AxesImage at 0x7fd63676a7d0>

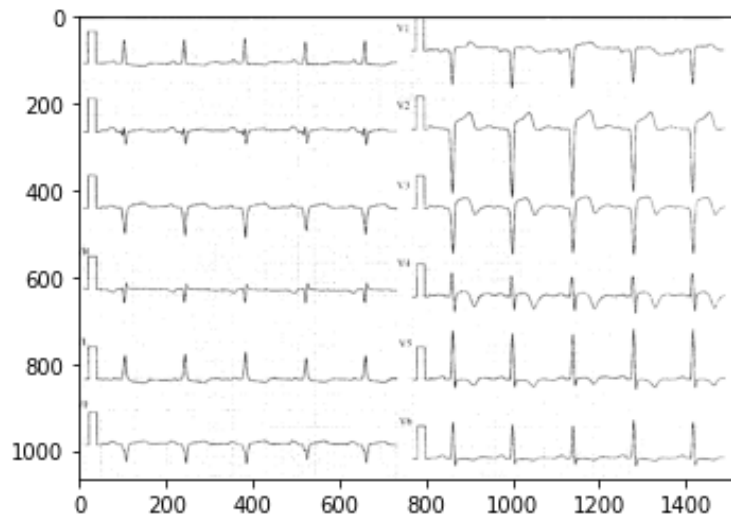


Figura 4.1.7. Outputi 7

IN:

```
import cv2
from matplotlib import pyplot as plt

img = crop_img

for i in range(0, crop_img.shape[0] - 10):
    for j in range(0, crop_img.shape[1] - 10):
        if img[i][j] == 0:
            # Check the surrounding pixels
            count = 0
            for k in range(i - 5, i + 5):
                for l in range(j - 5, j + 5):
                    if img[k][l] == 0:
                        count = count + 1
            if count < 5:
                img[i][j] = 255

plt.imshow(img, cmap="gray")
plt.show()
```

OUT: <matplotlib.image.AxesImage at 0x7fd6366dba50>

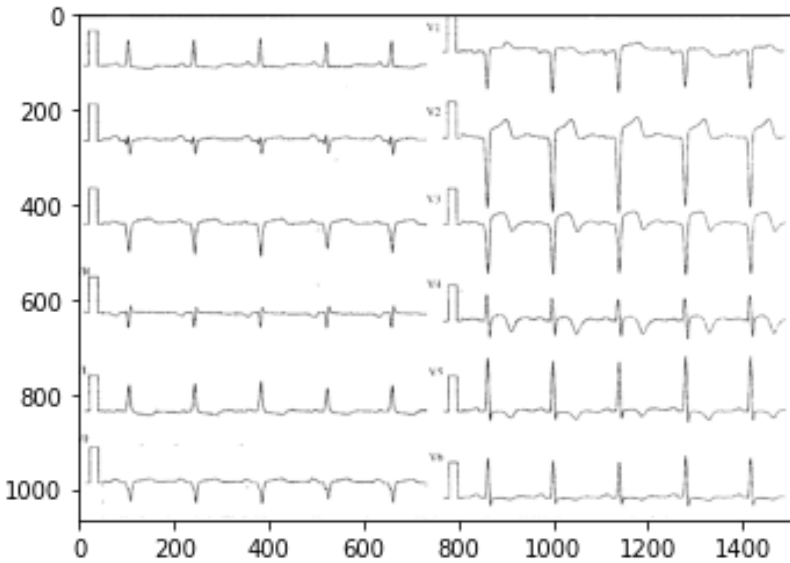


Figura 4.1.8. Outputi 8

Ky kod ka për qëllim të zëvendësojë pikselat e zeza të vogla dhe të izoluara me piksela të bardha në një imazh të bardhë e zi (nga shkalla e ngjyrave të hirit në imazh binar). Procedura kryhet nëpërmjet disa hapa:

Imazhi i caktuar për përpunim ruhet në variablën "crop_img". Ky imazh është i mëparshëm dhe rezulton nga një operacion prerjeje të kryer mbi imazhin origjinal.

Për çdo piksel në imazhin e prerë, kontrollohet nëse vlera e pikselit është 0 (e zezë). Nëse është kështu, bëhet një kontroll i piksela dhe pixelat e fqinjësisë në një zonë prej 10x10.

Nëse më pak se 5 prej piksave fqinje janë të zinj, atëherë piksela aktual ndryshohet nga e zeza në e bardhë (vlera 255).

Pas përpunimit të të gjitha piksave në imazh, imazhi i rezultuar ruhet në variabël "img".

Në fund, imazhi i rezultuar shfaqet përmes funksionit "plt.imshow" për të paraqitur ndryshimet e bëra, ku imazhi i paraqitur është në formën binare (shkalla e ngjyrave të hirit).

IN: `cv2.imwrite('cleaned.jpg',img)`

OUT: TRUE

4.2 DIGITIZING

```
IN: import cv2
IN: import matplotlib.pyplot as plt
IN: i = cv2.imread("cleaned.jpg",0)
IN: plt.imshow(i,cmap="gray")
```

OUT:

<matplotlib.image.AxesImage at 0x7fd636654350>

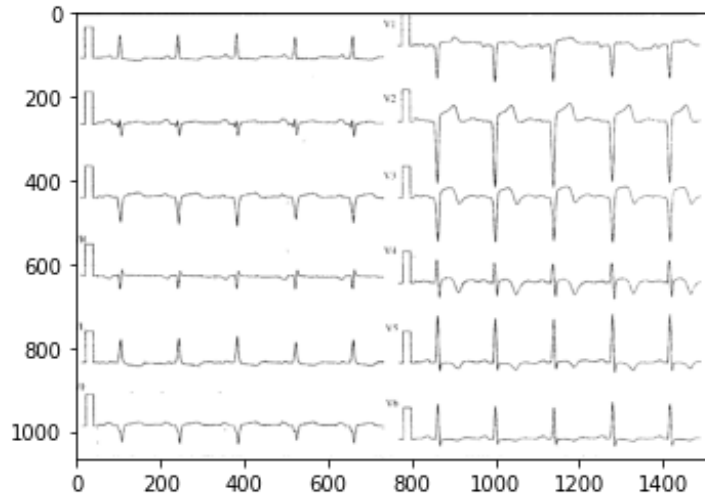


Figura 4.2.1. Outputi 1

```
IN:
print(i)
[[255 255 255 ... 255 255 255]
 [255 255 255 ... 255 255 255]
 [255 255 255 ... 255 255 255]
 ...
 [255 255 255 ... 255 255 255]
 [255 255 255 ... 255 255 255]
 [255 255 255 ... 255 255 255]]
```

```
IN: i.shape
```

OUT: (1065, 1516)

```
IN: crop_img = i[5:125, 0:725]
plt.imshow(crop_img,cmap="gray")
```

OUT:

