Primjena senzora s ugrađenim strojnim učenjem za detekciju stanja

Vujasinović, Hrvoje

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Hrvoje Vujasinović

PRIMJENA SENZORA S UGRAĐENIM STROJNIM UČENJEM ZA DETEKCIJU STANJA

DIPLOMSKI RAD

Varaždin, 2025.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Hrvoje Vujasinović

Matični broj: 0231042668

Studij: Informacijsko i programsko inženjerstvo

PRIMJENA SENZORA S UGRAĐENIM STROJNIM UČENJEM ZA DETEKCIJU STANJA

DIPLOMSKI RAD

Mentor:

Prof. dr. sc. Danijel Radošević

Varaždin, siječanj 2025.

Hrvoje Vujasinović

Izjava o izvornosti

Izjavljujem da je moj diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi

_____

Sažetak

U radu će se demonstrirati upotreba LSM6DSOX senzora s ugrađenim strojnim učenjem za detekciju stanja. Poslovni slučaj vezan je uz detekciju stanja poklopca cisterne kamiona, gdje se postojeći uređaj ugrađuje na cisternu, po jedan uz svaki poklopac za utakanje goriva, kako bi se mogao vršiti nadzor otvaranja i zatvaranja poklopaca. Na sam poklopac ugrađuje se magnet, te je u zatvorenom stanju poklopca vrijednost magnetskog polja značajno veća od uobičajene vrijednosti zemljinog polja. Ovo predstavlja bazu za metodu detekcije statusa poklopca. Funkcija samog senzora je očitavanje podataka o magnetskom polju u sve tri osi s velikom brzinom, te slanje tih podataka do kontrolnog uređaja. No uređaj koji koristi strojno učenje mogao bi se koristiti za detekciju bilo koje druge aktivnosti u raznim industrijama npr. praćenje gibanja vozila, praćenje brodova, detekcija vibracija/nagiba u industrijskim postrojenjima ili bilo koje druge aktivnosti koja se može detektirati s pomoću ugrađenog žiroskopa, akcelerometra i 6-osnog senzora. Za izvedbu uređaja koristit će se STEVAL-MKSBOX1V1 komplet za razvoj (eng. *devkit*) koji sadrži STM32 mikroprocesor i LSM6DSOX senzor. U radu će se osim teorijske osnove vezane uz internet stvari, strojno učenje i opis poslovnog slučaja prikazati i razvoj aplikacije za detekciju stanja upotrebom C programskog jezika te svih popratnih alata koji su dostupni za navedeni komplet za razvoj i senzor od strane proizvođača. Programsko rješenje obuhvatit će aplikaciju (eng. *firmware*) bazirano na RTOS okruženju i vanjsku aplikaciju za parsiranje i vizualizaciju podataka sa senzora.

Ključne riječi: embedded; iot; STM32; RTOS; strojno učenje

Sadržaj

1. Uvod

Internet stvari (eng. *Internet of Things*) revolucionizirao je način na koji uređaji međusobno djeluju na fizički svijet, transformirajući razne industrije i način na koji funkcioniraju potrošači domovi. Internet stvari olakšava razmjenu podataka između međusobno povezanih uređaja, omogućujući neviđene razine automatizacije i uvida u stvarni svijet. U ovom radu istražit će se različiti aspekti interneta stvari počevši od njegove povijesti i evolucije, do pregleda trenutačnih primjena i tehnologija koje ga pokreću. Obradit će se i uloga strojnog učenja unutar interneta stvari, ilustrirajući kako strojno učenje i umjetna inteligencija poboljšavaju donošenje odluka i omogućuju pametnije uređaje.

U teorijskom dijelu rada upoznat ćemo temelje interneta stvari, prednosti i nedostatke, moderne komunikacijske tehnologije na kojima se temelji kao što su Zigbee, Wi-Fi, NFC i druge, moderne IoT platforme poput Microsoft Azure IoT, AWS IoT Core, Node-RED, i Home Assistant, te popularne mikrokontrolere na kojima se baziraju IoT uređaji poput STM32 i ESP32. Opisat će se i pojam strojnog učenja, njegova povijest i primjena te neki popularni algoritmi koji omogućuju strojno učenje.

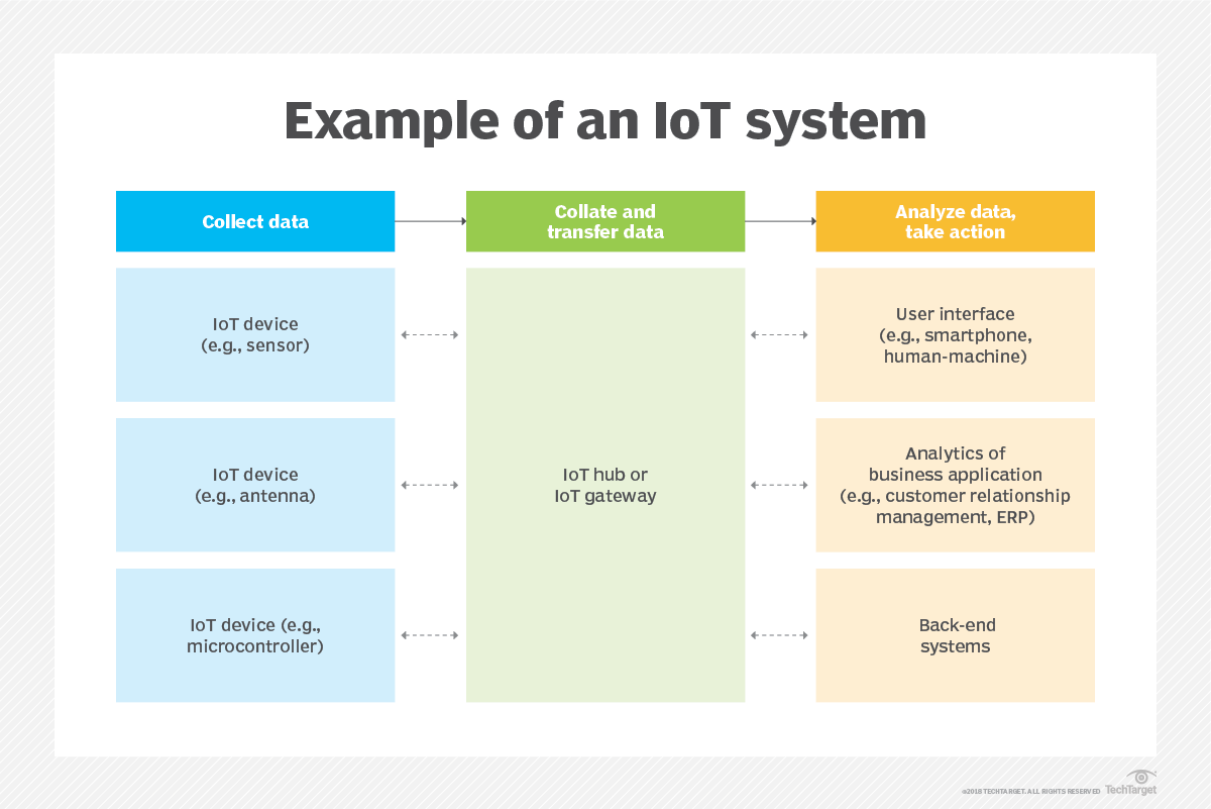
Praktični dio rada usredotočit će se na rješavanje poslovnog slučaja i razvoj prototipa uređaja za detekciju stanja s pomoću kompleta za razvoj koji sadrži senzor s ugrađenim strojnim učenjem. Bit će prikazan razvoj aplikacije za detekciju stanja upotrebom C++ programskog jezika i svih popratnih alata koji su dostupni za navedeni komplet za razvoj. Programsko rješenje obuhvatit će aplikaciju baziranu na RTOS okruženju i vanjsku aplikaciju za parsiranje i vizualizaciju podataka sa senzora. Ovakav uređaj bi kroz proces strojnog učenja analizirao obrasce promjena u kutnoj brzini, ubrzanju i orijentaciji senzora kako bi prepoznao specifične uzorke. Nakon završetka treniranja, senzor bi mogao autonomno detektirati otvaranje i zatvaranje poklopca u stvarnom vremenu, koristeći algoritme za prepoznavanje uzoraka, bez potrebe za dodatnim vanjskim komponentama. Ovakvo rješenje smanjuje kompleksnost hardvera, omogućuje fleksibilnije prilagođavanje različitim dizajnima i uvjetima primjene. Ovakav uređaj mogao bi se koristiti za detekciju bilo koje druge aktivnosti u raznim industrijama.

2. Internet stvari

Pojam internet stvari odnosi se na mrežu međusobno povezanih uređaja koji komuniciraju međusobno ili razmjenjuju podatke preko interneta s udaljenim serverima. Takvi uređaji obično imaju ugrađene senzore i programsku podršku za prikupljanje podataka iz okoline, te mogu varirati od mehaničkih i digitalnih strojeva pa sve do svakodnevnih potrošačkih proizvoda. Zahvaljujući internetskoj vezi podaci se mogu prenositi preko mreže i obrađivati bez potrebe za ljudskom interakcijom. Uređaji se obično povezuju na zajednički mrežni pristupnik (eng. *Gateway*) koji ima ulogu centralnog mjesta na koje se šalju prikupljeni podaci sa senzora. Nakon prikupljanja podaci se mogu obraditi lokalno ili se šalju u oblak na daljnju analizu kako bi bili dostupni korisnicima sustava. Lokalna obrada i analiza podataka ima prednost jer se smanjuje količina podataka koja se šalje na obradu što smanjuje potrebu za propusnosti mreže, ali zahtjeva više procesorske snage na samome uređaju. Podaci se koriste kako bi se prepoznali uzorci i na temelju njih donosile odluke, prijedlozi ili uočili potencijalni problemi u radu sustava. Često na temelju informacija povezanih senzora uređaji djeluju samostalno kako bi regulirali sustav ili obavili neki rad bez intervencije čovjeka iako i ljudi mogu svojom interakcijom utjecati na ponašanje sustava. To se obavlja preko grafičkog sučelja koje se koristi za interakciju i upravljanje IoT uređajima u obliku web stranice ili mobilne aplikacije za kontrolu, upravljanje, i dodavanje novih uređaja u mrežu. IoT pruža ljudima jednostavniji život i rad omogućujući pametnija i učinkovitija rješenja. Primjerice, svakodnevni kućanski uređaji poput pametnih svjetla i termostata mogu automatizacijom postaviti temperaturu doma i prigušiti svjetla po završetku dana ili kad ne detektiraju prisutnost čovjeka. Osim automatizacije doma IoT ima i veliku primjenu u poslovnom svijetu i industriji. Organizacijama u realnom vremenu pruža uvid u rad sustava obuhvaćajući cjelokupno poslovanje od ponašanja strojeva u radu pa sve do cjelokupnog opskrbnog lanca. Neke od prednosti IoT-a vezane su za određene industrije, ali većina njih ima široku primjenu neovisno o industriji. IoT je naročito raširen u proizvodnji, transportu i komunalnim uslugama gdje se senzori i IoT uređaji široko koriste, te također ima značajne primjene u poljoprivredi, infrastrukturi i kućnoj automatizaciji potičući digitalnu transformaciju u svim industrijama. [1]

U poljoprivredi IoT uređaji se mogu koristiti kako bi se mjerila količina padalina, vlažnost i temperatura zraka, optimizirali procesi upravljanja stokom, automatizirale tehnike uzgoja, nadzirala oprema i uređaji, ili pak pojednostavilo upravljanje opskrbnim lancem. IoT tehnologije imaju i transformativnu ulogu u pametnim gradovima i urbanim sredinama gdje pomažu pri smanjenju otpada i potrošnje energije korištenjem senzora za optimizaciju potrošnje energije, kao što je prilagodba klimatizacije na temelju zauzetosti sobe ili smanjenje grijanja kada su uredi prazni. Urbani sustavi imaju koristi od interneta stvari kroz poboljšano upravljanje

semaforima, parkirnim satovima, odlaganjem otpada i mrežama javnog prijevoza. Osim toga, u građevinarstvu IoT senzori mogu nadzirati infrastrukturu poput zgrada i mostova radi strukturalnih promjena, povećavajući sigurnost, operativnu učinkovitost i kvalitetu usluge. Nosivi IoT uređaji revolucionarizirali su osobnu i javnu sigurnost. U zdravstvu, IoT uređaji kao što su sustavi daljinskog nadzora, pametni medicinski alati i uređaji za praćenje lijekova omogućuju pružateljima usluga da pažljivo prate pacijente, upravljaju kroničnim stanjima i odrade pravovremene intervencije. Bolnice također koriste IoT za upravljanje zalihama, pojednostavljujući rukovanje lijekovima i medicinskim instrumentima. [1]



Slika 1. Primjer arhitekture IoT sustava (Izvor: [1])

Posljednjih godina IoT postao je jedna od najtransformativnijih tehnologija 21. stoljeća. Povezivanjem svakodnevnih predmeta poput kućanskih uređaja, automobila, termostata, lampi s internetskom vezom, neprekidna komunikacija između ljudi, procesa i uređaja postala je svakodnevica. Korištenjem jeftinih i pristupačnih uređaja, usluga u oblaku, podataka, analitike i mobilne tehnologije, fizički objekti sada mogu prikupljati i dijeliti podatke uz minimalnu ljudsku intervenciju. U ovom hiperpovezanom svijetu, digitalni sustavi mogu snimati, nadzirati i optimizirati interakcije između povezanih uređaja. Takav spoj fizičkog i digitalnog svijeta omogućuje im skladan zajednički rad. Najveći utjecaj na ubrzani razvoj IoT-a imao je razvoj pouzdanih senzorskih tehnologija niske cijene i male potrošnje energije, mnoštvo mrežnih protokola koji omogućuju povezivanje s oblakom i drugim uređajima za učinkovit

prijenos podataka, porast dostupnih platformi za računalstvo u oblaku i dostupne infrastrukture, te veliki napredak na područjima strojnog učenja i umjetne inteligencije koji se koriste za obradu i analizu velikih količina podataka kako bi se prepoznali uzorci i proizvele informacije za bolje donošenje odluka. [2]

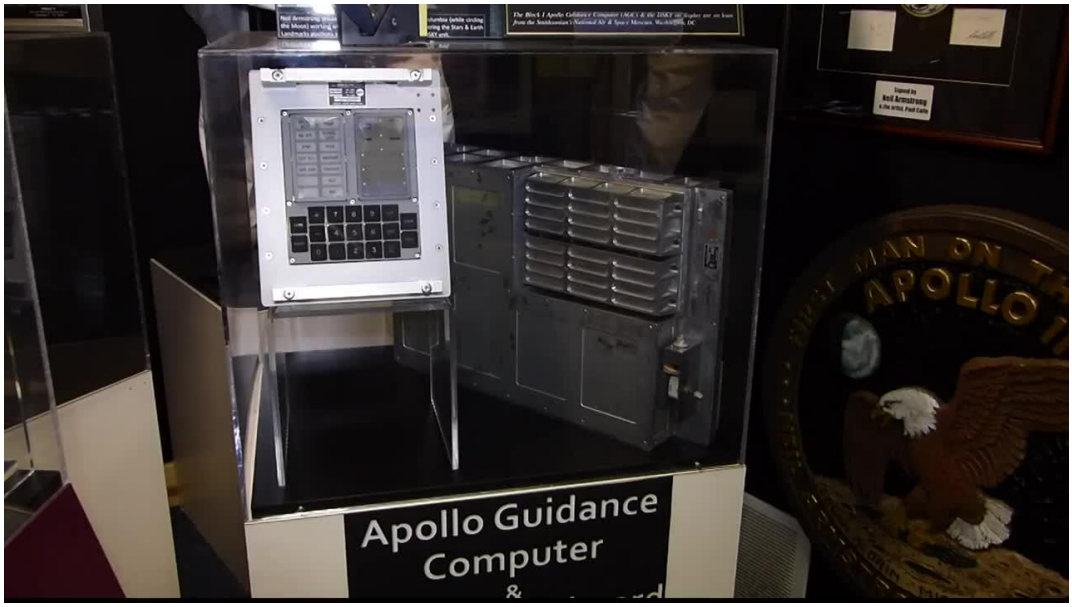
Prema [1] neke od prednosti IoT-a su njegova pristupačnost jer omogućuje jednostavan pristup informacijama bilo kada, bilo gdje i na bilo kojem uređaju u realnom vremenu preko sučelja prilagođenih korisniku ili proaktivnih obavijesti, poboljšanje komunikacije među povezanim elektroničkim uređajima jer omogućava učinkovitu razmjenu podataka, širenjem mrežne pokrivenosti, uštedom energije i davanjem prioriteta bitnim informacijama. Na primjer, u sustavu pametne kuće, senzor pokreta koji detektira aktivnost na ulaznim vratima može poslati upozorenje sustavu pametne rasvjete da aktivira vanjska svjetla. Omogućava prijenos podataka preko povezane mreže uređaja, što dovodi do uštede vremena i troškova. Primjer za to je prediktivno održavanje u industrijskim okruženjima. IoT senzori ugrađeni u strojeve prate parametre poput temperature, vibracija i radnih uvjeta u stvarnom vremenu. Podaci prikupljeni s ovih senzora analiziraju se algoritmima strojnog učenja kako bi se uočili obrasci koji ukazuju na moguće greške ili pad performansi, čime se u konačnici štedi vrijeme i novac. Može se koristiti za poboljšanje opskrbnog lanca i upravljanje zalihama, omogućujući proizvođačima smanjenje troškova i povećanje zadovoljstva kupaca. Praćenjem robe i materijala u stvarnom vremenu, proizvođači mogu pratiti razine zaliha, minimizirati višak zaliha i pojednostaviti logističke operacije. Lokalna obradu podataka smanjuje potrebu za slanjem velikih količina podataka u oblak. To omogućuje fizičkim uređajima učinkovitiju komunikaciju i rukovanje podacima na licu mjesta te dijeljenje samo relevantnih informacija s drugim uređajima ili uslugama u oblaku pa se tako povećava učinkovitost. Omogućuje automatizaciju zadataka kako bi poboljšao kvalitetu usluge i smanjio potrebu za ljudskom intervencijom. Na primjer, u poljoprivredi, sustavi za navodnjavanje koje pokreće IoT mogu automatski prilagoditi rasporede navodnjavanja na temelju podataka kao što su vlažnost tla, vremenski uvjeti, temperatura i potrebe usjeva. Olakšava stvaranje personaliziranih proizvoda i usluga koje zadovoljavaju individualne preferencije i potrebe korisnika. Primjeri za to uključuju pametne kućne uređaje, nosivu tehnologiju kao što su pametni satovi i prilagođene preporuke u maloprodaji, što sve poboljšava cjelokupno korisničko iskustvo. IoT sustavi su iznimno fleksibilni i mogu se skalirati kako bi se zadovoljile rastuće potrebe poslovanja, na primjer dodavanje novih uređaja, proširenje funkcija ili integracija s postojećim sustavima. Zbog velike količine podataka koju prikupljaju IoT uređaji moguće je analizom doći do uvida u ponašanje kupaca i tržišta što dovodi do donošenja boljih poslovnih odluka. Tvrtke mogu donositi informirane odluke na temelju informacija, optimizirati procese i otkriti nove prilike za povećanje prihoda. Samim time učinkovito korištenje resursa pomaže smanjiti utjecaj na okoliš

kroz inicijative kao što su pametno upravljanje energijom, smanjenje otpada i održive poljoprivredne prakse. Optimiziranjem korištenja resursa i smanjenjem otpada, IoT igra ključnu ulogu u podržavanju održivosti okoliša.

Međutim [1] navodi i neke izazove i rizike koji se javljaju kod IoT sustava kao što su sigurnosni rizici zbog povećanja broja povezanih uređaja čime raste i mogućnost napada povećavajući potencijal za sigurnosne provale. Zbog količine podataka koje se prikupljaju i dijele između uređaja velika je vjerojatnost da napadači dobiju pristup osjetljivim informacijama. Velika količina podataka koja se prikuplja prisiljava proizvođače i korisnike da se pridržavaju zakonskih regulativa o prikupljanju, zaštiti, privatnosti i obradi podataka koji se mogu bitno razlikovati ovisno o zemlji. Kako broj IoT uređaja raste, upravljanje njima postaje sve veći izazov. Korisnici se mogu suočiti s problemom rukovanja velikom količinom uređaja, što otežava prikupljanje, organiziranje i upravljanje podacima. Oštećeni uređaji ili uređaji s greškom u radu mogu kompromitirati druge uređaje povezane s internetom, ako postoji ranjivost u sustavu može doći do raširenog problema ili napada na sustav. Zbog nepostojanja univerzalnog standarda za interoperabilnosti mogu se javiti problemi kod povezivanja različitih IoT platformi. Različite platforme, protokoli i standardi razvijaju se neovisno jedni o drugima pa je integracija i komunikacija između različitih sustava i uređaja otežana. Smanjena potreba za ljudskim sudjelovanjem u mnogim zadacima zbog IoT-a može dovesti do smanjenja potrebe za radnom snagom, osobito za niskokvalificirane radnike. Na primjer, automatizacija u upravljanju zalihama i raširena uporaba bankomata smanjili su potražnju za fizičkim radom, što je rezultiralo gubitkom radnih mjesta i povećanom nesigurnošću posla za pojedince.

2.1. Povijest

Koncept IoT-a možemo prepoznati već 1960-ih kada se razvijalo računalo za navođenje Apollo rakete koje je pokazalo potencijal za međusobno povezane sustave za prikupljanje, obradu i prijenos podataka. Ovakvi rani eksperimenti u integraciji hardvera i softvera postavili su temelje za sustave uređaja koji mogu međusobno komunicirati u stvarnom vremenu. Računalo ACG (eng. *Apollo Guidance Computer*) bilo je razvijano u MIT instrumentacijskom laboratoriju dok je projekt vodio Charles Stark Draper. ACG bilo je veliko računalo teško oko 30 kg te je sadržavalo preko 4000 integriranih sklopova. Iako je računalna snaga tog sustava bila mala prema današnjim standardima postavljeni su principi automatizacije, obrade u stvarnom vremenu, i pouzdanosti koji su kritični za uspješan rad međusobno povezanih uređaja. Time je pokazan ogroman potencijal za umrežene tehnologije za kontrolu i upravljanje složenim operacijama na daljinu, inspirirajući napredak u senzorskim i komunikacijskim tehnologijama na kojima se moderan IoT temelji. Nakon uspješne implementacije u misijama u svemiru znanstvenici su krenuli s istraživanjem primjena takvih sustava na Zemlji. [3] [4]



Slika 2. Računalo za navođenje Apollo rakete - AGC (Izvor: [4])

Već 1982. godine pojavio se prvi koncept pametnog uređaja povezanog na mrežu. Na Sveučilištu Carnegie Mellon instalirani su senzori u aparat za Coca-Colu s pomoću kojih su studenti mogli pratiti temperaturu i stanje zaliha, tako je taj aparat postao prvi uređaj povezan na ARPANET, preteču današnjeg interneta. [5]

Koncept interneta stvari prvi je predstavio Peter T. Lewis u svojem govoru Kongresnoj zakladi *Black Caucus* 1985. godine. On je IoT opisao kao integraciju ljudi, procesa i tehnologije

s povezanim uređajima i senzorima, što omogućuje udaljeno praćenje, manipulaciju i analizu trendova na uređajima. Neovisno o njegovom radu, Kevin Ashton skovao je i sam pojam interneta stvari 1999. godine kao zaposlenik poduzeća Procter&Gamble, kasnije i zaposlenik Auto-ID centra na MIT-u. On je u početku IoT povezivao s upotrebom radiofrekvencijske tehnologije (eng. *RFID*) predviđajući sustav u kojem bi računala mogla samostalno upravljati pojedinačnim objektima. Radio je na problemu upotrebe specijalnih čipova kako bi se pratila lokacija proizvoda u opskrbnom lancu. Glavna ideja bila je ugraditi mobilne odašiljače kratkog dometa u različite uređaje i tako omogućiti komunikaciju između ljudi i uređaja, i komunikaciju među samim uređajima. Integracija RFID tehnologije u IoT bila je posebno značajna jer je uređajima omogućila prijenos informacija o svom statusu i lokaciji što čini temelj današnjih pametnih sustava. [5] [6]

John Romkey je 1990. godine demonstrirao internetski toster, prvi uređaj kojim se moglo upravljati putem interneta. Iako je bio jednostavan, mogao se isključiti i uključiti samo daljinski i simbolizirao je značajan korak u premošćivanju fizičkog i digitalnog svijeta. Do 2000. godine došlo je do razvoja bitnih tehnologija usko povezanih s internetom stvari. Američka vlada završila je prvu verziju GPS sustava i donesen je nacrt standarda za IPv6 adresiranje koji omogućuje povezivanje većeg broja uređaja na internet nego što je ranije dopuštao IPv4 standard. Kompanija LG je 2000. godine lansirala hladnjak povezan na internet koji je potrošačima omogućavao praćenje namirnica i predlaganje recepata na temelju dostupnih namirnica. Već 2008. godine broj povezanih uređaja bio je veći od svjetske populacije. Osim aplikacija namijenjenih potrošačima, rani IoT sustavi također su se razvijali za industrijsku upotrebu. Na primjer, Siemens je 1990-ih predstavio komunikacijske module, omogućujući industrijskoj opremi dijeljenje podataka preko GSM mreža. Ove su inovacije istaknule svestranost IoT-a u osobnom i poslovnom kontekstu, potaknuvši interes u nizu industrija, od kućne automatizacije do proizvodnje. [6] [7]

Prva internacionalna IoT konferencija održana je 2008. godine u Švicarskoj, a već sljedeće godine Google je počeo testirati autonomna vozila. 2010. godine Kineska vlada proglasila je IoT ključnom tehnologijom dok je kompanija Nest na tržište izbacila prvi pametni termostat koji je s pomoću algoritama strojnog učenja mogao upravljati grijanjem i hlađenjem ovisno o navikama korisnika te je tako predstavila koncept pametne kuće. Od 2010. do danas pojavile su se brojne druge primjene IoT-a u svakodnevnom životu poput uređaja Google Glass, Amazon Echo, brojnih primjena kod autonomnih vozila pa je tako došlo i do prvih sigurnosnih incidenata kao što je bio Mirai botnet napad na IoT uređaje zbog slabe sigurnosti. Industrijski IoT (skraćeno *IIoT*) napravio je revoluciju u proizvodnji optimizacijom troškova, povećanjem sigurnosti na radnom mjestu i poboljšanjem kvalitete proizvoda. Podaci u stvarnom vremenu iz IoT senzora pružaju menadžerima sveobuhvatan uvid u stanje opreme,

omogućujući bolje poslovno planiranje. Prediktivno održavanje iskorištavajući IoT sustave vođene umjetnom inteligencijom za otkrivanje anomalija i sprječavanje kvarova opreme, smanjuje vrijeme zastoja i financijske gubitke. Dodatne IIoT primjene uključuju praćenje imovine, opskrbni lanac i upravljanje zalihama, daljinski nadzor i korištenje digitalnih blizanaca koji simuliraju procese u stvarnom svijetu za sigurnije eksperimentiranje i inovacije. Tehnologije pametnih gradova integriraju različite IoT module za aplikacije kao što su upravljanje prometom, nadzor vode i pametno parkiranje. Ovi sustavi poboljšavaju urbani život uvođenjem pametne rasvjete, nadzora i učinkovitih rješenja, pridonoseći sigurnijim i održivijim gradovima. Međutim, široko usvajanje suočava se s izazovima zbog značajnih ulaganja potrebnih za infrastrukturu i održavanje. Jedan od ključnih faktora u učinkovitosti IoT sustava danas je brzina prijenosa i obrade podataka. Jedan veliki napredak u rješavanju ovog izazova je integracija 5G tehnologije. Standard 5G veze značajno je poboljšao kvalitetu prijenosa podataka, nudeći nižu latenciju, veću propusnost i bržu komunikaciju između uređaja. Ova poboljšanja omogućuju IoT sustavima da podržavaju aplikacije u stvarnom vremenu, kao što su autonomna vozila, zdravstvena skrb na daljinu i pametni gradovi, uz veću pouzdanost i učinkovitost. Ovaj je razvoj ključan jer podržava funkcioniranje podatkovno intenzivnih IoT ekosustava i potiče inovacije u raznim industrijama. [6] [8]

2.2. Pregled tehnologija

U ovom poglavlju bit će napravljen pregled odabranih popularnih tehnologija koje omogućuju suvremene IoT sustave. Ukratko će biti opisano nekoliko odabranih protokola koji služe za komunikaciju između IoT uređaja, nekoliko odabranih IoT platformi te dva popularna mikrokontrolera na kojima se najčešće baziraju IoT uređaji.

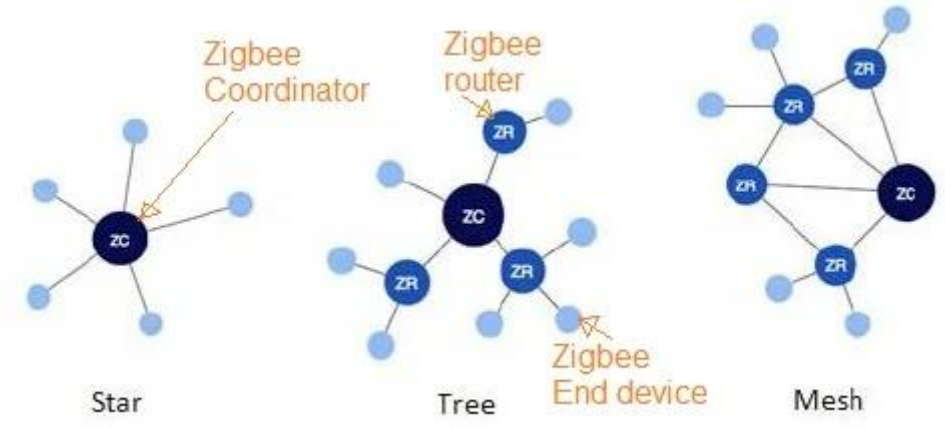
2.2.1.Komunikacijski protokoli

Komunikacijski protokoli imaju ključnu ulogu u omogućavanju razmjene podataka između uređaja, senzora i mreža. Ovi protokoli upravljaju načinom na koji uređaji komuniciraju, osiguravajući interoperabilnost, pouzdanost i učinkovito korištenje resursa. Zigbee i Bluetooth Low Energy podržavaju komunikaciju kratkog dometa, koja se često koristi u pametnim kućnim uređajima i nosivim tehnologijama. Za IoT aplikacije širokog područja, LoRaWAN i NB-IoT pružaju povezivost velikog dometa niske potrošnje. Dodatno, tradicionalni protokoli kao što su HTTP/HTTPS i Wi-Fi još uvijek se široko koriste za uređaje koji zahtijevaju veće brzine prijenosa podataka i izravan pristup internetu. Ovi protokoli osiguravaju da IoT sustavi ostanu skalabilni, sigurni i sposobni za prijenos kritičnih podataka, omogućujući aplikacije poput daljinskog nadzora, automatizacije i analitike u stvarnom vremenu u raznim industrijama. [9]

2.2.1.1. Zigbee

Zigbee je bežični komunikacijski protokol dizajniran za upotrebu u uvjetima gdje je potrebna niska potrošnja, niska brzina prijenosa podataka i rad na mali domet. Koristi se za kreiranje *ad hoc* isprepletenih mreža uređaja (eng. *Mesh network*). Definiran je Zigbee specifikacijom te mu je cilj da bude jednostavniji i jeftiniji od drugih bežičnih protokola za osobne bežične mreže (eng. *Personal area network*) kao što su Bluetooth ili Wi-Fi. Najčešće se koristi u automatizaciji doma za komunikaciju između bežičnih prekidača za svijetlo i nadzora potrošnje energije, za sustave upravljanja prometnom signalizacijom, prikupljanje podataka u zdravstvu, automatizaciji zgrada, detekciju dima i uljeza te druge za razne industrijske i potrošačke primjene gdje može ispuniti zahtjeve bežičnog prijenosa podataka niskog dometa i niske brzine. Domet prijenosa ograničen je na 10-100 metara kod uređaja koji su unutar vidokruga, no moguće je prenositi podatke na veće udaljenosti slanjem podataka kroz isprepletenu mrežu tako da se uređaji između primatelja i pošiljatelja ponašaju kao čvorovi koji prosljeđuju podatke dok ne stignu do krajnjeg čvora. Zigbee mreže osigurane su 128-bitnom simetričnom enkripcijom i zbog niske potrošnje koriste se kod uređaja koji zahtijevaju dugi radni vijek baterije. Definirana brzina slanja podataka je 250 kbit/s, a radna frekvencija je obično 2.4 GHz za potrošačke primjene. Prednosti Zigbee protokola su niska potrošnja energije, mogućnost kreiranja isprepletene mreže uređaja, interoperabilnost između uređaja različitih proizvođača, i niska cijena. Najveće mane ovog protokola su ograničeni domet, niska brzina prijenosa podataka, i problemi sa smetnjama zbog rada na istoj frekvenciji kao Wi-Fi i Bluetooth. [10] [11]

Postoje tri klase Zigbee uređaja: Zigbee koordinator, Zigbee ruter, Zigbee krajnji uređaj. Zigbee koordinator je korijen svake Zigbee mreže, postoji samo jedan te se može koristiti za spajanje na druge mreže. On pohranjuje podatke o mreži i djeluje kao repozitorij za sigurnosne ključeve. Zigbee ruter je uređaj koji vrši neku funkciju ovisno o instaliranoj aplikaciji, ali se ponaša i kao ruter pa tako omogućuje prosljeđivanje podataka prema drugim uređajima. Ovi uređaji se obično napajaju iz stalnog izvora napajanja da bi uvijek bili dostupni pa se još i nazivaju repetitorima ili proširivači dometa. Zigbee krajnji uređaji su uređaji koji imaju minimalnu funkcionalnost da bi mogli komunicirati s povezanim Zigbee čvorom (Zigbee koordinatorom ili Zigbee ruterom) što im omogućuje nisku potrošnju jer ne moraju uvijek biti aktivni. Ovi uređaji većinu vremena provode u mirovanju i napajaju se baterijama. [10]



Slika 3. Različite toplogije Zigbee mreža (Izvor: [11])

2.2.1.2. NFC

NFC (eng. *Near-field communication*) je tehnologija bežične komunikacije kratkog dometa koja uređajima omogućuje razmjenu informacija unutar nekoliko centimetara. NFC olakšava interakciju između uređaja, omogućujući siguran prijenos podataka, uparivanje uređaja i sustava za plaćanje. Radi na frekvenciji od 13.56 MHz i pri brzinama od 106 do 848 kbit/s. Značajan je za aplikacije poput pametnih domova, zdravstva i logistike, gdje je brza i sigurna komunikacija ključna. Na primjer, NFC oznake mogu se koristiti za praćenje imovine u stvarnom vremenu ili za provjeru autentičnosti korisnika u povezanim sustavima. NFC osim prijenosa podataka podržava i bežično punjenje pa ima primjenu kod kompaktnih prijenosnih IoT uređaja. Komunikacija se odvija između dva uređaja od kojih je jedan aktivan koji inicira komunikaciju i drugog uređaja koji može biti pasivan te je u tom slučaju napajan od strane aktivnog uređaja, ili isto može biti aktivan uređaj pa se komunikacija vrši tako da uređaji naizmjence prelaze iz aktivnog u pasivno stanje da bi razmjenjivali podatke. Postoje tri definirana načina rada u kojem može raditi svaki aktivan NFC uređaj. NFC emulacija kartice je način rada gdje se aktivan uređaj poput pametnog telefona može ponašati kao pametna kartica te koristiti za plaćanje i slično. NFC čitač/pisač je način rada koji omogućuje uređajima da čitaju ili pišu informacije po NFC oznakama. NFC *peer-to-peer* način rada omogućuje komunikaciju između NFC uređaja i razmjenu informacija na *ad hoc* način. Zbog ograničenog dometa standardni NFC nije siguran, ali se mogu implementirati sigurnosne značajke na razini aplikacije koja ga koristi. Prednosti su mu jednostavnost za korištenje, sigurnost komunikacije zbog niskog dometa s obzirom na druge protokole velikog dometa, i niska potrošnja energije. Pod mane možemo ubrojiti ograničeni domet, nisku brzinu prijenosa podataka, i nemogućnost proširenja na veće mreže zbog ograničenja dometa i brzine. [12]

2.2.1.3. Wi-Fi

Wi-Fi je skup protokola koji se koriste za kreiranje lokalnih mreža uređaja i pristup internetu. U kontekstu IoT-a ima ključnu ulogu omogućujući brzu povezivost za različite uređaje i sustave. Velika dostupnost i mogućnost prijenosa velike količine podataka velikim brzinama čine ga temeljem za pametne domove, industrijsku automatizaciju, i IoT sustave u zdravstvu. Najčešće radi na frekvencijama od 2.4 GHz i 5 GHz dok prema najnovijim standardima može postići brzine do 9.6 Gbit/s. Wi-Fi omogućuje uređajima poput pametnih termostata, sigurnosnih kamera i povezanih uređaja da prenose podatke na platforme u oblaku ili druge uređaje, olakšavajući nadzor i kontrolu u stvarnom vremenu. Prednosti su mu velika brzina prijenosa podataka, široka usvojenost budući da je to globalni standard, i velik domet koji omogućuje pokrivanje velikog područja. Kao nedostatke možemo navesti veću potrošnju energije, zagušenje u područjima s puno povezanih uređaja zbog smetnji, i sigurnosne rizike ako mreže nisu adekvatno zaštićene. [13]

2.2.1.4. 5G

Peta generacija mobilne mreže predstavlja veliki korak naprijed pružajući veće brzine, manja kašnjenja i veći kapacitet u usporedbi s prethodnom 4G generacijom. 5G omogućuje komunikaciju u stvarnom vremenu i podržava ogroman broj povezanih uređaja što omogućuje primjene kao što su autonomna vozila, pametni gradovi, industrijska automatizacija, proširena stvarnost i izvođenje medicinskih zahvata s udaljenog mjesta. 5G uređaji povezani su na internet i mobilnu mrežu preko baznih stanica, a brzine prijenosa mogu dostići do 20 Gbit/s u idealnim uvjetima. Ova tehnologija može biti implementirana u različitim frekvencijskim pojasima pa tako razlikujemo pojas niske frekvencije od 600-900 MHz koristeći milimetarske valove, pojas srednje frekvencije od 1.7-4.7 GHz koristeći mikrovalove i pojas visoke frekvencije od 24-47 GHz koristeći milimetarske valove. Trenutačno je najrašireniji implementacija u frekvencijskom rasponu od 24.25-29.5 GHz. Prednosti ove tehnologije su ogromne brzine prijenosa podataka, smanjena zagušenost mreže, niska latencija u komunikaciji te mogućnost spajanja ogromnih količina uređaja po kilometru kvadratnom što ju čini idealnom za rješenja poput pametnih gradova i industrijskog interneta stvari. No unatoč svim prednostima ova tehnologija suočava se s izazovima poput velikog broja potrebnih stanica za implementaciju što povećava troškove implementacije i smanjuje pokrivenost na ruralnim područjima, i potencijalni sigurnosni rizici povezani s upotrebom opreme Kineskih dobavljača koja potencijalno ima ugrađene mogućnosti za špijunažu. [14]