<matplotlib.image.AxesImage at 0x7fd6365cc250>

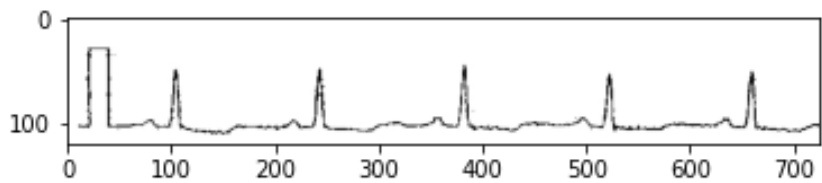


Figura 4.2.2. Outputi 2

IN: cv2.imwrite('crop.jpg',crop_img)

OUT: True

IN: plt.figure(figsize=(120,725))

plt.imshow(crop_img,cmap="gray")

OUT:

<matplotlib.image.AxesImage at 0x7fd636739650>

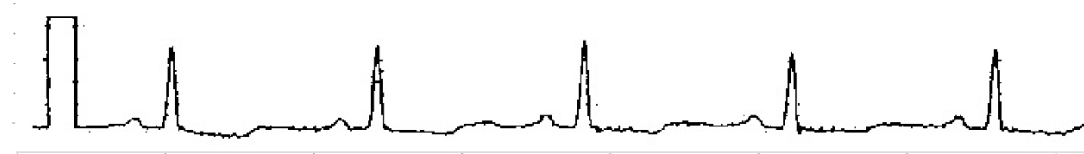


Figura 4.2.3. Outputi 3

IN: img = crop_img[:,75:725]

plt.imshow(img,cmap="gray")

OUT:

<matplotlib.image.AxesImage at 0x7fd636821610>

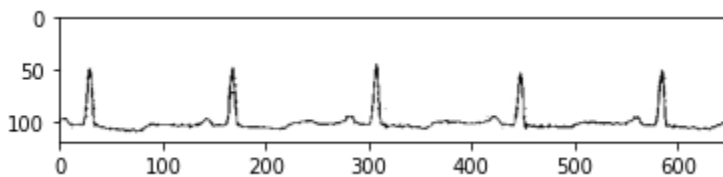


Figura 4.2.4. Outputi 4

IN:

img = crop_img[:,80:725]

plt.figure(figsize=(120,725))

plt.imshow(img,cmap="gray")

OUT:

<matplotlib.image.AxesImage at 0x7fd64973ae50>

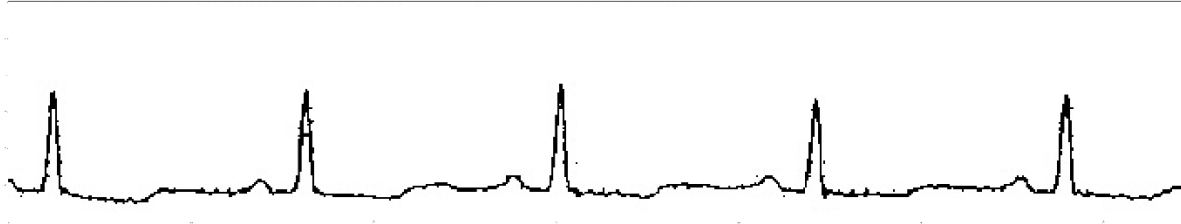


Figura 4.2.5. Outputi 5

IN: img.shape

OUT: (120, 645)

IN:

```
pixel_from_top = []
for i in range(0, 645):
    id = 0
    for j in range(0, 120):
        if img[j][i] == 0:
            pixel_from_top.append(j)
            id = 1
            break

    if id == 0:
        pixel_from_top.append(pixel_from_top[i-1])
```

IN: len(pixel_from_top)

OUT: 645

IN: plt.plot(pixel_from_top)

OUT:

[<matplotlib.lines.Line2D at 0x7fd6498e3bd0>]

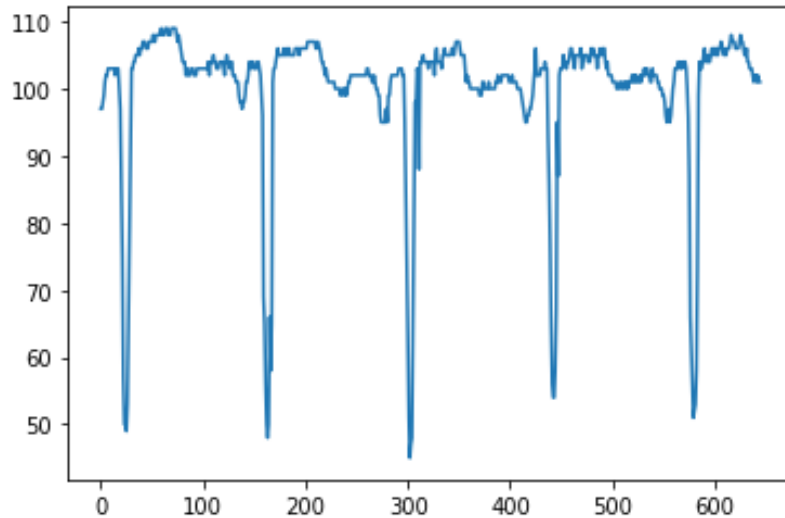


Figura 4.2.6. Outputi 6

Qellimi I ketij kodi eshte të përcaktojë për secilën kolonë të imazhit të cilën rresht fillon pikërisht pikseli i parë i zezë nga lart në poshtë.

Në fillim krijohet një listë bosh e quajtur "pixel_from_top". Pastaj, në një cikël "for" për secilën kolonë të imazhit (645 kolona në total), kontrollohet secila rresht për të gjetur pikselin e parë të zezë. Nëse gjen një piksel të zezë, rreshti i tij (indeksi "j") shtohet në listën "pixel_from_top", dhe variabla "id" vendoset në 1 për të treguar se një piksel është gjetur. Nëse nuk gjen piksel të zezë në kolonë, atëherë shtohet rreshti i pikselit të fundit të zezë të gjetur në kolonën e mëparshme në listën "pixel_from_top".

Pas përfundimit të ciklit, listat "pixel_from_top" përmban rreshtin fillestar të pikselit të parë të zezë për secilën kolonë të imazhit, duke ofruar një informacion mbi vendndodhjen vertikale të këtyre piksleve të zezë në imazh.

```
IN: pixel_from_bottom = []
for i in range(0, 645):
    pixel_from_bottom.append(120 - pixel_from_top[i])

IN: plt.xticks(range(0, 700, 50))
for i in range(0, 645, 140):
    plt.axvline(x=i)
plt.plot(pixel_from_bottom)
```

OUT:

[<matplotlib.lines.Line2D at 0x7fd635ea1550>]

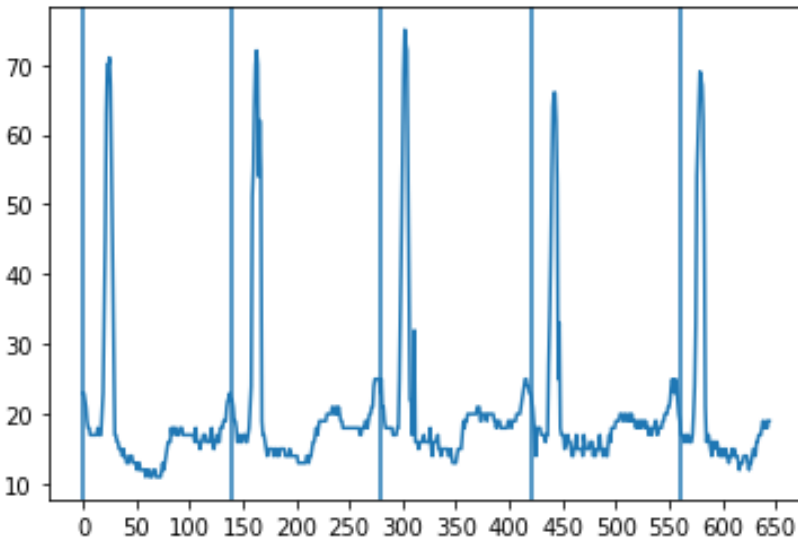


Figura 4.2.7. Outputi 7

Ky kod ka për qëllim të krijojë një grafik për të paraqitur distancën vertikale e pikselit të parë të zezë në secilën kolonë të imazhit nga fundi i imazhit.

Fillimisht, krijohet një listë e re bosh me emrin "pixel_from_bottom". Nëpërmjet një cikli "for", për secilën kolonë të imazhit (645 kolona në total), llogaritet distanca e pikselit të parë të zezë në secilën kolonë nga fundi i imazhit duke iu referuar vlerave të ruajtura në listën "pixel_from_top". Pasi vlerat e distancave janë llogaritur, ato shtohen në listën "pixel_from_bottom" duke u zbritur nga vlera 120 (lartësia e imazhit).

Pas kësaj, përdorohen funksionet e Matplotlib për të konfiguruar grafikun. `plt.xticks()` përcakton shiritat e shënimeve në oksenin x në intervale të rregullta prej 50. Nëpërmjet ciklit "for" tjetër, `plt.axvline()` shton vijat vertikale në intervale prej 140 për të ndarë vizualisht grafikun në seksione. Për fund, përdoret `plt.plot()` për të paraqitur vlerat e listës "pixel_from_bottom" në formën e një grafiku të linjave. Grafiku paraqet distancat vertikale të pikselit të parë të zezë në secilën kolonë të imazhit nga fundi i imazhit.

KREU V

APLIKACIONI

Hyrje

Kodi i dhënë është një shembull i implementimit të disa modeleve të makinës së mësimimit për klasifikimin e të dhënave të zemrës. Në vijim është një hyrje për këtë kod:

*Importimi i librave: Fillimisht, importohen librari të nevojshme si numpy, pandas, matplotlib dhe sklearn për të manipuluar dhe analizuar të dhënat, si dhe për të përdorur modelet e machine learning.

*Shfaqja e skedarëve të dhënave: Kjo pjesë e kodit shfaq rrugën e skedarëve të dhënave në direktorinë e dhënë, që është përdorur për analizë.

*Leximi i të dhënave të zemrës: Dy skedarët e dhënave të zemrës, një për rastet jonormale dhe një për rastet normale, ngarkohen në një strukturë të të dhënave të quajtur DataFrame duke përdorur funksionin `pd.read_csv()`. Të dhënat janë të organizuara në 188 kolona, ku 187 kolonat e para përfaqësojnë atributet e ndryshme të sinjalit të zemrës, ndërsa kolona e fundit përmban vlerën e klasifikimit (0 për rastet normale dhe 1 për rastet e patregullta).

*Shfaqja e informacionit për të dhënat: Pasi të dhënat janë ngarkuar, përdorohen funksionet `info()` dhe `head()` për të shfaqur informacionin rreth strukturës së të dhënave dhe për të shfaqur një pjesë të parë të të dhënave të ngarkuara.

*Analiza e të dhënave të zemrës: Në këtë pjesë, vizualizohen mesataret dhe standard deviation e të dhënave për rastet normale dhe ato të patregullta. Kjo ndihmon të kuptohen dallimet në shpërndarjen e të dhënave mes këtyre dy kategorive.

*Përgatitja e të dhënave për trajnim: Të dhënat e patregullta dhe të rastit normal bashkohen në një DataFrame të vetëm përdorur funksionin `concat()` në bibliotekën pandas. Të dhënat janë ndarë në X (vektorët e tipareve) dhe y (vlerat e klasifikimit). Më pas, të dhënat ndahen në pjesën e trajnimit dhe pjesën e testimit përdorur funksionin `train_test_split()` nga sklearn.

*Trajnim i modelit `KNeighborsClassifier` dhe `RandomForestClassifier`.

Për të bërë matje të saktësisë dhe performancës së një modeli të makinës së mësimimit, ka disa parametra të rëndësishëm që merren në shqyrtim. Këtu janë disa nga parametrat kryesorë që duhet të hynë në analizë:

1) Saktësia (Accuracy): Saktësia përbën një matje të thjeshtë të performancës së modelit dhe përcakton se sa mirë modeli klasifikon saktë rastet e dhëna. Saktësia llogaritet si numri i prediktiveve të sakta të modelit përqindur me numrin total të rasteve të testuara. Parametri i saktësisë është i rëndësishëm për të vlerësuar performancën fillestare të modelit.

2) Matrica e konfuzionit (Confusion Matrix): Matrica e konfuzionit jep një përmbledhje të saktësisë së modelit për secilën klasë të dhënë. Ajo përfshin katër vlera kryesore: True Positive (TP), False Positive (FP), True Negative (TN) dhe False Negative (FN). Me anë të matricës së konfuzionit,

mund të përcaktohet saktësia e ndarjes në mes klasave, precizionin, rënien (recall) dhe f-mjeresimin (F1 score).

3)Precizioni (Precision): Precizioni mat se sa mirë modeli identifikon rastet e vërteta pozitive në mes të të gjitha rasteve që identifikohen si pozitive (TP dhe FP). Kjo matje është e dobishme kur është e rëndësishme të minimizohen rastet e rënies së gabuar.

```
IN[1]:
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

#Shërben për të procesuar dhe analizuar të dhëna nëpërmjet librave të ndihmës të analizës si Num
Py dhe Pandas
#paraqet përdorimin e librarive për të lexuar dhe trajtuar të dhënat në formate të ndryshme si CS
V.

/kaggle/input/heartbeat/ptbdb_abnormal.csv
/kaggle/input/heartbeat/ptbdb_normal.csv
/kaggle/input/heartbeat/mitbih_test.csv
/kaggle/input/heartbeat/mitbih_train.csv
```

```
IN[2]: import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics
#Kodi importon librari të zakonshme të analizës së të dhënave dhe machine learning për të proces
uar dhe analizuar të dhënat në Python.
```

```

IN[3]
abNormalhHeartData = pd.read_csv('/kaggle/input/heartbeat/ptbdb_abnormal.csv', header=None)
normalhHeartData = pd.read_csv('/kaggle/input/heartbeat/ptbdb_normal.csv', header=None)

print('Abnormal heart data information')
abNormalhHeartData.info()

print('Normal heart data information')
normalhHeartData.info()
#Kodi shkruan të dhënat e zemrës normale dhe jonormale nga skedarët CSV dhe pastaj shfaq info
rmacionin për secilin set të të dhënave.