2.2.1.5. Bluetooth Low Energy

Bluetooth Low Energy (skraćeno *BLE*) je bežični komunikacijski protokol dizajniran za prijenos podataka kratkog dometa sličnog Bluetoothu, ali s bitno manjom potrošnjom energije. Specifikaciju je razvila tvrtka Nokia 2006. godine pod nazivom Wibree te je ona integrirana 2009. godine u Bluetooth 4.0 specifikaciju pod svojim sadašnjim imenom. Nije kompatibilna s klasičnim Bluetooth protokolom, ali radi na istoj frekvenciji od 2.4 GHz pa uređaji mogu koristiti jednu antenu za oba protokola. Brzine prijenosa podataka kreću se od 125 kbit/s pa sve do 2 Mbit/s ovisno o uvjetima okoline i generaciji protokola koja se koristi. Ciljane primjene su nosive tehnologije, pametni kućni uređaji i medicinski uređaji. Postoje predefinirani profili koji služe kao specifikacije kako se uređaj ponaša u nekoj primjeni kao što su na primjer profili za medicinske primjene, primjene u sportu, primjene za generičke senzore, primjena kod praćenja stanja baterija, audio primjene i mnogi drugi profili. Kao prednosti ovog protokola navodi se niska potrošnja energije, niska cijena i kompatibilnost s postojećim uređajima koji već podržavaju Bluetooth, no limitiran je kraćim dometom i mogućim smetnjama zbog preopterećenosti frekvencije koju koristi. [15]

2.2.1.6. LoRa

Skraćeno od *Long Range*, ovaj bežični komunikacijski protokol namijenjen je IoT namjenama koje zahtijevaju veliki domet i nisku potrošnju energije, posebno u uvjetima gdje uređaji trebaju prenositi male količine podataka na velike udaljenosti od 5-15 km. Radi na nelicenciranim frekvencijskim pojasevima ispod 1Ghz kao što su 433 MHz, 868 MHz i 915 MHz ovisno o zemlji u kojoj se koristi. Brzine prijenosa vrlo se niske i kreću se od 0.3 kbit/s do 27 kbit/s. Jedan je od najpopularnijih protokola jer pruža prijenos na velike udaljenosti poput Zigbee i Bluetooth protokola, ali uz bitno manju potrošnju energije i manje brzine prijenosa. Ovaj protokol definira samo fizički sloj za komunikaciju pa postoji i definiran LoRaWAN protokol za više slojeve i arhitekturu mreže. Prednosti LoRa i LoRaWAN uključuju produljeno trajanje baterije za IoT uređaje, pokrivenost velikog dometa i niske troškove implementacije zbog minimalnih infrastrukturnih zahtjeva. Oni također nude fleksibilnost, podržavajući i javne i privatne mrežne konfiguracije. Međutim, njihove niske brzine prijenosa podataka ograničavaju njihovu upotrebu u aplikacijama koje zahtijevaju brzi ili veliki prijenos podataka, a nelicencirani spektar sklon je smetnjama. Bez obzira na to, LoRaWAN je široko korišten u primjenama koje daju prednost dometu i učinkovitosti ispred brzine. [16]

2.2.1.7. NB-IoT

Narrowband Internet of things je protokol za širokopojasne mreže male snage (eng. *Low-power wide-area network*) razvijen za IoT primjene koje zahtijevaju pouzdanu komunikaciju niske propusnosti. Radi na licenciranim LTE frekvencijskim pojasevima koristeći

postojeću mobilnu infrastrukturu. Osigurava široku pokrivenost, čak i na udaljenim ili zatvorenim lokacijama, i dizajniran je za povezivanje uređaja koji povremeno prenose male količine podataka, kao što su pametna brojila, ekološki senzori i industrijski nadzorni sustavi. Omogućuje primjene kao što su infrastruktura pametnih gradova, poljoprivreda i praćenje imovine. Ima mogućnosti podržavanja velikog broja uređaja uz minimalnu potrošnju energije što ga čini atraktivnom opcijom za masovnu implementaciju IoT-a. Također nudi poboljšane sigurnosne značajke zbog oslanjanja na enkripciju mobilne mreže i protokole. Prednosti NB-IoT-a uključuju njegovu široku pokrivenost, nisku potrošnju energije i integraciju s postojećim mobilnim mrežama, čime se smanjuju troškovi implementacije. Međutim, njegovi nedostaci uključuju niže stope prijenosa podataka u usporedbi s drugim IoT tehnologijama i ovisnost o pružateljima mobilnih usluga za pokrivenost i infrastrukturu, što može ograničiti njegovu dostupnost u nekim regijama. [17]

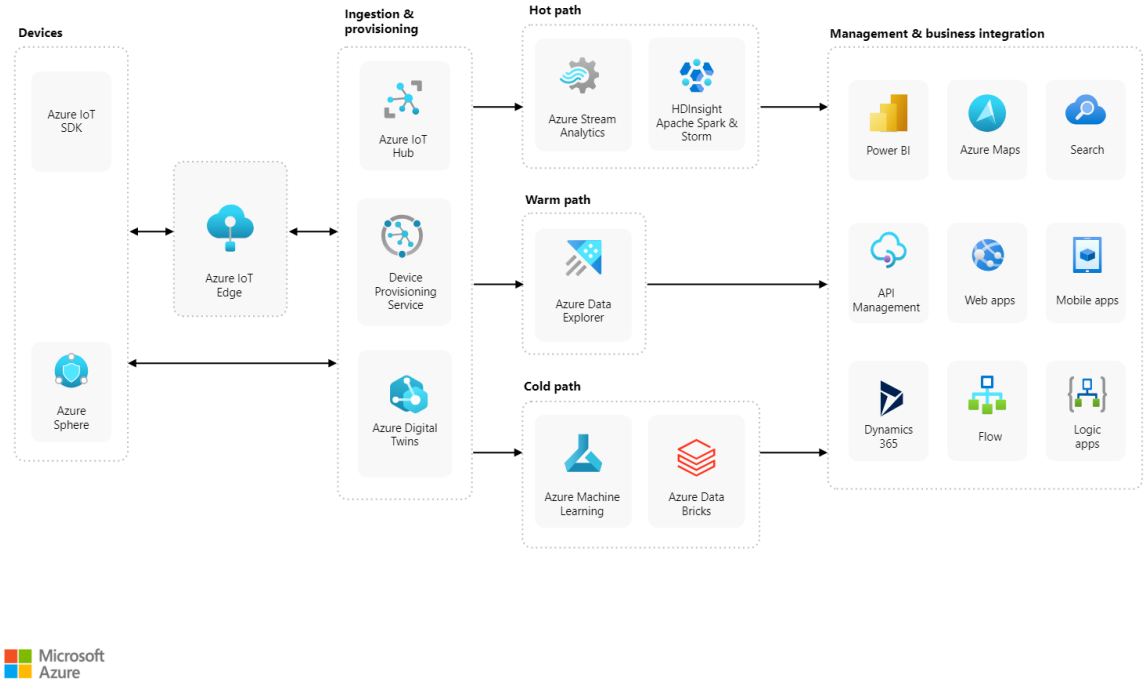
2.2.2.IoT platforme

Platforme služe kao okvir za implementaciju IoT ekosustava i pružaju funkcionalnosti poput upravljanja životnim ciklusom uređaja, alate za komunikaciju, analitiku, integraciju uređaja, i upravljanje aplikacijama. Obično IoT platforme funkcioniraju kao posrednici između hardvera i aplikacijskih slojeva. Olakšavaju siguran prijenos podataka, obrađuju podatke koristeći rubno računalstvo ili računalstvo u oblaku te pružaju uvide putem nadzornih ploča i API-ja za donošenje odluka. Prednosti IoT platformi uključuju pojednostavljenje procesa razvoja, poboljšanje skalabilnosti i ponudu integriranih rješenja s ugrađenim sigurnosnim značajkama. Njihovi nedostaci uključuju moguće probleme s kompatibilnošću između uređaja i platformi, oslanjanje na internetsku povezanost i troškove povezane s rješenjima na razini platforme. Unatoč ovim izazovima, IoT platforme ključne su za moderne IoT projekte, pojednostavljuju implementaciju i omogućuju napredne funkcionalnosti. [18]

2.2.2.1. Microsoft Azure IoT

Microsoft Azure IoT je paket usluga u oblaku dizajniran za pojednostavljenje stvaranja, implementacije i upravljanja IoT rješenjima. Kao dio Azure ekosustava, pruža alate za povezivanje, nadzor i kontrolu IoT uređaja putem centralizirane platforme. Azure IoT omogućuje integraciju s hardverom s pomoću protokola kao što su MQTT, HTTP, and AMQP i koristi napredne mogućnosti kao što su rubno računalstvo, umjetna inteligencija i strojno učenje za obradu podataka i analitiku. Podržava širok raspon primjena, uključujući prediktivno održavanje, daljinski nadzor i primjene za pametne gradove. Ključne prednosti Azure IoT uključuju njegovu skalabilnost, robusne sigurnosne značajke i integraciju s drugim Microsoftovim uslugama kao što su Power BI i Azure Machine Learning, koje omogućuju naprednu analitiku. Modularna struktura Azure IoT-a omogućuje programerima odabir usluga

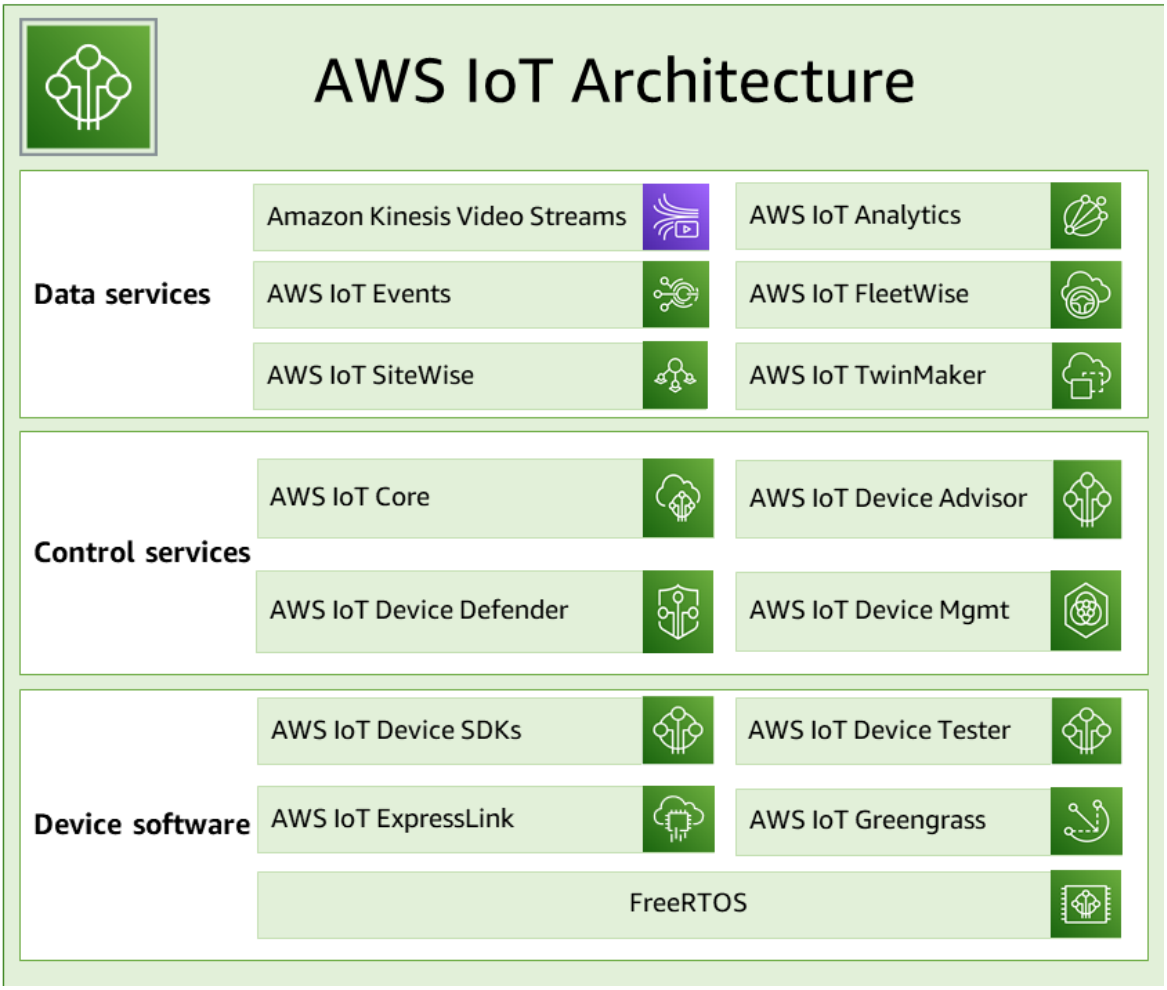
prilagođenih njihovim specifičnim potrebama. Neki od izazova kod primjene su složenost upravljanja i optimizacija troškova implementacije, kao i krivulja učenja za korisnike koji su novi u korištenju IoT rješenja u oblaku. [19]



Slika 4. Referentna arhitektura Azure IoT-a (Izvor: [19])

2.2.2.2. AWS IoT Core

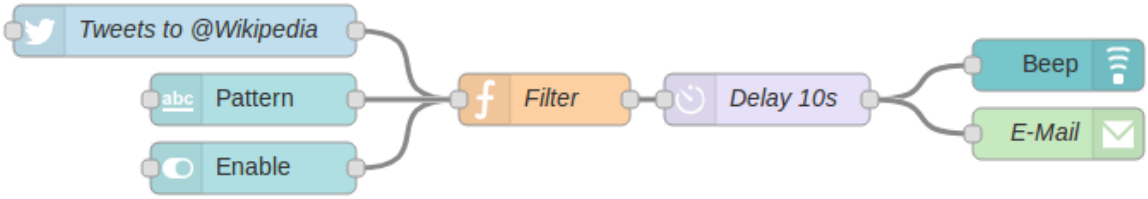
AWS IoT Core je Amazon-ova usluga u oblaku dizajnirana da omogući sigurnu i učinkovitu interakciju između IoT uređaja i aplikacija u oblaku. Pruža robustan okvir za povezivanje milijuna uređaja, omogućujući im da komuniciraju međusobno i s oblakom u stvarnom vremenu. AWS IoT Core podržava komunikacijske protokole kao što su MQTT, WebSockets i HTTP, što ga čini pogodnim za razne primjene. Usluga se također može integrirati s drugim AWS uslugama, kao što je Lambda za obradu vođenu događajima, Amazon S3 za pohranu i SageMaker za strojno učenje, omogućujući naprednu analitiku podataka i automatizaciju. AWS IoT Core nudi prednosti kao što su skalabilnost, jake sigurnosne značajke (npr. upravljanje identitetom i pristupom, i autentifikacija uređaja) i opsežna integracija s AWS ekosustavom. Nedostaci uključuju potencijalno visoke troškove za implementaciju i krivulju učenja za korisnike koji nisu upoznati s opsežnom ponudom usluga AWS-a. Unatoč tome, AWS IoT Core moćna je platforma za razvoj, upravljanje i skaliranje IoT rješenja. [20]



Slika 5. Pregled AWS IoT Core usluga (Izvor: [21])

2.2.2.3. Node-RED

Node-RED je razvojni alat otvorenog koda koji se temelji na tokovima i dizajniran je za grafičko programiranje i povezivanje uređaja, API-ja i mrežnih usluga. Razvio ga je IBM, a temelji se na Node.js-u koji se pokreće u pregledniku, što ga čini jednostavnim i vrlo proširivim. Node-RED pruža grafičko sučelje gdje korisnici mogu stvoriti tokove kreiranjem čvorova i veza između objekata koji predstavljaju uređaje, usluge ili funkcije za obradu podataka. Objekti se povezuju kako bi stvorili radne tokove automatizacije bez potrebe za opsežnim znanjem programiranja. Node-RED pojednostavljuje IoT projekte podržavajući brojne protokole kao što su MQTT, HTTP i WebSockets te se integrira se s raznim uslugama u oblaku i hardverskim sustavima, što ga čini svestranim za primjenu u automatizaciji i obradi podataka. Node-RED je popularan izbor za IoT aplikacije zbog svoje fleksibilnosti. Prednosti Node-RED-a uključuju jednostavnost korištenja, mogućnosti brze izrade prototipova i veliku, aktivnu zajednicu koja nudi čvorove i dodatke za višekratnu upotrebu. Međutim, njegovo oslanjanje na JavaScript može predstavljati ograničenje za programere koji nisu upoznati s jezikom, a performanse mogu biti ograničene u scenarijima koji zahtijevaju izuzetno velike resurse. [22] [23]



Slika 6. Primjer Node-RED toka (Izvor: [22])

2.2.2.4. Home Assistant

Home Assistant je platforma za kućnu automatizaciju otvorenog koda dizajnirana da pomogne korisnicima da integriraju, kontroliraju i automatiziraju svoje pametne uređaje. Razvoj je započeo Paulus Schoutsen 2013. godine kao Python aplikaciju. Nudi opsežnu kompatibilnost sa širokim rasponom IoT uređaja i protokola, omogućujući korisnicima da upravljaju svime, od svjetala i termostata do kamera i glasovnih pomoćnika s jedne upravljačke ploče. Razvijen je imajući na umu privatnost i može raditi lokalno, bez oslanjanja na usluge u oblaku, osiguravajući da korisnički podaci ostanu privatni. Home Assistant radi koristeći konfiguracijski sustav temeljen na YAML-u i modularnu arhitekturu. Podržava integraciju s uređajima koji koriste protokole kao što su Zigbee, Z-Wave, MQTT i mnoge druge. Automatizacije se mogu postaviti s pomoću skripti ili s pomoću ugrađenog grafičkog alata za programiranje, omogućujući korisnicima da definiraju uvjete i akcije za različite pametne uređaje. Može se implementirati na uređajima kao što je Raspberry Pi ili se implementirati u virtualiziranim okruženjima na lokalnom poslužitelju. Ključne prednosti Home Assistanta uključuju njegovu fleksibilnost, veliku kompatibilnost uređaja i jaku podršku zajednice. No oslanjanje na ručnu konfiguraciju može predstavljati problem i povećati krivulju učenja za početnike, a upravljanje ažuriranjima za integrirane uređaje može zahtijevati puno vremena. Unatoč ovim problemima Home Assistant je jako raširen među tehnički potkovanim korisnicima koji traže prilagodljivo, privatno i snažno rješenje za kućnu automatizaciju. [24] [25]

2.2.3.Mikrokontroleri

Mikrokontroleri su kompaktni, integrirani krugovi koji sadrže jedan ili više procesora s uključenom memorijom i programabilnim ulazima i izlazima. Zbog svoje male veličine i niske cijene često se integriraju u IoT uređaje gdje nije potrebna velika procesorska snaga za prikupljanje podataka sa senzora i kontroliranje mehaničkih komponenti poput releja. Aplikacije za mikrokontrolere najčešće se pišu u programskim jezicima koji su blizu razine hardvera poput Assemblya i C programskog jezika. Aplikacije ili programirane instrukcije pohranjuju se na ugrađenu EEPROM ili flash memoriju koja je ograničena svojom veličinom i obično rade u okruženjima u stvarnom vremenu gdje su dizajnirani za specifične primjene. Popularne obitelji

mikrokontrolera koje se koriste u IoT-u su ESP32 i STM32 koji pružaju široki spektar mikrokontrolera s različitim razinama procesorske snage i različitim podržanim komunikacijskim protokolima. Prednosti mikrokontrolera uključuju njihovu nisku cijenu, energetsku učinkovitost i prikladnost za male, ugrađene sustave. Oni su svestrani i mogu se prilagoditi specifičnim primjenama s minimalnim hardverom. Međutim, njihova ograničenja uključuju relativno nisku procesorsku snagu u usporedbi s robusnijim procesorima i ograničenu memoriju, što može ograničiti njihovu upotrebu u podatkovno intenzivnim ili vrlo složenim zadacima. [26]