```

```

Abnormal heart data information
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10506 entries, 0 to 10505
Columns: 188 entries, 0 to 187
dtypes: float64(188)
memory usage: 15.1 MB
Normal heart data information
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4046 entries, 0 to 4045
Columns: 188 entries, 0 to 187
dtypes: float64(188)
memory usage: 5.8 MB

```

```

IN[4] abNormalhHeartData.head()
OUT[4]

```



ptbdb_abnormal.csv
v.xlsx

5 rows × 188 columns

```

IN[5] normalhHeartData.head()

```

OUT[5]



ptbdb_normal_csv.x
lsx

5 rows × 188 columns


```

IN[6]
_, (ax1,ax2) = plt.subplots(2,1, figsize=(16,9))

ax1.plot(normalhHeartData.loc[:,186].mean(), label='noraml')
ax1.plot(abNormalhHeartData.loc[:,186].mean(), label='abnoraml')
ax1.set_title("Mean of the data points for heart data")
ax1.legend()

ax2.plot(normalhHeartData.loc[:,186].std(), label='noraml')
ax2.plot(abNormalhHeartData.loc[:,186].std(), label='abnoraml')
ax2.legend()
ax2.set_title("STD for heart data")
# Kodi krijon dy grafikë për të shfaqur mesataren dhe devijimin standard të të dhënave të zemrës
së zakonit dhe të pazakontë.
OUT[6] : Text(0.5, 1.0, 'STD for heart data')

```

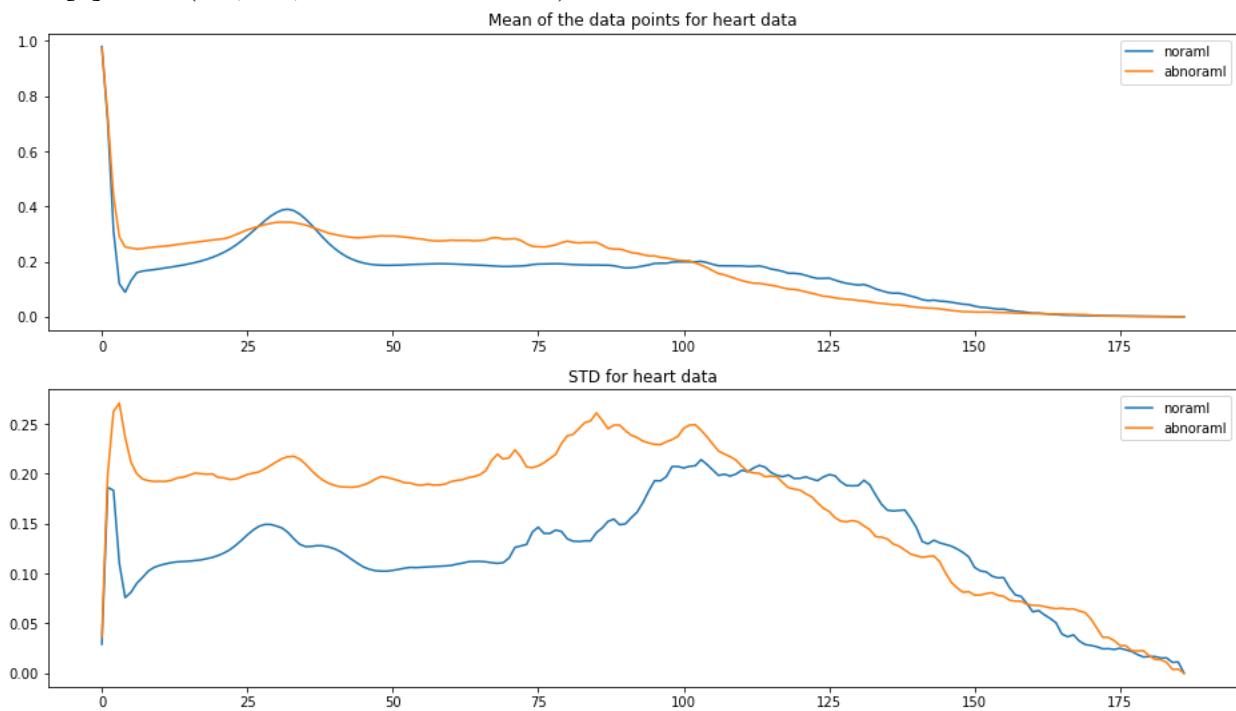


Figura 5.1. Outputi 1

```

IN[7]
all_heart_data = pd.concat([abNormalhHeartData, normalhHeartData], axis=0, ignore_index=True)
# all_heart_data = all_heart_data.sample(frac=1, random_state=0) # shuffle the data

```

```
print(all_heart_data.info())
all_heart_data.head()
#Lidh dhe kombinon të dhënat normale me të dhënat jonormale dhe paraqet 5 rreshtat e parë të të
dhënave të kombinuara
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14552 entries, 0 to 14551
Columns: 188 entries, 0 to 187
dtypes: float64(188)
memory usage: 20.9 MB
None
```



ptbdb_abnormal&n
ormal.csv.xlsx

```
IN[8] all_heart_data.loc[:,187].value_counts()
```

OUT:

```
1.0    10506
```

```
0.0     4046
```

```
Name: 187, dtype: int64
```

```
IN[9]
```

```
X = all_heart_data.loc[:,186]
```

```
y = all_heart_data.loc[:,187]
```

```
# të dhënat e zemrës së kombinuar ndahen në X, dhe y
```

```
IN[10]
```

```
X_train, X_test, y_train, y_test = train_test_split(
```

```
    X, y, test_size=0.3, random_state=0)
```

```
# ku X_train dhe y_train përfaqësojnë të dhënat e trajnimit, ndërsa X_test dhe y_test përfaqësojnë
të dhënat e testimit.
```

5.1 Algoritmi KNeighborsClassifier

KNeighborsClassifier është një algoritëm klasifikimi në machine learning që i përket familjes së algoritmeve të të mësuarit të bazuar në shembull. Është një algoritëm i thjeshtë por efektiv që përdoret si për detyrat e klasifikimit binar ashtu edhe për shumë klasa.

Këtu janë disa pikët kyçe rreth algoritmit KNeighborsClassifier:

*Algoritmi bazohet në parimin që instancat e ngjashme priren t'i përkasin të njëjtës klasë. Ai klasifikon raste të reja duke gjetur k fqinjët më të afërt në grupin e të dhënave të trajnimit dhe duke caktuar etiketën e klasës bazuar në votimin e shumicës.

*Numri i fqinjëve (k) është një hiperparametër që duhet të specifikohet. Ai përcakton se sa fqinjë merren parasysh kur bëhen parashikime.

*Matrica e distancës, si distanca Euklidiane ose distanca e Manhatanit, përdoret për të matur ngjashmërinë midis rasteve.

*Algoritmi nuk përfshin trajnim të qartë. Në vend të kësaj, ai ruan instancat e trajnimit dhe etiketat e tyre përkatëse të klasës në memorie për përdorim të mëvonshëm gjatë parashikimit.