2.2.3.1. STM32

STM32 je serija 32-bitnih RISC mikrokontrolera koju razvija tvrtka STMicroelectronics baziranih na ARM Cortex jezgrama. Ovisno o Cortex jezgrama koje se koriste mikrokontroleri su podijeljeni na različite kategorije: mikrokontroleri visokih performansi, mikrokontroleri za opću upotrebu, mikrokontroleri jako niske snage, i mikrokontroleri s podrškom za bežičnu povezivost. Unutar svake kategorije mikrokontroleri se dijele na serije ovisno o primjeni, performansama, i cijeni. Svaka serija mikrokontrolera bazirana je na jednoj specifičnoj ARM Cortex jezgri. [27]

Unutar kategorije mikrokontrolera visokih performansi proizvode se serije STM32F2, STM32F4, STM32F7, STM32H5 i STM32H7. Ova serija namijenjena je za primjene koje zahtijevaju veliku procesorsku snagu, široku podršku za periferiju i povezivost s drugim IoT uređajima, i sigurnosne značajke za sigurne IoT sustave. Karakterizira ih visoki radni takt jezgri, veća količina radne memorije, široka podrška za raznu periferiju i energetska efikasnost. Mikrokontroleri u ovoj kategoriji imaju nešto višu cijenu u usporedbi s ostalim i povećanu kompleksnost kod razvoja s obzirom na njihovu više jezgrenu arhitekturu. [28]

Kategorija mikrokontrolera za opću namjenu uključuje serije STM32C0, STM32F0, STM32F1, STM32F3, STM32G0, STM32G4. Namijenjeni su za veliki raspon primjena zbog svoje široke ponude jezgri i periferije koju podržavaju. Jednostavnija arhitektura i podrška razvojnih alata čine ih dobrim izborom za uvjete gdje vrijeme razvoja i puštanja uređaja u prodaju treba biti što kraće. Ugrađuju se u sustave koji trebaju biti priuštivi, robusni i s dugim vijekom trajanja poput upravljača za industrijske motore ili napajanja i pretvarača. [29]

Mikrokontroleri jako niske snage bazirani su na STM-ovoj tehnologiji ultra-niskog curenja i optimiziranom dizajnu koji nudi energetsku učinkovitost i ravnotežu između performansi, snage, sigurnosti, i cijene. Idealni su za scenarije u kojima je niska potrošnja energije kritična. Široka ponuda u ovoj kategoriji mikrokontrolera osigurava skalabilnost u različitim primjenama, od jednostavnih IoT senzora do sigurnih i visokoučinkovitih nosivih

uređaja. Kategorija uključuje serije STM32L0, STM32L1, STM32L4, STM32L4+, STM32L5, STM32U0 i STM32U5. [30]

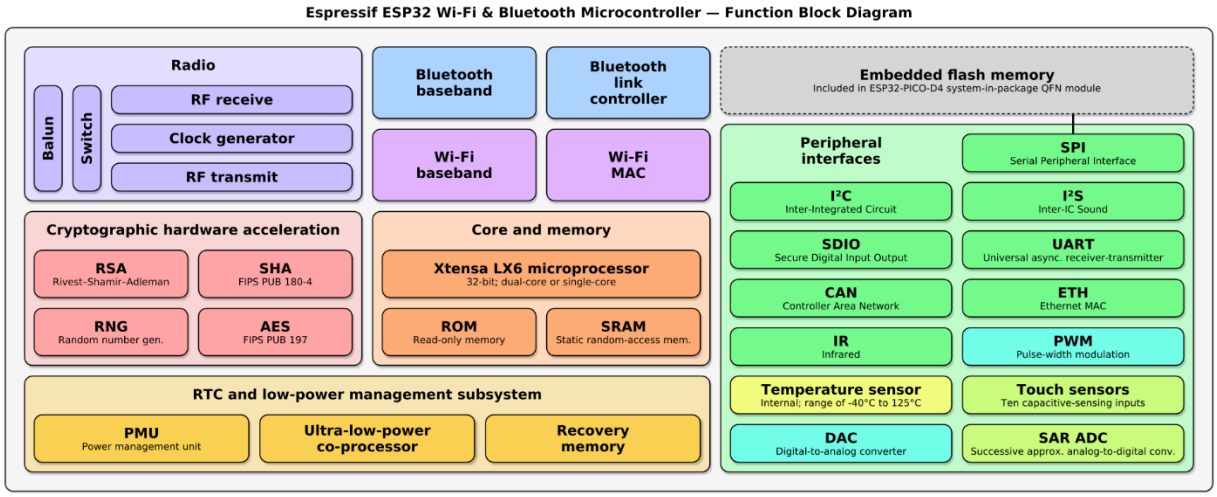
Kategorija s podrškom za bežičnu povezivost je najmanja po broju mikrokontrolera u ponudi i sastoji se od serija STM32WB, STM32WB0, STM32WBA za komunikaciju kratkog dometa i STM32WL seriju za komunikaciju velikog dometa. Radijski moduli ugrađuju se paralelno u mikrokontroler i smanjuju potrebu za vanjskim modulima. Ovakva dvojezgrena arhitektura i integracija smanjuje složenost hardvera, ukupnu veličinu uređaja i potrošnju energije. Kompatibilni su s širokim rasponom bežičnih protokola poput ZigBee, Bluetooth Low Energy, LoRa i drugim protokolima u rasponu ispod 1 GHz. Primjenjuju se za nosive uređaje u zdravstvu, potrošačkoj elektronici i za industrijska IoT rješenja poput udaljenog mjerenja, praćenja imovine, i daljinskog nadzora. [31]

Uz široku ponudu mikrokontrolera STMicroelectronics nudi i veliki broj alata za razvoj, biblioteka i kompleta za razvoj kako bi programerima omogućio jednostavniju izradu aplikacija koristeći STM32 mikrokontrolere. Razvojni paketi podijeljeni su na tri kategorije, Nucleo, Discovery i Evaluation, koje nude platforme za testiranje i izradu prototipa temeljenih na STM mikrokontrolerima. U kombinaciji s alatima poput STM32CubeMX, STM32Cube Programmer, STM-MCU-Finder, bibliotekama koje nude široku podršku za razne upravljačke programe i opsežnom dokumentacijom bitno pojednostavljuju i ubrzavaju razvojni proces. Posljedica ovakve široke ponude mikrokontrolera i veliki ekosustav programske podrške je visoka krivulja učenja i kompleksnost za početnike. [32]

2.2.3.2. ESP32

ESP32 je mikrokontroler nasljednik ESP8266 mikrokontrolera proizvođača Espressif Systems. Dolazi u varijanti s jednom ili dvije Tensilica Xtensa LX6 jezgrama, varijanti s dvije Xtensa LX7 jezgre, ili s jednom ili više RISC-V jezgrom. Za razliku od prethodnika koji je uključivao samo WiFi modul uključuje i Bluetooth modul. Dolazi u nekoliko serija i to su ESP32, ESP32-S, ESP32-C i varijantama ESP32-H2 i ESP32-P4. ESP32 je originalna varijanta opće namjene koja dolazi s Xtensa LX6 jezgrama i 34 programabilna ulaza/izlaza. Podržava razne načine rada sa smanjenom potrošnjom energije i hardverske sigurnosne značajke. ESP32-S serija sastoji se od ESP32-S2 i ESP32-S3 varijante koje sadrže brže Xtensa LX7 jezgre. Najveća razlika između ove dvije varijante je što je S2 jednojezgreni mikroprocesor bez podrške za Bluetooth dok je S3 dvojezgreni i uključuje Bluetooth. Osim toga oba imaju 40-ak programabilnih ulaza/izlaza i podršku za raznu periferiju. ESP32-C serija bazirana je na jednoj RISC-V jezgri i sastoji se od varijanti ESP32-C2, ESP32-C3 i ESP32-C6. Varijanta ESP32-H2 također je jednojezgreni RISC-V mikrokontroler dok je ESP32-P4 dvojezgreni i podržava napredne AI funkcionalnosti. Sve serije ESP32 mikrokontrolera su 32-bitne i uključuju

višekanalne pretvarače analognog u digitalni signal i obrnuto, i raznu periferiju poput CAN, SPI, I2C, UART, PWM i druge. Proizvode se i razni razvojni kompleti bazirani na ESP32 mikrokontrolerima od strane raznih proizvođača koji olakšavaju prototipiranje i razvoj aplikacija od kojih su poznatiji Adafruit, Arduino i SparkFun. Uz dostupnu dokumentaciju i razne razvojne alate koje nudi Espressif poput ESP HomeKit SDK, ESP-ADF, ESP-AT, i ESP-IDF službenog okvira za razvoj IoT aplikacija baziranih na ESP32 mikrokontroleru omogućuje jednostavan razvoj i upoznavanje s ekosustavom ovog proizvođača. ESP32 mikrokontroleri imaju široku primjenu za industrijska i komercijalna IoT rješenja i nude napredne mogućnosti obrade podataka, pouzdanu povezivost, napredne sigurnosne značajke i energetsku učinkovitost za rješavanje različitih problema. No, kao i kod STM32 mikrokontrolera ovakav veliki ekosustav predstavlja visoku krivulju učenja i kompleksnost za početnike. [33] [34]



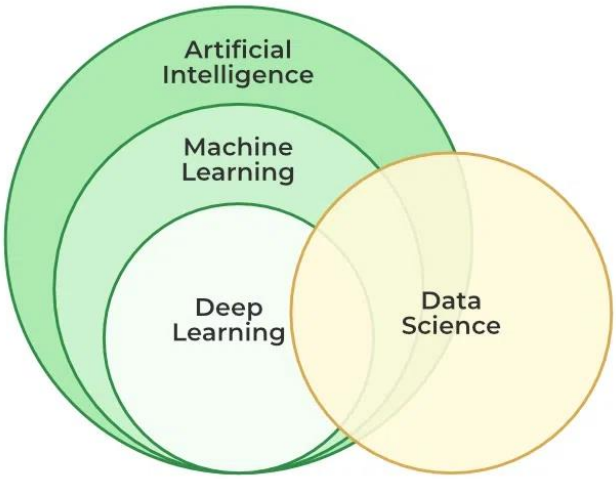
Slika 7. ESP32 funkcijski blok dijagram (Izvor: [33])

3. Strojno učenje

Strojno učenje (*eng. Machine Learning*) podskup je umjetne inteligencije koji se fokusira na razvoj algoritama za otkrivanje obrazaca unutar velikih skupova podataka i omogućuje aplikacijama da poboljšaju svoje performanse kroz vrijeme. Tradicionalno strojno učenje kombinira podatke sa statističkim alatima za predviđanje i omogućuje računalima da uče iz podataka i donose odluke ili predviđanja a da su za to nisu eksplicitno programirana. Koristeći povijesne podatke kao ulazne podatke, ovi algoritmi mogu napraviti predviđanja, klasificirati informacije, grupirati podatkovne točke, smanjiti dimenzionalnost i generirati novi sadržaj. U tradicionalnom programiranju računalo slijedi skup unaprijed definiranih uputa za izvođenje zadatka dok se kod strojnog učenja računalu daje skup primjera i zadatak koji treba izvršiti, ali na računalu je da otkrije obrasce u podacima kako izvršiti zadatak na temelju primjera koje mu je dalo. Na primjer, ako želimo da računalo prepoznaje slike osoba, ne dajemo mu posebne upute o tome kako osoba izgleda, umjesto toga dajemo mu veliku količinu slika koje prikazuju osobu i prepuštamo algoritmu strojnog učenja da otkrije obrasce i značajke koje definiraju osobu. Kako algoritam obrađuje sve više slika, kroz vrijeme postaje sve bolje u prepoznavanju osobe, čak i kada mu se prezentiraju slike koje prije nije obradilo. Ova sposobnost učenja iz podataka i poboljšanja tijekom vremena čini strojno učenje nevjerojatno moćnim. Strojno učenje široko je primjenjivo u mnogim industrijama i nalazi primjenu u različitim područjima kao što su prepoznavanje slike i govora, obrada prirodnog jezika, sustavi preporuka, otkrivanje prijevara, optimizacija portfelja, glasovni asistenti, autonomna vozila i automatizacija zadataka. Treniranje modela strojnog učenja često zahtijeva velike količine visokokvalitetnih podataka za dobivanje točnih rezultata. Sami rezultati, osobito oni iz složenih algoritama kao što su duboke neuronske mreže, mogu biti teški za razumijevanje, modeli mogu biti skupi i teški za podešavanje. Ipak, većina organizacija koristi strojno učenje, bilo izravno ili putem proizvoda s ugrađenim strojnim učenjem. Danas je strojno učenje neophodan alat kako bi se obradila i iskoristila sve veća količina podataka koju generiraju moderni sustavi. [35] [36]

Algoritmi strojnog učenja obično rade u nekoliko koraka. Prvi korak je prikupljanje relevantnih podataka, ovi podaci uključuju primjere i atribute koji su važni za problem koji se rješava i mogu biti raznim oblicima kao što su slike, tekst, numerički podaci i drugi oblici. Drugi korak je predobrada podataka koja služi kako bi se podaci strukturirali u oblik pogodan za korištenje s odabranim algoritmom. To može uključivati čišćenje podataka, popunjavanje vrijednosti koje nedostaju, modifikacija vrijednosti koje odstupaju, transformaciju podataka poput normalizacije ili skaliranja, i dijeljenje podataka u skupove. Zatim se ovisno o problemu koji se rješava odabire odgovarajući model strojnog učenja. Primjeri modela su stabla odlučivanja, neuronske mreže ili napredniji modeli poput dubokog učenja. Nakon što se

odabere model sljedeći korak je treniranje modela ulaznim podacima. Prilikom treniranja algoritmi iterativno pronalaze obrasce i veze između podataka i prema tome prilagođavaju model kako bi se što više smanjila razlika između predviđenih i stvarnih izlaza ili rezultata. Kad je treniranje modela završeno pristupa se evaluaciji modela. Procjenjuje se koliko dobro model daje izlaze na temelju nekih testnih podataka za procjenu. Mjere se metrike poput točnosti, preciznosti i srednja kvadratna pogreška. Na temelju toga može se zaključiti koliko dobro model obrađuje i klasificira nove, nepoznate podatke. Nakon toga na modelu se mogu podešavati hiperparametri, parametri koji se ne uče izravno tijekom treniranja, poput brzine učenja ili broja skrivenih slojeva u neuronskoj mreži kako bi se poboljšala učinkovitost modela. Na kraju se istrenirani model može koristiti za donošenje odluka ili predviđanja na temelju novih podataka. Ovaj proces podrazumijeva primjenu naučenih i otkrivenih obrazaca na nove ulazne podatke kako bi se generirali novi izlazi. [35]



Slika 8. Odnosi između polja podatkovnih znanosti (Izvor: [37])

Strojno učenje može se podijeliti na nadzirano strojno učenje, nenadzirano strojno učenje i podržano strojno učenje. Kod nadziranog strojnog učenja algoritam se trenira na skupu podataka koji su označeni i definirano je na temelju kojih varijabli treba pronaći obrasce i veze između podataka. Algoritam ima ulazne značajke i odgovarajuće izlazne oznake, te uči generalizirati iz tih podataka kako bi napravio predviđanja o novim podacima. Postoje dvije glavne vrste nadziranog strojnog učenja i to su regresija i klasifikacija. Kod regresije algoritam uči predviđati kontinuirane vrijednosti na temelju ulaznih značajki. Izlazne oznake su najčešće kontinuirane vrijednosti poput cijena dionica ili nekretnina. Algoritmi regresije mogu biti linearna regresija, polinomna regresija, grebenska regresija, regresijska stabla odlučivanja, regresija slučajne šume i drugi. Klasifikacija je vrsta nadziranog učenja kod koje algoritam uči podijeliti ulazne podatke na određene kategorije ili klase na temelju ulaznih značajki. Izlazne oznake u klasifikaciji su diskretne vrijednosti, mogu biti binarne ili višeklasne. Neki od algoritama

klasifikacije su logistička regresija, naivni Bayes klasifikator, stablo odlučivanja, metoda K-najbližih susjeda i stroj potpornih vektora. [35]

S druge strane nenadzirano strojno učenje je treniranje modela na neoznačenom skupu podataka te je na modelu da sam pronađe obrasce i odnose između podataka. Cilj mu je otkriti strukturu ili distribuciju podataka bez nadzora. Često se koristi za grupiranje i smanjenje dimenzionalnosti podataka. Kod grupiranja se podaci sa sličnim podatkovnim točkama grupiraju zajedno, a kod smanjenja dimenzionalnosti se smanjuje broj varijabli koje se razmatraju da bi se dobio skup glavnih varijabli. Većina algoritama za duboko učenje poput neuronskih mreža koriste nenadzirane algoritme. Dva glavna tipa nenadziranog strojnog učenja su klasteriranje i smanjenje dimenzionalnosti. Algoritmi klasteriranja grupiraju slične podatke zajedno na temelju njihovih karakteristika. Na taj se način identificiraju skupine podataka koji su slični, a istovremeno se razlikuju od drugih skupina. Popularni algoritmi uključuju K-srednje vrijednosti, hijerarhijsko klasteriranje i DBSCAN. Algoritmi za smanjenje dimenzionalnosti smanjuju broj ulaznih varijabli u skupu podataka dok istovremeno pokušavaju održati što je više moguće informacija. Ovime se smanjuje složenost skupa podataka i olakšava se analiza. Neki popularni algoritmi koji se koriste su PCA, t-SNE i autoenkoderi. [35]

Podržano učenje je vrsta strojnog učenja gdje se algoritam dizajnira tako da postoje unaprijed definirana pravila i postavljeni cilj. Model se nagrađuje za radnje koje ga dovode bliže cilju, a kažnjava se za radnje koje ga udaljavaju od cilja. Za razliku od nadziranog i nenadziranog učenja, podržano učenje posebno je prikladno za probleme u kojima su podaci sekvencijalni, a odluka donesena u svakom koraku može utjecati na buduće ishode. Često se koristi u robotici kako bi roboti naučili igrati igre ili izvršavati zadatke u fizičkom svijetu, i kod alokacije resursa u organizacijama. Dva glavna tipa podržanog učenja su podržano učenje bazirano na modelu i podržano učenje bez modela. U podržanom učenju baziranom na modelu, agent uči model okoline, uključujući prijelazne vjerojatnosti između stanja i nagrade povezane sa svakim parom stanje-radnja. Agent zatim koristi ovaj model za planiranje svojih radnji kako bi maksimizirao očekivanu nagradu. Kod podržanog učenja bez modela agent uči politiku izravno iz iskustva bez eksplicitne izgradnje modela okoline. Agent je u interakciji s okolinom i ažurira svoju politiku na temelju nagrada koje prima. [35] [38]