*Vendimi i klasifikimit merret bazuar në etiketat e klasave të k fqinjëve më të afërt. Në rastin e klasifikimit binar, zgjidhet klasa me shumicën e votave. Për klasifikimin me shumë klasa, mund të përdoren strategji të ndryshme, të tilla si one-vs-rest ose one-vs-one.

```
IN[11]
knn_model = KNeighborsClassifier(n_neighbors=5)
knn_model.fit(X_train, y_train)
```

```
OUT[11]KNeighborsClassifier()
```

```
IN[12]
predicted = knn_model.predict([X_test.iloc[0,:]])
print('predict value using the knn model: ', end=")
print(predicted)
print("The true value: ",end=")
print(y_test.iloc[0])
# Kodi përdor modelin e kNN (K-Nearest Neighbors) për të parashikuar një vlerë të re duke përd
orur të dhënat e testim-it të dhënë si input. Rezultati i parashikuar printohet në output, si dhe vler
a e vërtetë që është përdorur për krahasim me rezultatin e parashikuar.
```

```
OUT[12]
predict value using the knn model: [0.]
The true value: 0.0
```

```
IN[13]
print(f'The accuracy of the knn model on train data = {knn_model.score(X_train,y_train)*100:.3}
%')
# Kodi shfaq saktësinë e modelit kNN (K-Nearest Neighbors) në të dhënat e trajnimit në formë të
përqindjes.
```

Saktësia e modelit KNN në të dhënat e trenit = 95.7%

```
IN[14]
print(f'The accuracy of the knn model on test data = {knn_model.score(X_test,y_test)*100:.3}%')
# Kodi shfaq saktësinë e modelit KNN (K-Nearest Neighbors) në të dhënat e testimit në formë të
përqindjes.
```

Saktësia e modelit KNN në të dhënat e provës = 92.2%

5.1.2 Matrica e Konfuzionit

```
IN[15]
predicted = knn_model.predict(X_test)
actual = y_test
confusion_matrix = metrics.confusion_matrix(actual, predicted)

cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix = confusion_matrix, display_labels = [False, True])

cm_display.plot()
plt.show()
# Kodi përdor modelin kNN (K-Nearest Neighbors) për të parashikuar vlerat e testimit dhe më pas krijon matricën e konfuzionit duke krahasuar vlerat e parashikuara me vlerat e vërteta. Pastaj, kodi krijon një vizualizim të matricës së konfuzionit dhe e shfaq në ekran.
```

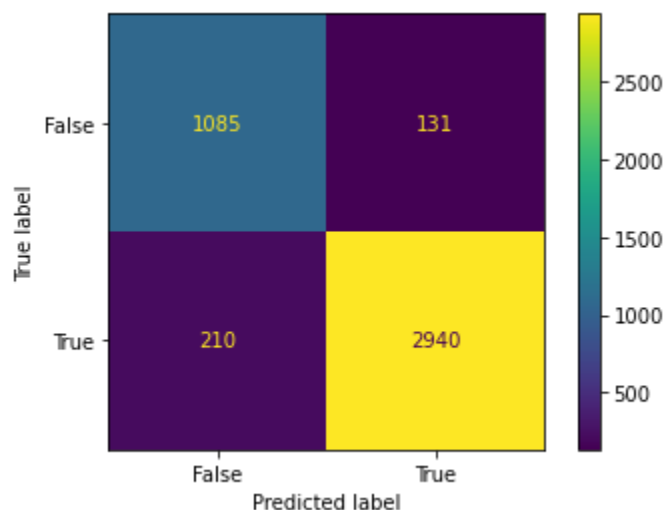


Figura 5.1.2. Outputi 2

5.2 Algoritmi RandomForestClassifier

Random Forest Classifier është një algoritëm në fushën e machine learning të ndarjes së klasave. Ai përdor një teknikë të quajtur "Random Forest" për të krijuar një grup përzgjedhjesh të ndarëve të përbashkët (decision trees). Ai mund të trajtojë grupin e të dhënave që përmban variabla të vazhdueshme, si në rastin e regresionit, dhe variabla kategorikë, si në rastin e klasifikimit. Kryen më mirë për detyrat e klasifikimit dhe regresionit

Këtu janë disa pikët kyçe rreth algoritmit RandomForestClassifier:

*Është një algoritëm i mbështetur në grup që përfshin ndarjen e shumë përzgjedhjesh të pavarura të përbashkët, të njohura si "Random Forest".

*Secila përzgjedhje e ndarë është një model të vetëm i bazuar në pemë vendimmarrëse (decision tree) dhe përdoret për të kryer parashikime.

*Për të përcaktuar klasën e një instance të re, algoritmi merret me shumë përzgjedhje të ndara, dhe pastaj përdor votimin e shumicës për të zgjedhur klasën e pranishme.

*Random Forest përdor teknikat e rastit (randomness) për të zgjedhur një nëngrup të veçantë të karakteristikave dhe instancave për secilën përzgjedhje të ndarë, duke krijuar ndarje të ndryshme të pavarura dhe të diversifikuara.

*Ky algoritëm është i shpërndarë dhe i përshtatshëm për trajnimin paralel, kështu që mund të punojë më shpejt në shkallë të lartë dhe të zgjidhë probleme komplekse të ndarjes së klasave.

*Një avantazh i Random Forest është se është më i qëndrueshëm ndaj rreziqeve të overfitting se një përzgjedhje e ndarë e vetme, pasi merr parasysh votimet e shumë përzgjedhjeve të ndara.

[IN 16]

```
rand_model = RandomForestClassifier()  
rand_model.fit(X_train, y_train)  
# Kodi krijon një model të Random Forest (Rand_Model) dhe e trajton me të dhënat e trajnimit (X_train dhe y_train).
```

OUT[16] RandomForestClassifier()

IN[17] predicted = rand_model.predict([X_test.iloc[0,:]])

```
print('predict value using the knn model: ', end="")
```

```
print(predicted)
```

```
print('The true value: ', end="")
```

```
print(y_test.iloc[0])
```

predict value using the knn model: [0.]