3.1. Povijest

Prvi matematički model povezan sa strojnim učenjem prezentirali su Walter Pitts i Warren McCulloch 1943. godine. U znanstvenom radu “A logical calculus of the ideas immanent in nervous activity” prezentirali su model neuronske mreže. Pojam strojno učenje skovao je Arthur Samuel 1959. godine, IBM-ov zaposlenik i jedan od pionira na područjima

računalnih igara i umjetne inteligencije. Jedan od najranijih modela strojnog učenja je njegov program koji je izračunavao šanse za pobjedu u dami još 1950. godine. Alan Turning kreirao je Turingov test iste godine kojemu je cilj odrediti posjeduje li računalo inteligenciju. Tijekom testa čovjek u ulozi suca komunicira prirodnim jezikom s jednim čovjekom i jednim računalom te ako on ne može sa sigurnošću utvrditi koji sugovornik je računalo smatra se da je prošlo test. No povijest strojnog učenja provlači se kroz desetljeća istraživanja s ciljem da se prouče ljudski kognitivni procesi. Kanadski psiholog Donald Hebb je 1949. godine objavio knjigu Organizacija ponašanja (eng. *The Organization of Behavior*) u kojoj je predstavio teoretsku neuronsku strukturu koju formiraju određene interakcije između živčanih stanica. Takav model neurona u međusobnoj interakciji predstavlja temelje za rad algoritama umjetne inteligencije i strojnog učenja. 1960-ih godina tvrtka Raytheon razvila je eksperimentalni stroj za učenje s memorijom na bušenoj vrpci imena Cybertron. Korišten je za analizu sonarnih signala, elektrokardiograma i govora korištenjem podržanog učenja. Operator ga je neprestano morao trenirati da prepoznaje obrasce s pomoću gumba kojim mu je signalizirao netočne odluke. Interes za prepoznavanje uzoraka nastavio se pa je 1981. godine objavljen izvještaj o korištenju strategija podržanog učenja kako bi umjetna neuronska mreža naučila prepoznavati 40 različitih znakova uključujući slova, znamenke i posebne simbole. 1990-ih godina fokus znanstvenih radova na područjima strojnog učenja prešao je s pristupa vođenog znanjem na pristupe vođene podacima. Znanstvenici su počeli stvarati programe za analizu velikih količina podatak i izvlačenje znanja i učenje iz rezultata. IBM je izradio Deep Blue, sustav za igranje šaha koji je 1997. godine pobijedio svjetskog prvaka u šahu. 2006. godine pojavio se pojam duboko učenje koji je skovao Geoffrey Hinton kako bi objasnio nove algoritme koji omogućuju računalima da vide i razlikuju objekte na slikama i videozapisima. Do 2015. godine razvijeni su različiti sustavi za specifične namjene poput Microsoftovog Kinect sustava za praćenje ljudskih karakteristika brzinom od 30 puta u sekundi omogućujući ljudima interakciju s računalom pokretima tijelom, IBM je razvio Watson koji je uspio pobijediti ljudsku konkurenciju u kvizu Jeopardy, Google predstavlja Google Brain koji je s pomoću duboke neuronske mreže mogao naučiti otkrivati i kategorizirati objekte na način na koji to radi mačka, dok je Facebook razvio DeepFace, softverski algoritam koji može prepoznati pojedince na fotografijama na istoj razini kao što to mogu i ljudi. 2015. godine Amazon je lansirao svoju platformu za strojno učenje, dok je Microsoft predstavio razvojni okvir Distributed Machine Learning Toolkit za učinkovitu distribuciju zadataka strojnog učenja na više računala. Iste godine više od 3000 istraživača umjetne inteligencije i robotike, potpomognuti poznatim osobama poput Stephena Hawkinga, Elona Muska i Stevea Wozniaka, potpisalo je pismo u kojem upozoravaju protiv autonomnog oružja sposobnog odabrati i gađati mete bez ljudskog nadzora. U 2016. godini algoritam Google DeepMind-a AlphaGo ušao je u povijest pobijedivši profesionalnog igrača u složenoj društvenoj igri Go, pobijedivši u svih pet igara. Do 2017. Waymo je počeo testirati autonomne

automobile u SAD-u s rezervnim vozačima, a kasnije je u Phoenixu predstavio potpuno autonomne taksije. Usred COVID pandemije 2020. godine OpenAI je predstavio GPT-3, revolucionarni model obrade prirodnog jezika sa 175 milijardi parametara, obučen s pomoću superračunala Microsoft Azure. Zbog svoje sposobnosti da proizvede tekst nalik ljudskom iz upita postaje jedan od najnaprednijih jezičnih modela do danas. [39] [40]

3.2. Algoritmi strojnog učenja

Algoritmi strojnog učenja temelj su sustava koji se koriste za obradu velikih količina podataka i izvlačenje znanja iz njih. Omogućujući računalima da identificiraju obrasce, donose predviđanja ili automatiziraju odluke na temelju podataka, računala mogu obraditi ogromne količine strukturiranih i nestrukturiranih podataka kako bi izvukli znanje i riješili složene probleme u raznim domenama. Za odabir algoritma ključno je znanje iz domene problema koji se rješava kako bi se izradili učinkoviti modeli i dobile smislene informacije i uvidi iz podataka. Strojno učenje je polje u kojem se događa stalni tehnološki napredak i razumijevanje algoritama ključno je za svakog inženjera ili podatkovnog znanstvenika. Algoritmi strojnog učenja raznoliki su i prilagođeni specifičnim vrstama problema, odabir pravog algoritma ovisi o prirodi problema, dostupnim podacima i željenom ishodu. Svaki algoritam donosi jedinstvene prednosti i kompromise. U ovom poglavlju bit će objašnjeno nekoliko odabranih algoritama koji se najčešće koriste u strojnom učenju. [41]

3.2.1. Linearna regresija

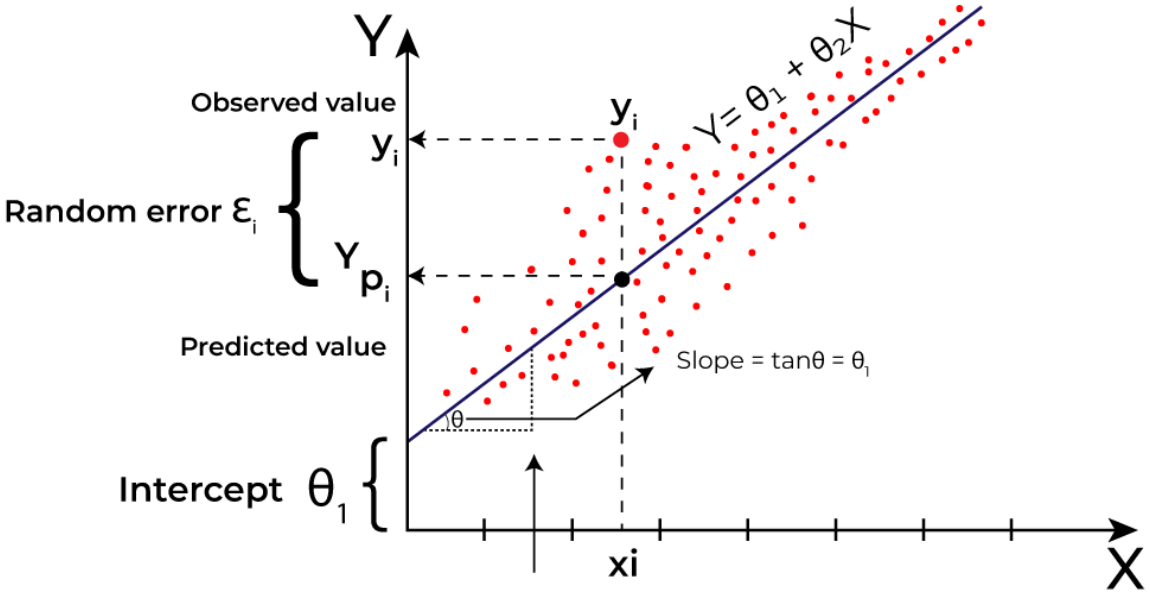
Linearna regresija algoritam je koji se koristi kod nadziranog strojnog učenja, vrsta je algoritma koji izračunava linearni odnos između dvije zavisne varijable i jedne ili više nezavisnih značajki prilagođavanjem linearne jednadžbe. Kada postoji samo jedna zavisna varijabla smatra se univarijatnom linearnom regresijom, ako ih postoji više od jedne onda je to multivarijatna regresija. Isto tako ako postoji samo jedna neovisna značajka naziva se jednostavna linearna regresija, a ako ih postoji više naziva se višestruka linearna regresija. Jednadžba modela jasno odražava utjecaj svake nezavisne varijable na zavisnu varijablu s pomoću koeficijenata. Jednostavnost i jednostavna implementacija čini osnovu pri implementaciji složenijih algoritama. Najjednostavniji oblik linearne regresije opisan je jednadžbom koja uključuje samo jednu nezavisnu i samo jednu zavisnu varijablu

$$Y = \beta_0 + \beta_1 X$$

gdje je Y zavisna varijabla, X nezavisna varijabla, β_0 presjek, a β_1 nagib pravca. Kod multivarijatne linearne regresije jednadžba glasi

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n$$

Cilj ovog algoritma je pronaći pravac (eng. *fit line*) koji najbolje može predvidjeti vrijednosti na temelju nezavisnih varijabli sa što manjom razlikom između predviđenih i stvarnih vrijednosti. Nagib pravca pokazuje koliko se brzo zavisna varijabla mijenja u ovisnosti o nezavisnoj varijabli. [42]



Slika 9. Linearna regresija (Izvor: [42])

3.2.2. Logistička regresija

Logistička regresija algoritam koji se koristi kod nadziranog učenja za binarnu klasifikaciju kod kojeg se koristi sigmoidna funkcija koja kao ulaz koristi nezavisnu varijablu na temelju koji kao rezultat daje vrijednost između 0 i 1. Na primjer ako imamo dvije izlazne klase i rezultat funkcije za neki ulaz je veća od 0.5 onda taj ulaz pripada drugoj klasi, u protivnom pripada prvoj klasi. Izlaz iz ovog algoritma mora biti diskretna vrijednost npr. da ili ne, 0 ili 1, istina ili laž, ali umjesto točne vrijednosti daje vjerojatnosti između 0 i 1. Za razliku od linearne regresije koja je funkcija koja opisuje pravac, kod logističke regresije funkcija ima oblik slova "S". Ova funkcija preslikava bilo koju ulaznu vrijednost u vjerojatnost unutar raspona 0 i 1. Ako definiramo ulazne neovisne varijable kao

$$\mathbf{A} = \begin{bmatrix} X_{11} & \cdots & X_{1m} \\ \vdots & \ddots & \vdots \\ X_{n1} & \cdots & X_{nm} \end{bmatrix}$$

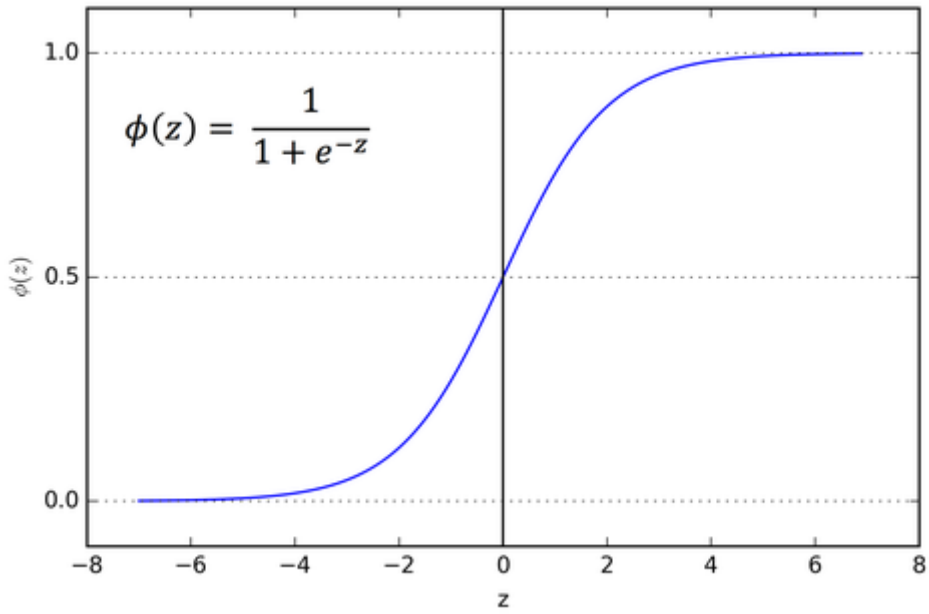
i ovisnu varijablu Y kao binarnu vrijednost iz nekog skupa npr. 0 i 1

$$Y = \begin{cases} 0, & ako\ je\ klasa\ 0 \\ 1, & ako\ je\ klasa\ 1 \end{cases}$$

zatim primjenjujemo višelinearnu funkciju na ulazne varijable X

$$z = (\sum_{i=1}^{n} w_i x_i) + b$$

x_i označava i-tu vrijednost iz X, w_i je koeficijent iz [w_1, w_2,w_3, ..., w_i], a b predstavlja koeficijent pristranosti. Zatim se koristi sigmoidna funkcija gdje se za ulaz z dobije vjerojatnost između 0 i 1 ili drugim riječima rezultat klase Y. [43]

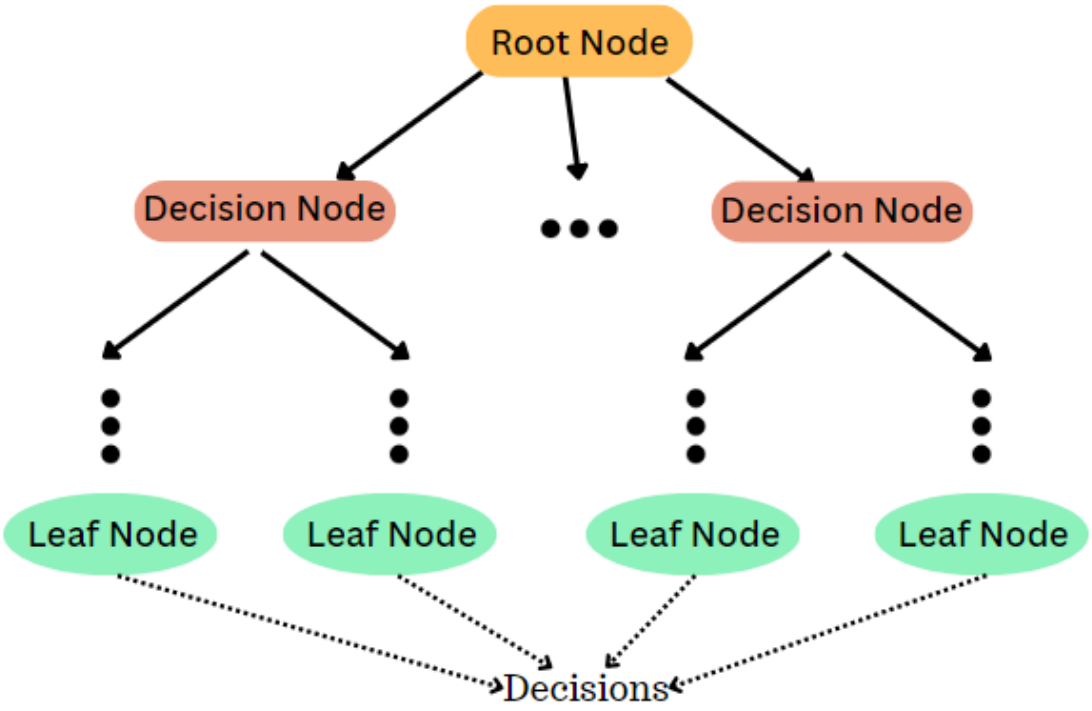


Slika 10. Sigmoidna funkcija (Izvor: [44])

3.2.3.Stabla odlučivanja

Stabla odlučivanja su algoritam nadziranog učenja koji se koristi kod klasifikacije i regresije. Struktura nalik stablu sastoji se od korijenskog čvora, grana, unutarnjih čvorova i čvorova listova. Korijenski čvor stabla predstavlja početni izbor ili značajku iz koje se stablo grana, najviši je čvor. Čvorovi u stablu su izbori određeni vrijednostima određenih atributa. Iz svakog internog čvora postoje grane koje idu do drugih čvorova. Svaki interni čvor testira neki atribut i svaka grana odgovara nekoj vrijednosti tog atributa. Grane su veze između čvorova koje pokazuju kako se donose odluke kao odgovor na pravilo ili uvjet koji se koristi za određivanje načina na koji se podaci trebaju podijeliti u čvoru. Na kraju svaki lisni čvor predstavlja konačnu vrijednost i rezultat algoritma. Stablo odlučivanja formira se tako da se

rekurzivno podijele podaci na temelju vrijednosti različitih atributa, algoritam odabire najbolji atribut za dijeljenje na svakom unutarnjem čvoru. Ovakav proces dijeljenja nastavlja se sve dok se ne ispuni kriterij zaustavljanja poput postizanje maksimalne visine stabla ili postojanje minimalnog broja lisnih čvorova. Ovakve hijerarhijske strukture pružaju jednostavnu vizualizaciju i mogu prikazati složene procese donošenja odluka, te se mogu lako prilagoditi različitim skupovima podataka. Preporučljivo je da su vrijednosti atributa kategoričke, ako su kontinuirane tada se prije izgradnje modela diskretiziraju. Glavni izazov kod kreiranja stabla odlučivanja je odabir atributa za korijenski čvor na svakoj razini stabla. Kod odabira atributa koriste se dvije mjere odabira, dobitak informacije i Ginijev indeks. Kada neki čvor koristimo za podjelu u manje skupove mijenja se entropija. Dobitak informacije je mjera ove promjene u entropiji. Ginijev indeks je metrika za mjerenje koliko često bi nasumično odabrani element bio netočno identificiran. To je mjera nejednakosti ili nečistoće distribucije i kreće se u rasponu od 0 do 0,5 gdje se preferira atribut sa što nižim indeksom. Izračunava se zbrajanjem kvadrata vjerojatnosti svakog ishoda u distribuciji i oduzimanjem tog rezultata od 1. [45]



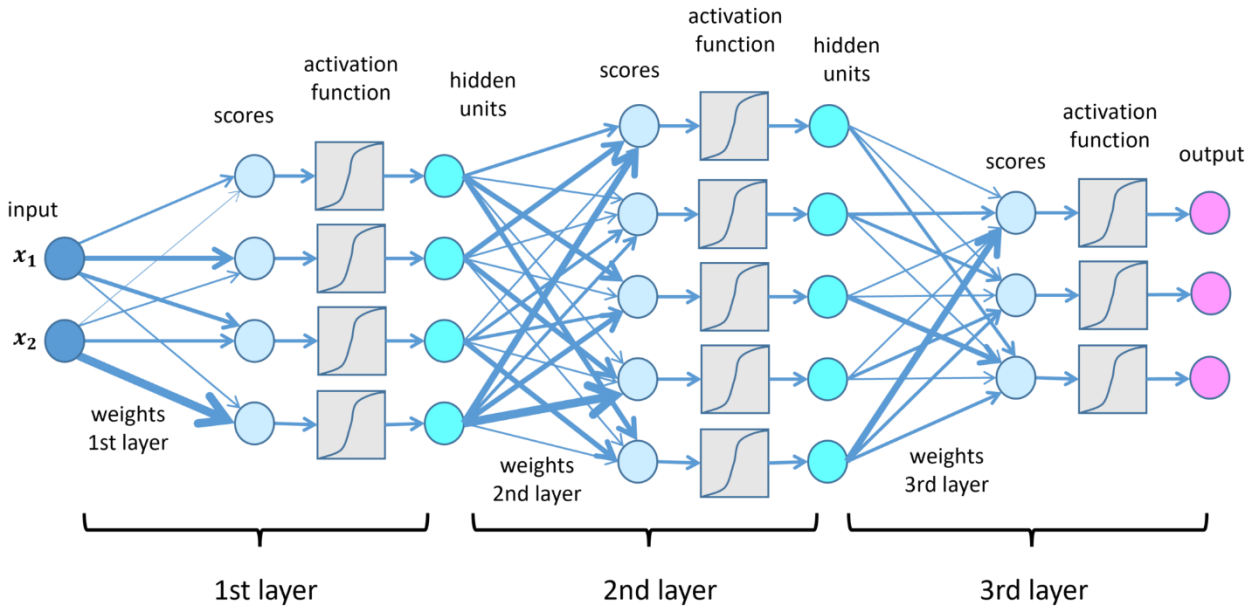
Slika 11. Struktura stabla odlučivanja (Izvor: [46])

3.2.4. Neuronske mreže

Neuronske mreže su model strojnog učenja koji pokušava oponašati složenu strukturu ljudskog mozga, sastoji se od slojeva međusobno povezanih čvorova koji predstavljaju neurone za obradu podataka i uče na temelju podataka. Neuronske mreže sposobne su učiti i identificirati obrasce izravno iz podataka bez unaprijed definiranih pravila. Strukturu čine neuroni koji primaju ulazne podatke i upravljani su pragom i funkcijom aktivacije, veze između neurona koje prenose informacije i regulirane su težinama, težine koje su parametri koji određuju snagu veza, i funkcije propagacije koje pomažu u obradi i prijenosu podataka kroz slojeve mreže. Mreža uči tako da se podaci unose u mrežu i na temelju trenutačnih parametara mreža generira izlaz, zatim se iterativno poboljšava izlaz prilagođavanjem težina. Sa svakom prilagodbom, odgovor mreže se razvija, omogućujući joj da se učinkovito prilagodi različitim zadacima ili okruženjima. U mreži postoji ulazni sloj neurona koji prima ulazne podatke, svaki ulazni neuron odgovara jednoj značajki u ulaznim podacima. Zatim slijedi jedan ili više skrivenih slojeva koji obavljaju većinu rada, svaki sloj sastoji se od neurona koji pretvaraju ulazne podatke u nešto što sljedeći sloj može koristiti kao ulaze. Na kraju je izlazni sloj koji daje rezultate iz modela ovisno o zadatku npr. klasifikacija ili regresija. Kada se podaci unesu u mrežu oni prolaze kroz nju u procesu širenja naprijed. Svaki neuron u sloju prima ulaze koji se množe s težinama povezanim s vezama. Umnošci se zbrajaju i dodaje se pristranost što se može opisati formulom

$$z = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b$$

gdje w predstavlja težinu veze, x predstavlja ulazni podatak, a b je pristranost. Rezultat z se zatim unosi u aktivacijsku funkciju koja određuje izlaz iz neurona. Nakon širenja naprijed mreža mjeri razliku između stvarnog i predviđenog izlaza s pomoću funkcije gubitaka. Zatim slijedi širenje unatrag kod kojeg mreža izračunava gradijente funkcije gubitka s obzirom na svaku težinu i pristranost u mreži kako bi se otkrilo koliko svaki dio pogreške izlaza ovisi o pojedinoj težini i pristranosti. Nakon što se izračunaju gradijenti ažuriraju se težine i pristranosti s pomoću nekog algoritma optimizacije. Veličina i brzina promjene u svakom širenju određuju se brzinom učenja mreže. Ovaj proces se ponavlja i s vremenom se smanjuje gubitak, a predviđanja mreže postaju točnija. Kroz ove korake neuronske mreže mogu prilagoditi svoje parametre kako bi poboljšale izlaze i rezultate. [47]

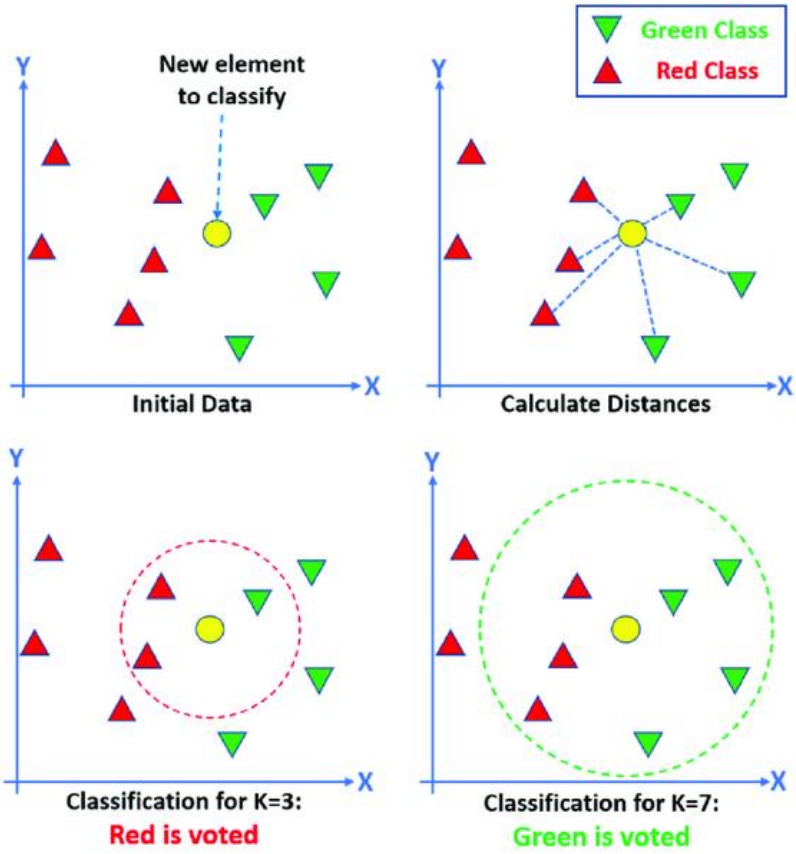


Slika 12. Struktura neuronske mreže (Izvor: [48])

3.2.5.K-najbližih susjeda

Algoritam K-najbližih susjeda (eng. *K-nearest neighbors - KNN*) nadzirana je metoda strojnog učenja koja se koristi za rješavanje problema klasifikacije i regresije. Jedan je od najosnovnijih, ali bitnih klasifikacijskih algoritama u strojnom učenju i nalazi veliku primjenu u prepoznavanju obrazaca, rudarenju podataka i otkrivanju upada. Može se koristiti u scenarijima iz stvarnog života budući da je neparametarski, što znači da ne donosi nikakve temeljne pretpostavke o distribuciji podataka. Radi na principu sličnosti, predviđa oznaku ili vrijednost nove podatkovne točke tako da uzima u obzir oznake ili vrijednosti svojih K najbližih susjeda u skupu podataka za treniranje. Za određivanje najbližih grupa ili najbližih susjeda koristimo neku formulu za izračun udaljenosti između točaka u podacima na primjer Manhattan udaljenost, Minkowski udaljenost ili Euklidska udaljenost. Za algoritam je jako bitan odabir vrijednosti broja K. Ovaj broj se odabire na temelju ulaznih podataka, ako u njima ima velikih odstupanja onda bi veća vrijednost za K bila bolja. Preporučuje se odabir neparne vrijednosti za K da bi se izbjegle veze kod klasifikacije. Algoritam radi tako da se predviđa izlaz ili rezultat na temelju sličnosti oznake ili vrijednosti K najbližih susjeda u skupu podataka. Prvo se odabire vrijednost za K koji predstavlja broj najbližih susjeda koji će se uzeti u obzir prilikom predviđanja. Nakon toga izračunava se udaljenost između svake podatkovne točke u skupu i ciljne točke kako bi se odredilo K najbližih susjeda. U slučaju da se radi klasifikacija u grupe ili klase određuje se nova grupa ili klasa za promatranu točku na temelju grupe ili klase većine najbližih K susjeda. Ako se radi regresija onda se oznaka grupe ili klase izračunava uzimanjem prosjeka ciljnih vrijednosti K najbližih susjeda. Najveća prednost ovog algoritma je što je lagan

za implementaciju no algoritam se suočava s teškoćama ispravnog klasificiranja podatkovnih točaka kada je dimenzionalnost prevelika. [49]

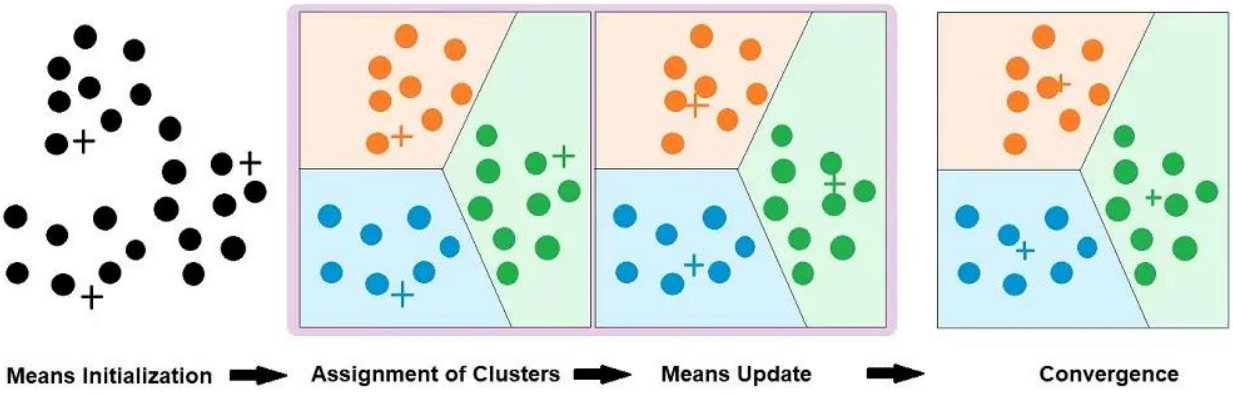


Slika 13. Klasifikacija za različite vrijednosti K (Izvor: [50])

3.2.6.K-srednje vrijednosti

K-srednje vrijednosti (eng. *K-Means Clustering*) je algoritam nenadziranog strojnog učenja koji grupira neoznačeni skup podataka u različite klastere. Ovaj algoritam grupira podatkovne točke jednom on K klastera ovisno o njihovoj udaljenosti od središta tog klastera. Radi tako da na početku nasumično dodjeljuje središta klastera u prostoru te zatim svaku točku dodjeljuje jednom od kreiranih klastera na temelju udaljenosti od središta klastera. Kad se točka dodjeli jednom od klastera dodjeljuju se nove težišne točke klastera. Ovaj proces provodi se iterativno sve dok se ne pronađe dobar klaster. U slučaju da broj klastera K nije unaprijed definiran potrebno ga je optimalno definirati. Algoritam je brži u usporedbi s drugim metodama grupiranja i kada najbolje radi kada su podaci dobro odvojeni, a nije prikladan kada podatkovne točke u ulaznim podacima imaju preklapanja. Kao i kod algoritma K-najbližih susjeda cilj mu je podijeliti ulazne podatke na klastere ili grupe tako da su točke unutar svakog klastera međusobno usporedive i drugačije od točaka unutar drugih klastera. Rezultat može ovisiti o početnim klasterima. Budući da je algoritam brz, uobičajeno ga je pokretati više puta s različitim početnim uvjetima. Pronalaženje optimalnog broja klastera K ključni je korak kako bi se

osiguralo da rezultati klasteriranja budu korisni. Postoje različite tehnike određivanja odgovarajućeg broja klastera. [51]

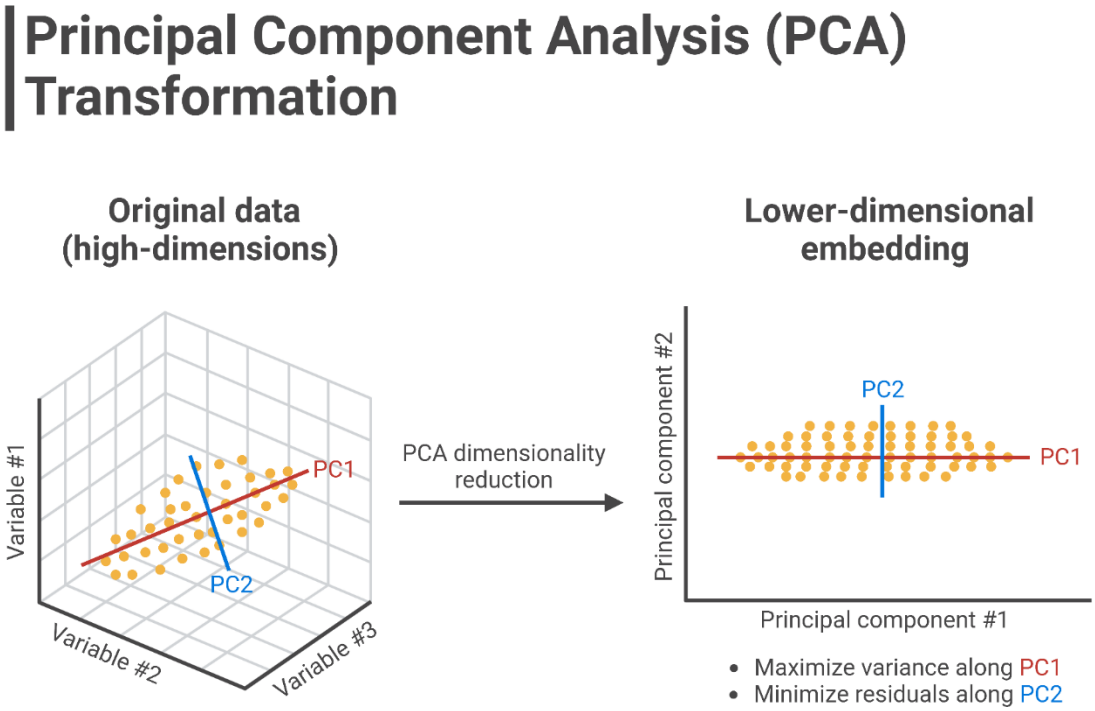


Slika 14. Koraci algoritma K-srednje vrijednosti (Izvor: [52])

3.2.7. Analiza glavnih komponenti

Analiza glavnih komponenti (eng. Principal Component Analysis - PCA) je algoritam nenadziranog učenja kojim se smanjenje dimenzionalnosti skupa podataka pronalaženjem novog skupa varijabli, manjeg od izvornog skupa varijabli, zadržavajući većinu informacija o uzorku i korisnih za regresiju i klasifikaciju podataka. Koristi za ispitivanje međuodnosa između skupa varijabli. Glavni cilj ovog algoritma je smanjiti dimenzionalnost skupa ulaznih podataka, ali uz očuvanje najvažnijih obrazaca i odnosa između varijabli bez znanja o ciljnim ili izlaznim varijablama. Ova tehnika smanjuje dimenzionalnosti tako da identificira skup ortogonalnih osi, zvanih glavne komponente, koje obuhvaćaju maksimalnu varijancu u podacima. Glavne komponente su linearne kombinacije izvornih varijabli u skupu ulaznih podataka i poredane su prema opadajućem redoslijedu važnosti. Ukupna varijanca koju obuhvaćaju sve glavne komponente jednaka je ukupnoj varijanci u izvornom skupu podataka. Prva glavna komponenta obuhvaća najviše varijacija u podacima, a svaka sljedeća glavna komponenta obuhvaća maksimalnu varijancu koja je ortogonalna prvoj glavnoj komponenti. Pretpostavlja se da se informacije prenose u varijaciji značajki, što je veća varijacija u značajki to značajka nosi više informacija. Može se koristiti za vizualizaciju podataka iscrtavanjem visokodimenzionalnih podataka u dvije ili tri dimenzije što olakšava njihovu interpretaciju, odabir značajki za identifikaciju najvažnijih varijabli u skupu podataka, i kompresiju podataka za smanjenje veličine skupa podataka bez gubitka važnih informacija. Provodi se tako da se najprije standardizira svaka značajka kako bi imala srednju vrijednost 0 i standardnu devijaciju 1. Zatim se izračunava matrica kovarijance za sve značajke pri čemu pozitivna kovarijanca označava ako se jedna značajka povećava onda se povećava i povezana značajka, negativna kovarijanca označava da se pri povećanju jedne značajke druga smanjuje, dok nule

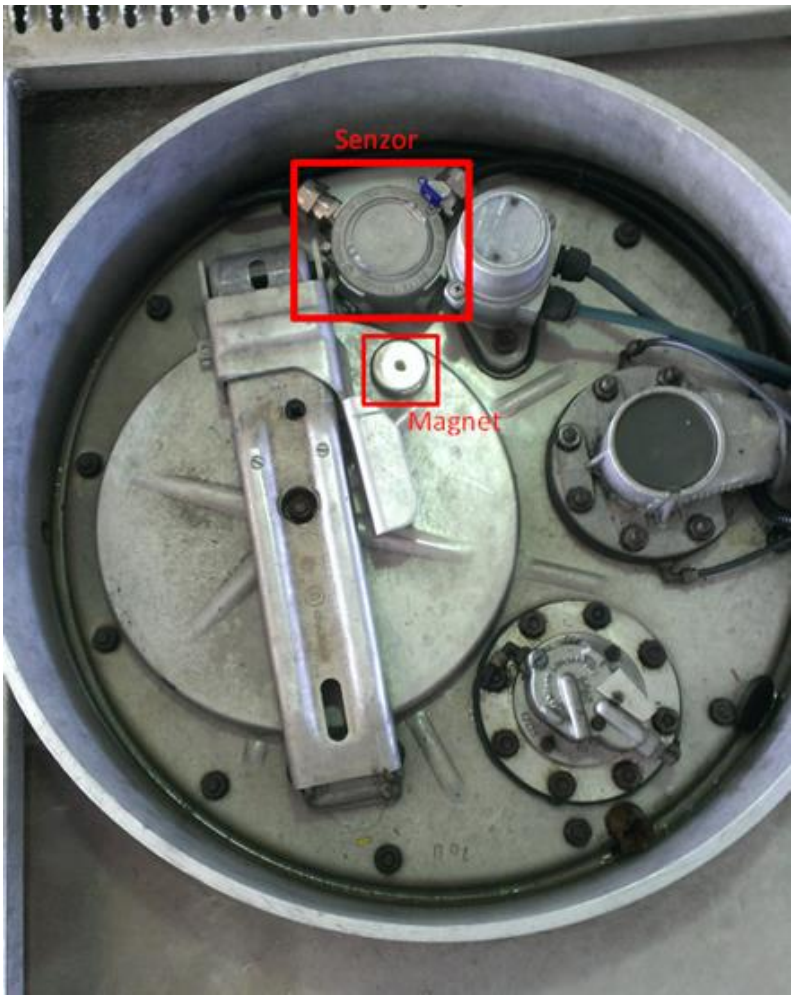
označavaju da značajke nisu direktno povezane. U posljednjem koraku izračunavaju se svojstvene vrijednosti i svojstveni vektori matrice kovarijance kako bi se identificirale glavne komponente. Na taj način koristi se linearna transformacija kako bi se očuvale najveće varijance u podacima s najmanjim brojem dimenzija. Smanjenje dimenzionalnosti postiže se zadržavanjem samo onih osi (dimenzija) koje čine većinu varijance i odbacivanjem svih ostalih. [53]



Slika 15. PCA smanjenje dimenzionalnosti (Izvor: [54])

4. Analiza MagSense Fuel Tank senzora

Postojeći MagSense Fuel Tank senzor (skraćeno *MSFT*) ugrađuje se na cisternu, po jedan uz svaki poklopac za utakanje goriva, kako bi se mogao vršiti nadzor otvaranja i zatvaranja poklopaca. Na sam poklopac ugrađuje se magnet, te je u zatvorenom stanju poklopca vrijednost magnetskog polja značajno veća od uobičajene vrijednosti zemljinog polja. Ovo predstavlja bazu za metodu detekcije statusa poklopca. Funkcija samog MSFT senzora je očitavanje podataka o magnetnom polju u sve tri osi s velikom brzinom od 75 Hz, te slanje tih podataka putem CAN sučelja do kontrolnog uređaja. Na kontrolnom uređaju obrađuju se mjerenja s MSFT senzora kako bi se ustanovio status otvorenosti poklopca. Ovdje se ne detektiraju samo jednostavne vrijednosti u smislu jako ili slabo magnetno polje, nego se tijekom instalacije senzora provodi proces kalibracije, pri čemu se pohranjuju svi vektori magnetskog polja tijekom kompletnog otvaranja i zatvaranja poklopca. Na ovaj način sustav ima dovoljno informacija da osim otvorenog i zatvorenog stanja, može prepoznati i stanja koja upućuju na manipulaciju sustava (vektor polja izvan svih dozvoljenih). [55]



Slika 16. Ugrađeni MSFT uređaj (Izvor: [55])

Na slici 16. prikazan je trenutačni način ugradnje MSFT uređaja na jedan od poklopaca cisterne. Potrebno je redizajnirati postojeći MSFT senzor, kako bi se zamijenile zastarjele komponente (primarno magnetometar HMC5883), te modernizirala platforma. Primarni cilj redizajna je u potpunosti zadržati kompatibilnost s načinom rada prethodnog sustava kako bi se izbjegle bilo kakve promijene na razini kontrolnog uređaja, dok je sekundarni cilj unaprijediti rad cijelog sustava i potencijalno pronaći bolji pristup izvedbi detekcije stanja.