The true value: 0.0

IN[18]

```
print(f'The accuracy of the knn model on train data = {rand_model.score(X_train,y_train)*100:.5}%')
```

```
# Kodi shfaq saktësinë e modelit Random Forest në të dhënat e trajnimit në formë të përqindjes.
```

Saktësia e modelit KNN në të dhënat e trenit = 100.0%

```
IN[ 19]
print(f'The accuracy of the knn model on test data = {rand_model.score(X_test,y_test)*100:.3}%')
# Kodi shfaq saktësinë e modelit Random Forest në të dhënat e testim-it në formë të përqindjes.
```

Saktësia e modelit KNN në të dhënat e provës = 96.8%

```
IN[ 20]
predicted = rand_model.predict(X_test)
actual = y_test
confusion_matrix = metrics.confusion_matrix(actual, predicted)

cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix = confusion_matrix, display_labels = [False, True])

cm_display.plot()
plt.show()
# Kodi përdor modelin Random Forest për të parashikuar vlerat e testim-it dhe më pas krijojë matricën e konfuzionit duke krahasuar vlerat e parashikuara me vlerat e vërteta. Pastaj, kodi krijon një vizualizim të matricës së konfuzionit dhe e shfaq në ekran.
```

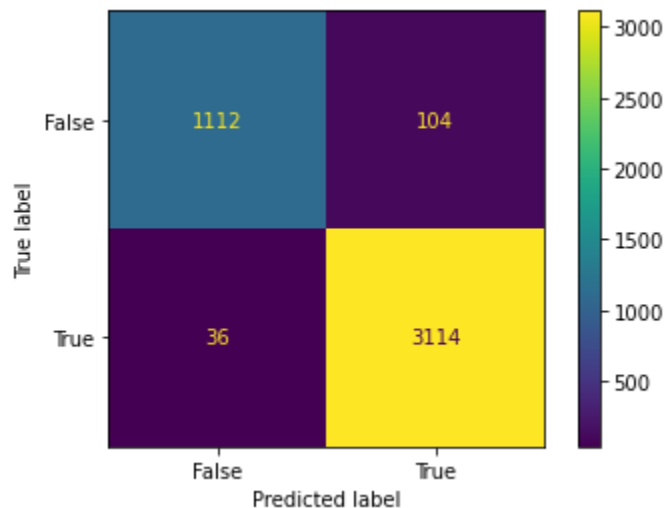


Figura 5.2. Outputi 1

Perfundimi:

Kodi i dhënë paraqet një shembull të thjeshtë të përdorimit të modeleve të machine learning për klasifikimin e të dhënave të zemrës. Disa nga përfundimet që mund të nxirren nga ky kod janë:

- *Importimi i librave: Librat si numpy, pandas, matplotlib dhe sklearn janë importuar për të mbështetur analizën dhe modelimin e të dhënave. Këto librari janë të rëndësishme për të manipuluar të dhënat, vizualizuar ato dhe përdorur modele të machine learning.

- *Ngarkimi dhe shfaqja e të dhënave: Të dhënat e zemrës janë ngarkuar nga skedarët e dhënave dhe janë paraqitur në një formë të përshtatshme të quajtur DataFrame. Ky hap lejon shikimin e parë të të dhënave dhe informacionit rreth tyre.

- *Analiza e të dhënave: Analiza e të dhënave është bërë për të kuptuar dallimet në mesataret dhe standard deviation e të dhënave për rastet normale dhe ato të patregullta. Kjo mund të ndihmojë në identifikimin e karakteristikave të rëndësishme të sinjalit të zemrës që ndihmojnë në dallimin e tyre.

- *Përgatitja e të dhënave për trajnim: Të dhënat janë përgatitur për trajnim duke i ndarë në pjesën e trajnimit dhe pjesën e testimit. Ky hap siguron që modeli të trajnohet në një pjesë të të dhënave dhe të testohet në pjesë të tjera të të dhënave të pavarura.

- *Krijimi i modelit dhe parashikimi: Kemi krijuar modele të KNeighborsClassifier dhe RandomForestClassifier dhe kemi bërë parashikime për të dhënat e testim-it.

- *Vlerësimi i modelit: Kemi llogaritur saktësinë e modeleve të KNeighborsClassifier dhe RandomForestClassifier në të dhënat e trajnimit dhe testim-it.

- *Vizualizimi i matricës së konfuzionit: Kemi krijuar dhe shfaqur një vizualizim të matricës së konfuzionit për të vlerësuar performancën e modelit RandomForestClassifier.

Në përgjithësi, kodet paraqesin një proces të plotë për të përpunuar, analizuar dhe modeluar të dhënat e zemrës, duke përdorur algoritmet e machine learning të K-Neighbors dhe Random Forest Classifier.

Përgjithësisht, ky kod tregon një proces të thjeshtë për përdorimin e modeleve të machine learning për klasifikimin e të dhënave të zemrës. Megjithatë, ky është vetëm një shembull i thjeshtë dhe në përdorimin real, do të ketë nevojë për më shumë rafinim të kodit dhe analizë të thellë të të dhënave për të arritur rezultate më të mira.

REFERENCAT

- [1] "Fakte Mbi Sëmundjet e Zemrës në Shqipëri – Instituti i ... - ISHP." <https://www.ishp.gov.al/fakte-mbi-semundjet-e-zemres-ne-shqiperi/>.
- [2] Greco, L., Percannella, G., Ritrovato, P., Tortorella, F., Vento, M., Trends in IoT based solutions for healthcare: Moving AI to the edge. *Pattern Recognit. Lett.*, 135, pp. 346–353, 2020.
- [3] A. Gupta, S. Yadav, S. Shahid dhe U. Venkanna, "HeartCare: Sistemi i parashikimit të sëmundjeve të zemrës i bazuar në IoT", në *Proc. Int. Konf. Inf. Teknol (ICIT)*, Bhubaneswar, Indi, dhjetor 2019, fq. 88–93. [Online]. E disponueshme: <https://ieeexplore.ieee.org/document/9031924/>
- [4] M. Taştan, "IoT based wearable smart health monitoring system," *Celal Bayar Üniversitesi Fen Bilimleri Dergisi*, vol. 14, no. 3, pp. 343–350, Sep. 2018. [Online]. Available: <http://dergipark.gov.tr/doi/10.18466/cbayarfbe.451076/>
- [5] D. A. Davis, N. V. Chawla, N. Blumm, N. Christakis, and A.-L. Barabasi, "Predicting individual disease risk based on medical history," in *Proc bio-inspired optimization algorithm*, *Int. J. Sci. Eng. Res.*, vol. 4, no. 9, pp. 1978–1986, Jun. 2018.
- [6] *17th ACM Conf. Inf. Knowl. Mining (CIKM)*, Napa Valley, CA, USA, 2008, p. 769. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1458082.1458185>
- [7] M. A. Khan, "An IoT framework for heart disease prediction based on MDCNN classifier," *IEEE Access*, vol. 8, pp. 34717–34727, 2020.
- [8] G. B. Berikol, O. Yildiz, and İ. T. Özcan, "Diagnosis of acute coronary syndrome with a support vector machine," *J. Med. Syst.*, vol. 40, no. 4, p. 84, Apr. 2016.
- [9] M. Taştan, "IoT based wearable smart health monitoring system," *Celal Bayar Üniversitesi Fen Bilimleri Dergisi*, vol. 14, no. 3, pp. 343–350, Sep. 2018. [Online]. Available: <http://dergipark.gov.tr/doi/10.18466/cbayarfbe.451076>
- [10] D. A. Davis, N. V. Chawla, N. Blumm, N. Christakis, and A.-L. Barabasi, "Predicting individual disease risk based on medical history," in *Proc. 17th ACM Conf. Inf. Knowl. Mining (CIKM)*, Napa Valley, CA, USA, 2008, p. 769. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1458082.1458185>
- [11] D. Novak, X. Omlin, R. Leins-Hess, and R. Riener, "Predicting targets of human reaching motions using different sensing technologies," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 9, pp. 2645–2654, Sep. 2013. [Online]. Available: <https://ieeexplore.ieee.org/document/6515157/>