Kao alternativno rješenje predlaže se korištenje senzora LSM6DSOX s ugrađenim strojnim učenjem koji bi u eliminirao potrebu za ugradnjom magneta na poklopac pa bi se detekcija otvaranja i zatvaranja poklopca vršila na temelju podataka koje senzor detektira s pomoću ugrađenog žiroskopa, akcelerometra i 6-osnog senzora. Senzor bi se unaprijed trenirao na uzorcima podataka koji predstavljaju karakteristične pokrete otvaranja i zatvaranja poklopca. Kroz proces strojnog učenja, sustav bi analizirao obrasce promjena u kutnoj brzini, ubrzanju i orijentaciji kako bi prepoznao specifične sljedove povezane s tim radnjama. Nakon završetka treniranja, senzor bi mogao autonomno detektirati otvaranje i zatvaranje poklopca u stvarnom vremenu, koristeći algoritme za prepoznavanje uzoraka, bez potrebe za dodatnim vanjskim komponentama poput magneta. Ovo rješenje ne samo da smanjuje kompleksnost hardvera, već omogućuje i fleksibilnije prilagođavanje različitim dizajnima i uvjetima primjene. Ovakav uređaj mogao bi se koristiti za detekciju bilo koje druge aktivnosti u raznim industrijama npr. praćenje gibanja vozila, praćenje brodova, detekcija vibracija/nagiba u industrijskim postrojenjima ili bilo koje druge aktivnosti koja se može detektirati s pomoću ugrađenog žiroskopa, akcelerometra i 6-osnog senzora. U nastavku rada bit će izrađen prototip ovakvog uređaja s pomoću STEVAL-MKSBOX1V1 kompleta za razvoj (eng. *devkit*) koji uključuje STM32 mikroprocesor i LSM6DSOX senzor.

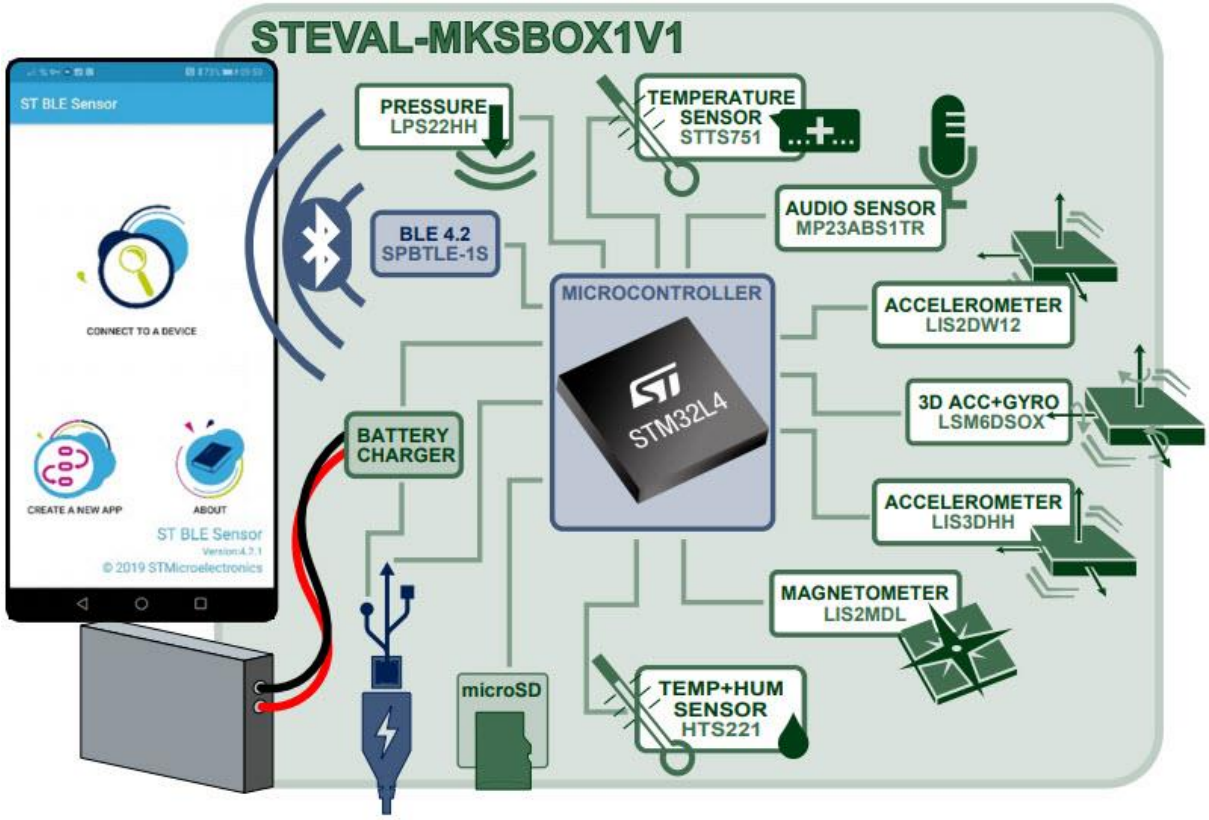
5. Komplet za razvoj STEVAL-MKSBOX1V1

MKSBOX1V1 je kompaktan komplet za razvoj namijenjen za brzo prototipiranje i razvoj aplikacija temeljenih na podacima iz okoline i podacima senzora za kretanje. Dolazi u kućištu s IP54 razinom zaštite, Li-Po baterijom od 500 mA/h, microSD karticom od 8 GB, te je kompatibilan sa ST BLE Sensor aplikacijom za povezivanje s pomoću Bluetootha s kompletima za razvoj. Ova aplikacija omogućuje prikaz različitih pogleda na temelju vrsti podataka koji se čitaju, prikaz dijagrama, prepoznavanje aktivnosti, izvoz podataka i mnoge druge funkcionalnosti ovisno o kompletu s kojim se povezuje. Unutar kompleta nalazi se kompaktna pločica s 32-bitnim STM32L4R9ZIJ6 mikroprocesorom, digitalnim senzorom temperature STTS751, 6-osnim inercijskim senzorom LSM6DSOX, 3-osnim akcelerometrom LIS2DW12 i LIS3DHH, 3-osnim magnetometrom LIS2MDL, audio senzorom MP23ABS1, senzorom vlage HTS221, senzorom tlaka LPS22H, Bluetooth modulom BlueNRG-M2, i sučeljem za programiranje i profesionalni razvoj aplikacija. [56]

Slika 17. STEVAL-MKSBOX1V1 komplet (Izvor: [57])

Na procesoru je inicijalno instalirana aplikacija koja na jednostavan način omogućuje korištenje s funkcionalnošću za detektiranje aktivnosti. Preko ST BLE Sensor aplikacije moguće je odabrati jedan od tri načina rada. U prvom načinu rada (*Entry*) moguće je koristiti predefinirane aktivnosti kako bi se istražile mogućnosti senzora. Aplikacije koje je inicijalno moguće koristiti su brojač koraka koji omogućuje konfiguriranje LSM6DSOX senzora za praćenje hodanja i brzine trčanja te prikaz informacija u realnom vremenu, detekcija plača djeteta s pomoću konfiguriranja MP23ABS1 audio senzora koji detektira ljudski glas i šalje

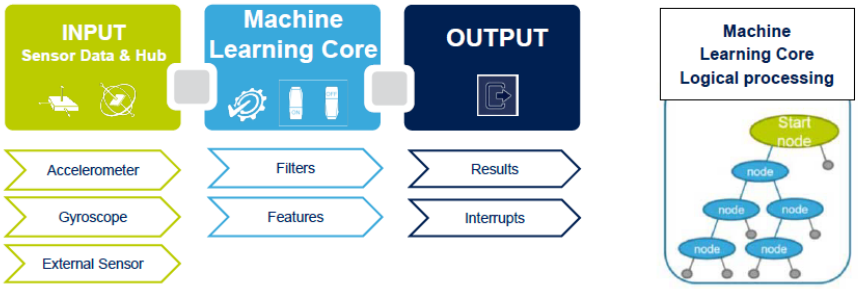
upozorenje na pametni telefon ili aktivira LED svjetla na pločici, barometar aplikacija koja omogućuje konfiguraciju senzora temperature STTS751, senzora tlaka LPS22HH, i senzora vlažnosti HTS221 za praćenje informacija o okolišu u stvarnom vremenu na pametnom telefonu ili prikupljanje i prikaz podataka u obliku grafa, aplikacija za praćenje vozila/robe koja omogućuje odabir i konfiguraciju odgovarajućih senzora okoliša i kretanja za bilježenje odabranih uvjeta prijevoza i skladištenja robe tijekom vremena, aplikacija za praćenje vibracija koja omogućuje konfiguriranje LSM6DSOX akcelerometra i učenje senzora na normalan rad motorizirane kućne ili industrijske opreme, te zatim praćenje opreme za nenormalne vibracije u svrhu prediktivnog održavanja, aplikacija za kompas i inklinaciju koja omogućuje konfiguriranje akcelerometra i žiroskopa LSM6DSOX i LIS2MDL magnetometra za praćenje smjera i nagiba u stvarnom vremenu i prikaz informacija tijekom vremena, te kompenzirana aplikacija magnetometra koja omogućuje izradu dodatnih aplikacija iz izlaza magnetometra i algoritam povezivanja senzora za kompenzaciju smetnji od vanjskih magnetskih polja. U drugom načinu rada (*Expert*) moguće je preko mobilne aplikacije kreirati vlastite aplikacije u jednostavnom grafičkom sučelju povezivanjem i konfiguracijom različitih senzora, definiranjem izlaza i događaja te primjenu algoritama za obradu podataka. Treći (*Pro*) način rada namijenjen je razvojnim programerima za izradu potpuno prilagođenih programskih rješenja kako bi iskoristili puni potencijal dostupnih senzora. [56]



Slika 18. STEVAL-MKSBOX1V1 blok dijagram (Izvor: [56])

5.1. Senzor LSM6DSOX

LSM6DSOX je sustav u paketu (eng. *System-in-package*) koji sadrži digitalni 3-osni akcelerometar i digitalni 3-osni žiroskop, odlikuje ga niska potrošnja energije i ultrakompaktno kućište. Od ugrađenih funkcionalnosti na čipu podržava spremnik podataka od 9 kB koji omogućava kompresiju podataka dva do tri puta i mogućnost spremanja vremenskih oznaka, prekide (eng. *Interrupts*) na temelju događaja, funkcije pedometra, detekcije nagiba, detekcije pokreta, stroj konačnih stanja za akcelerometar, žiroskop, i vanjske senzore, jezgru za strojno učenje i integraciju od ukupno 6 senzora od kojih su 2 interna, a ostali eksterni. U kontekstu ovog rada može se istaknuti stroj konačnih stanja koji se može konfigurirati korisnički definiranim obrascima kretanja za generiranje signala prekida, moguće je neovisno programirati do 16 ugrađenih stojeva konačnih stanja za detekciju pokreta. Svaki od 16 stojeva konačnih stanja neovisno je i svaki ima svoje memorijsko područje i neovisno se izvršavaju. Prekid se generira kada se postigne krajnje stanje ili kada se izvrši neka određena naredba. Ugrađena jezgra za strojno učenje omogućuje premještanje nekih algoritama koji se izvode u aplikacijskom procesoru na senzor što uzrokuje smanjenje potrošnje energije. Logika jezgre omogućuje prepoznavanje obrazaca u podacima koji odgovaraju korisnički definiranim skupovima klasa. Tipični primjeri mogu biti detekcija aktivnosti poput trčanja, hodanja, vožnje i slično. Obrada se vrši na temelju podataka koji dolaze iz akcelerometra i žiroskopa senzora, ali moguće je spojiti i obraditi podatke vanjskog senzora. Ulazni podaci mogu se filtrirati korištenjem namjenskog konfigurabilnog proračunskog bloka koji sadrži filtre i značajke koje se izračunavaju u fiksnom vremenskom razdoblju koje definira korisnik. Obrada podataka u strojnom učenju temelji se na logičkoj obradi koja se sastoji od niza konfigurabilnih čvorova koje karakteriziraju uvjeti „*if-then-else*“ gdje se vrijednosti značajki uspoređuju s definiranim vrijednostima koje definiraju stablo odlučivanja. Moguće je konfigurirati do 8 neovisnih stabla koja se izvode istovremeno. Svako stablo može generirati do 16 rezultata pa ukupni broj čvorova može biti 256. Rezultati obrade strojnog učenja dostupni su u rezerviranim izlaznim registrima koji se mogu čitati iz aplikacijskog procesora u bilo kojem trenutku, a moguće je i konfigurirati generiranje prekida kada se promijeni rezultat. [58]



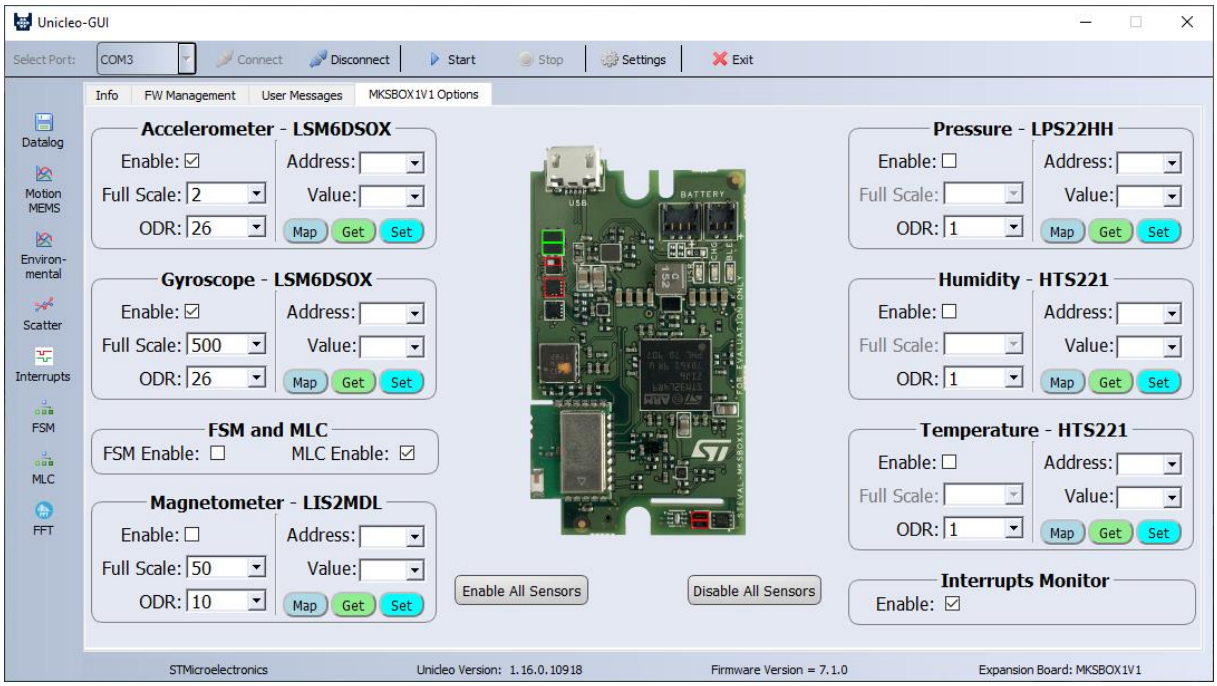
Slika 19. Tijek rada jezgre strojnog učenja (Izvor: [58])

6. Razvoj aplikacije za detekciju stanja

Izrada aplikacije za detekciju stanja sastoji se od dva dijela. Prvi korak je izrada modela strojnog učenja s pomoću STM-ovih alata Unicleo i Unico koji služe za prikupljanje podataka sa senzora, spremanje podataka, konfiguraciju i generiranje stabla odlučivanja, i testiranje modela prije upotrebe u aplikaciji. Zatim se generirani model može koristiti u aplikaciji za kako bi se konfigurirala jezgra za strojno učenje. Aplikacija će biti pisana u C programskom jeziku i sadržavat će sloj apstrakcije hardvera (eng. *Hardware Abstraction Layer - HAL*) koji će biti generiran s pomoću STM32CubeMX alata te će uključivati FreeRTOS okruženje. Drugi sloj bit će SPI upravljački program (eng. *Driver*) za komunikaciju sa senzorom putem SPI sučelja i aplikacijski kod za konfiguraciju senzora i detekciju stanja čitanjem relevantnih registara sa senzora. Detektirano stanje slat će se preko USB sučelja na vanjsku aplikaciju vizualizaciju. [59] [60]

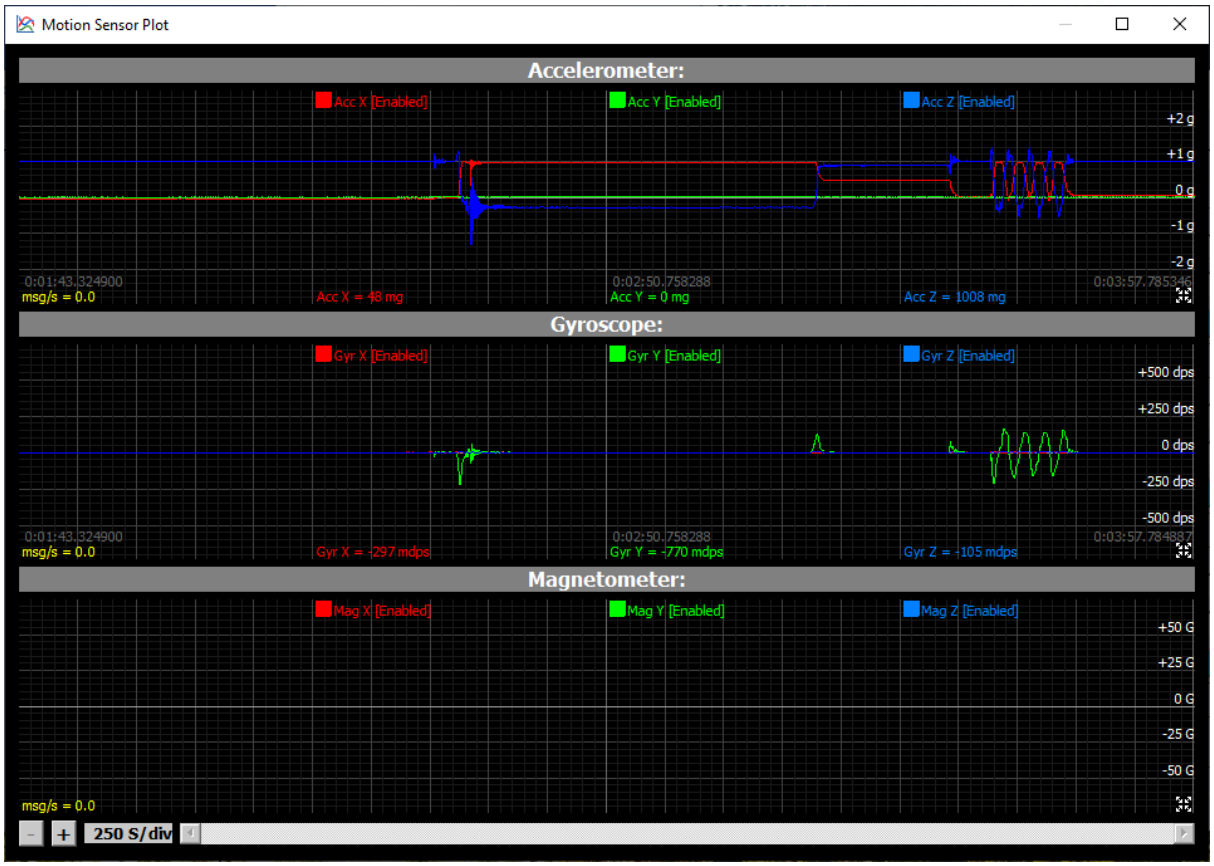
6.1. Izrada modela stabla odlučivanja

Prije razvoja same aplikacije za detekciju stanja potrebno je proći proces kojim se generira stablo odlučivanja koje će biti korišteno za konfiguriranje jezgre za strojno učenje. Za ovaj proces koriste se STM-ovi alati Unicleo i Unico. S pomoću Unicleo alata konfigurira se senzor te se podaci snimaju i spremaju kao csv datoteka. Potrebno je snimiti nekoliko uzoraka od kojih svaki karakterizira jedno stanje ili pokret koji se želi kasnije detektirati. Na slici 20. vidljiva je konfiguracija koja je postavljena prije snimanja podataka gdje su uključeni akcelerometar i žiroskop. Za akcelerometar odabrana je skala od 2 g i brzina snimanja od 26 Hz. Za žiroskop odabrana je skala od 500 stupnjeva u sekundi i brzina snimanja od 26 Hz. Budući da stanja i pokret koji se želi detektirati karakteriziraju niske brzine i akceleracija odabrane su niže vrijednosti kako bi senzor bio osjetljiviji te se tako postiže veća preciznost pri detekciji.