- [12] B. Chen, E. Zheng, X. Fan, T. Liang, Q. Wang, K. Wei, and L. Wang, "Locomotion mode classification using a wearable capacitive sensing system," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 21, no. 5, pp. 744–755, Sep. 2013. [Online]. Available: <http://ieeexplore.ieee.org/document/6517244/>
- [13] F. Tabei, J. M. Gresham, B. Askarian, K. Jung, and J. W. Chong, "Cuff-less blood pressure monitoring system using smartphones," *IEEE Access*, vol. 8, pp. 11534–11545, 2020.
- [14] M. Ganesan and N. Sivakumar, "IoT based heart disease prediction and diagnosis model for healthcare using machine learning models," in *Proc. IEEE Int. Conf. Syst., Comput., Autom. Netw. (ICSCAN)*, Mar. 2019, pp. 1–5
- [15] "Run Data Science & Machine Learning Code Online | Kaggle." <https://www.kaggle.com/code>.
- [16] "GitHub - ritikajha/ECG-Digitization: A tool developed using python to" <https://github.com/ritikajha/ECG-Digitization>.
- [17] "A Review on the Role of Machine Learning in Enabling IoT" 16 Feb. 2021, <https://ieeexplore.ieee.org/document/9355143/>.
- [18] Agrawal, V., Singh, P., Sneha, S., Hyperglycemia prediction using machine learning: A probabilistic approach. International Conference on Advances in Computing and Data Sciences, Springer, Singapore, pp. 304–312, 2019.
- [19] Yan, Z., Zhan, Y., Peng, Z., Liao, S., Shinagawa, Y., Zhang, S., Zhou, X.S., Multi-instance deep learning: Discover discriminative local anatomies for bodypart recognition. *IEEE Trans. Med. Imaging*, 35, 5, 1332–1343, 2016.
- [20] Anthimopoulos, M., Christodoulidis, S., Ebner, L., Christe, A., Mougiakakou, S., Lung pattern classification for interstitial lung diseases using a deep convolutional neural network. *IEEE Trans. Med. Imaging*, 35, 5, 1207–1216, 2016.
- [21] Shen, W., Zhou, M., Yang, F., Yang, C., Tian, J., Multi-scale convolutional neural networks for lung nodule classification. International Conference on Information Processing in Medical Imaging, Springer, Cham, pp. 588–599, 2015.
- [22] Singh, N., Singh, P., Singh, K.K., Singh, A., Diagnosing of Disease using Machine Learning, accepted for ML & Internet of medical things in healthcare, Elsevier Publications (In Press), United States, 2021.
- [23] Schlemper, J., Caballero, J., Hajnal, J.V., Price, A.N., Rueckert, D., A deep cascade of convolutional neural networks for dynamic MR image reconstruction. *IEEE Trans. Med. Imaging*, 37, 2, 491–503, 2017.
- [24] Mehta, J. and Majumdar, A., RODEO: robust DE-aliasing autoencoder for real-time medical image reconstruction. *Pattern Recognit.*, 63, 499–510, 2017.
- [25] Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., Larochelle, H., Brain tumor segmentation with deep neural networks. *Med. Image Anal.*,

35, 18–31, 2017.

[26] Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Lungren, M.P., Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. arXiv preprint arXiv, 1711, 05225, 2017.

[27] Gulshan, V., Peng, L., Coram, M., Stumpe, M.C., Wu, D., Narayanaswamy, A., Kim, R., Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *Jama*, 316, 22, 2402–2410, 2016.

[28] Latif, S., Asim, M., Usman, M., Qadir, J., Rana, R., Automating motion correction in multishot MRI using generative adversarial networks. arXiv preprint arXiv, 1811, 09750, 2018.

[29] Singh, R., Rajesh, E., Prediction of heart disease by clustering and classification techniques Prediction of Heart Disease by Clustering and Classification Techniques. *International Journal of Computer Sciences and Engineering*, 7, 5, pp. 861–866, 2019.

[30] Polat, K. and Güneş, S., Prediction of hepatitis disease based on principal component analysis and artificial immune recognition system. *Appl. Math. Comput.*, 189, 2, 1282–1291, 2007.

[31] Afshar, P., Mohammadi, A., Plataniotis, K.N., Brain tumor type classification via capsule networks. 2018 25th IEEE International Conference on Image Processing (ICIP), IEEE, pp. 3129–3133, 2018.

[32] Zhu, W., Liu, C., Fan, W., Xie, X., Deeplung: Deep 3d dual path nets for automated pulmonary nodule detection and classification. 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, pp. 673–681, 2018.

[33] Wang, Z., Shah, A.D., Tate, A.R., Denaxas, S., Shawe-Taylor, J., Hemingway, H., Extracting diagnoses and investigation results from unstructured text in electronic health records by semi-supervised machine learning. *PLoS One*, 7, 1, e30412, 2012.

[34] Nestor, B., McDermott, M., Boag, W., Berner, G., Naumann, T., Hughes, M.C., Ghassemi, M., Feature robustness in non-stationary health records: Caveats to deployable model performance in common clinical ML tasks. arXiv preprint arXiv, 1908, 00690, 2019.

[35] Anwar, S.M., Majid, M., Qayyum, A., Awais, M., Alnowami, M., Khan, M.K., Medical image analysis using convolutional neural networks: A review. *J. Med. Syst.*, 42, 11, 226, 2018.

[36] Jing, B., Xie, P., Xing, E., On the automatic generation of medical imaging reports. arXiv preprint arXiv, 1711, 08195, 2017.

[37] Wang, X., Peng, Y., Lu, L., Lu, Z., Summers, R.M., Tienet: Text-image embedding network for common thorax disease classification and reporting in chest x-rays. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9049–9058, 2018.

[38] Xue, Y., Xu, T., Long, L.R., Xue, Z., Antani, S., Thoma, G.R., Huang, X., Multimodal recurrent model with attention for automated radiology report generation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer,

Cham, pp. 457–466, 2018.

[39] Weng, S.F., Reps, J., Kai, J., Garibaldi, J.M., Qureshi, N., Can machine- learning improve cardiovascular risk prediction using routine clinical data? *PloS One*, 12, 4, e0174944, 2017.

[40] "Survey of Machine Learning Algorithms for Disease Diagnostic - ResearchGate." https://www.researchgate.net/publication/312629315_Survey_of_Machine_Learning_Algorithms_for_Disease_Diagnostic/fulltext/58909b4ea6fdcc7f21b9717b/Survey-of-Machine-Learning-Algorithms-for-Disease-Diagnostic.pdf.

[41] "The internet of things in healthcare: an overview - ResearchGate." 01 Jan. 2016, https://www.researchgate.net/publication/308517435_The_internet_of_things_in_healthcare_an_overview.

[42] N. Bui and M. Zorzi, "Health care applications: A solution based on the Internet of Things," in *Proc. 4th Int. Symp. Appl. Sci. Biomed. Commun. Technol. (ISABEL)*, New York, NY, USA, 2011, pp. 1–5, doi: 10.1145/2093698.2093829.

[43] N. L. W. S. R. Ginantra, I. G. A. D. Indradewi, and E. Hartono, "Machine learning approach for acute respiratory infections (ISPA) prediction: Case study Indonesia," *J. Phys., Conf. Ser.*, vol. 1469, Feb. 2020, Art. no. 012044, doi: 10.1088%2F1742-6596%2F1469%2F1%2F012044.

[44] S. Challa, M. Wazid, A. K. Das, and M. K. Khan, "Authentication protocols for implantable medical devices: Taxonomy, analysis and future directions," *IEEE Consum. Electron. Mag.*, vol. 7, no. 1, pp. 57–65, Jan. 2018.

[45] F. Ahmed, "An Internet of Things (IoT) application for predicting the quantity of future heart attack patients," *Int. J. Comput. Appl.*, vol. 164, no. 6, pp. 36–40, Apr. 2017.

[46] S. H. Almotiri, M. A. Khan, and M. A. Alghamdi, "Mobile health (m-health) system in the context of IoT," in *Proc. IEEE 4th Int. Conf. Future Internet Things Cloud Workshops (FiCloudW)*, Aug. 2016, pp. 39–42.

[47] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design* (The Pearson Service Technology Series From Thomas Erl). London, U.K.: Pearson, 2005. [Online]. Available: <https://books.google.co.in/books?id=y2MALc9HOF8C>

[48] S.-J. Lee, C.-H. Tseng, G. Lin, Y. Yang, P. Yang, K. Muhammad, and H. M. Pandey, "A dimension-reduction based multilayer perceptron method for supporting the medical decision making," *Pattern Recognit. Lett.*, vol. 131, pp. 15–22, Mar. 2020. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0167865519303484>

[49] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, and P. Liljeberg, "Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach," *Future Gener. Comput. Syst.*, vol. 78, pp. 641–658, Jan. 2018. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0167739X17302121>

- [50] J. Granados, A.-M. Rahmani, P. Nikander, P. Liljeberg, and H. Tenhunen, "Towards energy-efficient healthcare: An Internet-of-Things architecture using intelligent gateways," in *Proc. 4th Int. Conf. Wireless Mobile Commun. Healthcare Transf. Healthcare Through Innov. Mobile Wireless Technol. (ICST)*, Athens, Greece, 2014, pp. 279–282. [Online]. Available: <http://eudl.eu/doi/10.4108/icst.mobihealth.2014.257394>
- [51] N. L. W. S. R. Ginantra, I. G. A. D. Indradewi, and E. Hartono, "Machine learning approach for acute respiratory infections (ISPA) prediction: Case study Indonesia," *J. Phys., Conf. Ser.*, vol. 1469, Feb. 2020, Art. no. 012044, doi: 10.1088%2F1742-6596%2F1469%2F1%2F012044.
- [52] A. Brighente, F. Formaggio, G. M. Di Nunzio, and S. Tomasin, "Machine learning for in-region location verification in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2490–2502, Nov. 2019.
- [53] I. Izonin, A. Trostianchyn, Z. Duriagina, R. Tkachenko, T. Tepla, and N. Lotoshynska, "The combined use of the Wiener polynomial and SVM for material classification task in medical implants production," *Int. J. Intell. Syst. Appl.*, vol. 10, no. 9, pp. 40–47, Sep. 2018.
- [54] F. Ahmed, "An Internet of Things (IoT) application for predicting the quantity of future heart attack patients," *Int. J. Comput. Appl.*, vol. 164, no. 6, pp. 36–40, Apr. 2017.
- [55] I. Azimi, T. Pahikkala, A. M. Rahmani, H. Niela-Vilén, A. Axelin, and P. Liljeberg, "Missing data resilient decision-making for healthcare IoT through personalization: A case study on maternal health," *Future Gener. Comput. Syst.*, vol. 96, pp. 297–308, Jul. 2019.
- [56] T. Hossain, M. A. R. Ahad, T. Tazin, and S. Inoue, "Activity recognition by using LoRaWAN sensor," in *Proc. ACM Int. Joint Conf. Int. Symp. Pervasive Ubiquitous Comput. Wearable Comput.*, Oct. 2018, pp. 58–61.
- [57] A. Gondalia, D. Dixit, S. Parashar, V. Raghava, A. Sengupta, and V. R. Sarobin, "IoT-based healthcare monitoring system for war soldiers using machine learning," *Procedia Comput. Sci.*, vol. 133, pp. 1005–1013, Jan. 2018. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S1877050918310202>
- [58] Z. Yang, Q. Zhou, L. Lei, K. Zheng, and W. Xiang, "An IoT-cloud based wearable ECG monitoring system for smart healthcare," *J. Med. Syst.*, vol. 40, no. 12, p. 286, Dec. 2016. [Online]. Available: <http://link.springer.com/10.1007/s10916-016-0644-9>
- [59] S. K. Sood and I. Mahajan, "Wearable IoT sensor based healthcare system for identifying and controlling chikungunya virus," *Comput. Ind.*, vol. 91, pp. 33–44, Oct. 2017. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0166361516303190>
- [60] J. Kim, S. Lee, G. Lee, Y. Park, and Y. Hong, "Using a method based on a modified K-means clustering and mean shift segmentation to reduce file sizes and detect brain

tumors from magnetic resonance (MRI) images,” *Wireless Pers. Commun.*, vol. 89, no. 3, pp. 993–1008, Aug. 2016. [Online]. Available: <http://link.springer.com/10.1007/s11277-016-3420-8>

[61] S.-B. Cho, “Exploiting machine learning techniques for location recognition and prediction with smartphone logs,” *Neurocomputing*, vol. 176, pp. 98–106, Feb. 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S092523121500569X>

[62] T. Xie, R. Li, X. Zhang, B. Zhou, and Z. Wang, “Research on heartbeat classification algorithm based on CART decision tree,” in *Proc. 8th Int. Symp. Next Gener. Electron. (ISNE)*, Oct. 2019, pp. 1–3.

[63] F. Al Hossain, A. A. Lover, G. A. Corey, N. G. Reich, and T. Rahman, “FluSense: A contactless syndromic surveillance platform for influenza-like illness in hospital waiting areas,” *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 4, no. 1, pp. 1–28, Mar. 2020, doi: 10.1145/3381014.

[64] T. Gupta, V. Nunavath, and S. Roy, “CrowdVAS-Net: A deep-CNN based framework to detect abnormal crowd-motion behavior in videos for predicting crowd disaster,” in *Proc. IEEE Int. Conf. Syst., Man Cybern. (SMC)*, Oct. 2019, pp. 2877–2882.

[65] M. Alhussein, G. Muhammad, M. S. Hossain, and S. U. Amin, “Cognitive IoT-cloud integration for smart healthcare: Case study for epileptic seizure detection and monitoring,” *Mobile Netw. Appl.*, vol. 23, no. 6, pp. 1624–1635, Dec. 2018. [Online]. Available: <http://link.springer.com/10.1007/s11036-018-1113-0>

[66] “A fully-automated paper ECG digitisation algorithm using deep ... - Nature.” 05 Dec. 2022, <https://www.nature.com/articles/s41598-022-25284-1>.