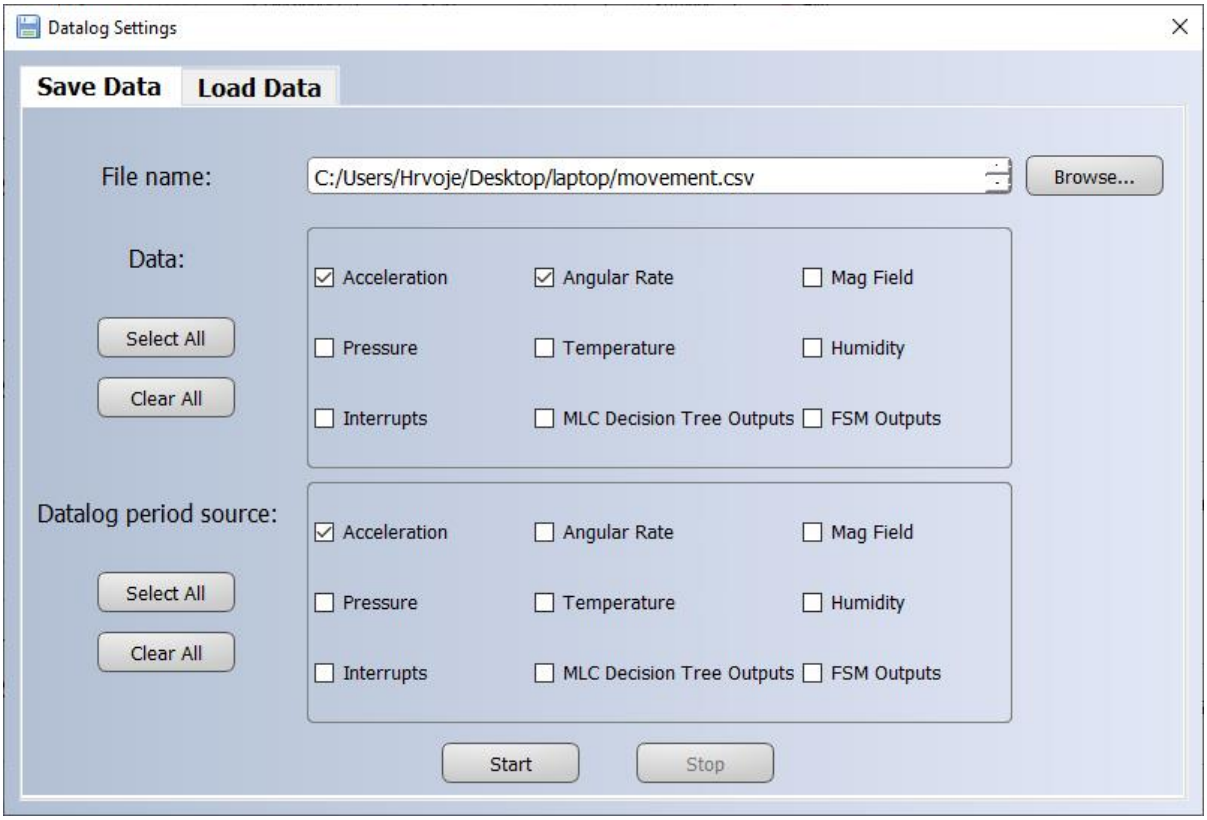
Slika 20. Konfiguracija senzora za prikupljanje podataka (Izvor: autorski rad)

Nakon konfiguracije senzora pritiskom na tipku *Start* započinje prikupljanje podataka sa senzora što je vidljivo na kratici *Motion MEMS* gdje se podaci prikazuju u stvarnom vremenu.



Slika 21. Graf s prikupljenim podacima (Izvor: autorski rad)

Da bi se podaci spremili za kasniju upotrebu potrebno je otići na kraticu *Datalog*, odabrati parametre koji se žele snimati i pokrenuti spremanje podataka dok je prikupljanje aktivno. Na slici 22. vidljivo je da su za spremanje odabrani podaci s akcelerometra i žiroskopa dok je za izvor periode snimanja odabran žiroskop. Za snimanje podataka komplet za razvoj bio je pričvršćen obostranom trakom za ekran laptopa kako bi se simulirali pokreti karakteristični za otvaranje i zatvaranje poklopca na cisterni. Nakon ovog koraka generirane su tri datoteke u vrijeme kad je senzor bio u pozicijama koje karakteriziraju otvoreno stanje, zatvoreno stanje, te pokret otvaranja i zatvaranja.



Slika 22. Odabir parametara za spremanje (Izvor: autorski rad)

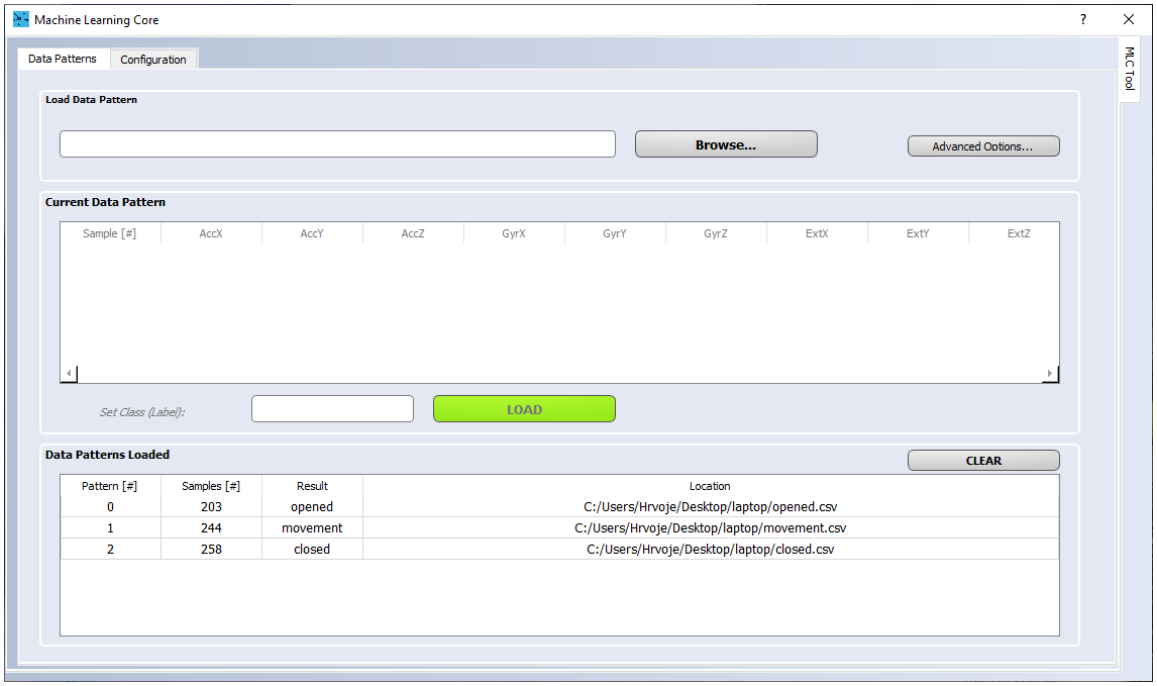
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	#Format version: 1.0														
2	time[us]	accX[mg]	accY[mg]	accZ[mg]	gyroX[mdp	gyroY[mdp	gyroZ[mdp	magnX[m	magnY[m	magnZ[m	press[hPa	temp[°C]	hum[%RH	newData	interrupts	fsr
3	1.75E+08	955	3	-282	-297	-752	-87							0x058		
4	1.75E+08	955	4	-283	-280	-770	-87							0x018		
5	1.75E+08	956	4	-281	-262	-700	-105							0x018		
6	1.75E+08	955	3	-278	-245	-700	-87							0x058		
7	1.75E+08	954	3	-284	-332	-980	-70							0x018		
8	1.75E+08	954	4	-287	-332	-752	-52							0x058		
9	1.75E+08	955	4	-280	-280	-560	-52							0x018		
10	1.75E+08	956	5	-281	-297	-805	-70							0x018		
11	1.75E+08	955	5	-279	-297	-665	-70							0x058		
12	1.75E+08	955	4	-281	-280	-892	-105							0x018		
13	1.75E+08	955	3	-287	-262	-910	-105							0x058		
14	1.75E+08	954	3	-282	-280	-577	-70							0x018		
15	1.75E+08	955	4	-281	-315	-717	-70							0x018		
16	1.75E+08	956	5	-282	-297	-752	-87							0x058		
17	1.75E+08	955	5	-279	-315	-700	-87							0x018		
18	1.75E+08	955	4	-283	-332	-875	-105							0x058		
19	1.75E+08	955	3	-285	-280	-805	-70							0x018		
20	1.75E+08	954	3	-282	-280	-647	-70							0x018		
21	1.75E+08	955	4	-282	-297	-735	-70							0x058		
22	1.75E+08	955	4	-281	-297	-735	-87							0x018		
23	1.75E+08	956	4	-280	-280	-787	-105							0x058		
24	1.75E+08	955	3	-284	-280	-857	-87							0x018		
25	1.75E+08	955	4	-285	-280	-787	-87							0x058		
26	1.76E+08	955	4	-283	-297	-682	-70							0x018		
27	1.76E+08	956	4	-280	-297	-682	-87							0x018		

Slika 23. Sadržaj csv datoteke prikupljenih podataka (Izvor: autorski rad)

Nakon što su prikupljeni podaci za sva stanja uvozimo datoteke u alat Unico s pomoću kojeg se generira stablo odlučivanja. Na sučelju alata potrebno je otići na kraticu *MLC* (eng. *Machine Learning Core*) na kojoj možemo uvesti svaku od prethodno generiranih csv datoteka i postaviti joj klasu koju ti podaci označavaju.

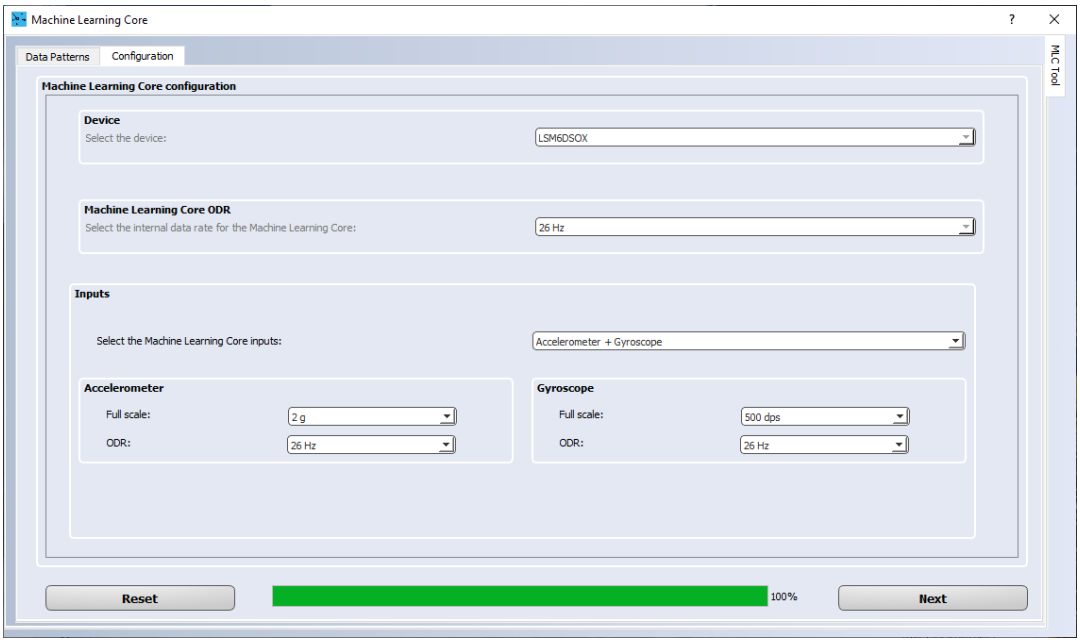
Slika 24. Sučelje alata Unico (Izvor: autorski rad)

Na slici 25. prikazane su unesene klase kojima su označene uvezene datoteke koje će biti rezultat stabla odlučivanja. Klasa *opened* označava otvoreno stanje poklopca, klasa *closed* označava zatvoreno stanje poklopca, dok klasa *movement* označava pokret otvaranja ili zatvaranja poklopca.



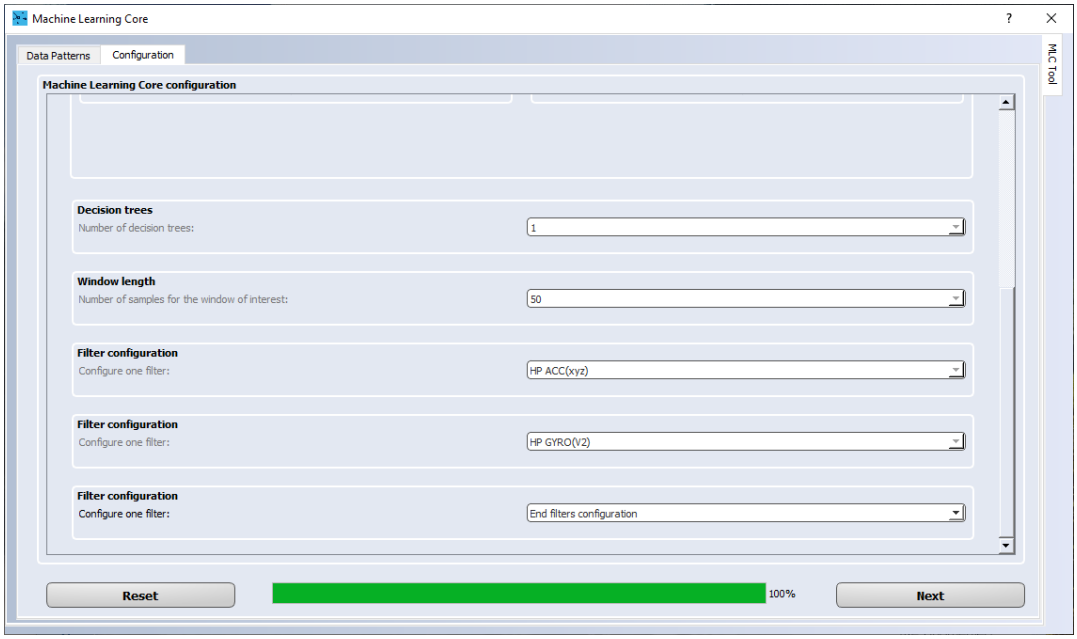
Slika 25. Uvoz i klasificiranje podataka (Izvor: autorski rad)

Nakon što smo uvezli sve datoteke potrebno je odabrati kraticu *Configuration* na kojoj kreće proces konfiguracije i generiranja stabla odlučivanja. Prvi korak je odabir senzora i postavljanje parametara prema kojima su snimljeni podaci. Na slici 26. vidljiv je odabir senzora LSM6DSOX, brzine snimanja podataka od 26 Hz, za ulaze odabrani su akcelerometar i žiroskop s istim postavkama koje su korištene kod snimanja podataka.



Slika 26. Postavke senzora kojim su prikupljeni podaci (Izvor: autorski rad)

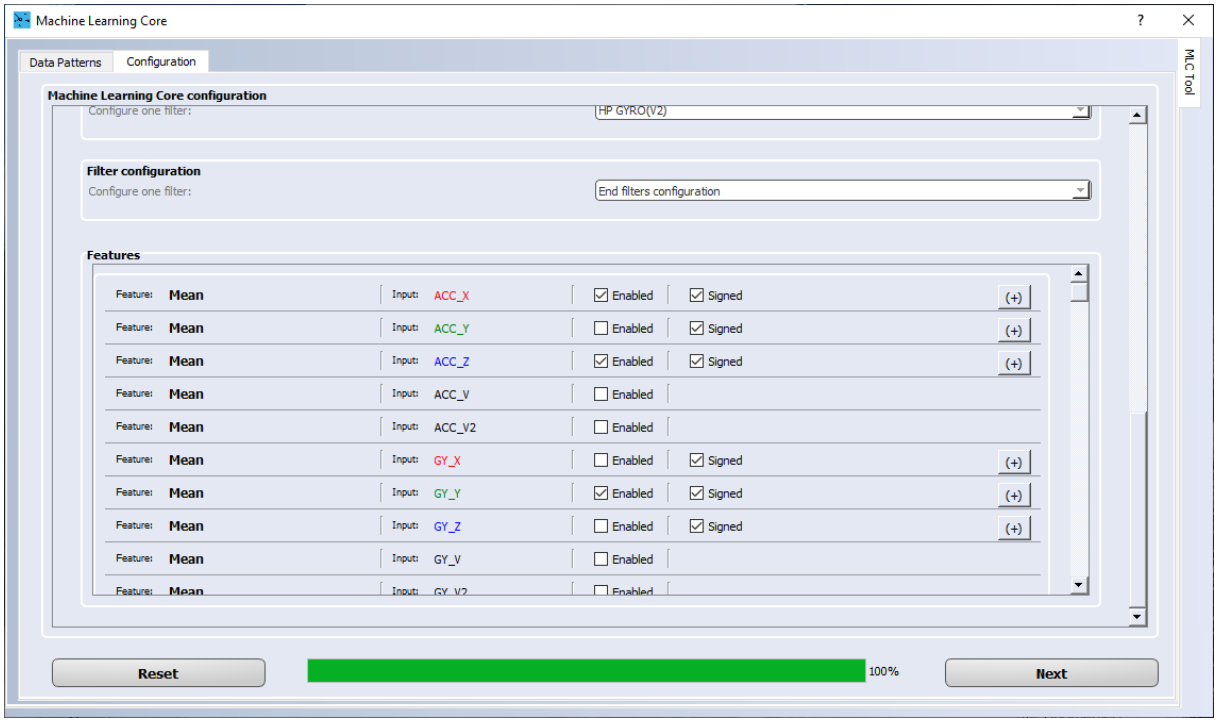
Pritiskom na gumb *Next* prelazi se na sljedeći korak u konfiguraciji. Na slici 27. u sljedećem koraku biramo broj stabla odlučivanja koje želimo generirati, broj uzoraka koji će označavati jedan interval pri detekciji, i filtere za pretprocesiranje podataka. Odabrano je generiranje jednog stabla odlučivanja, 50 uzoraka po intervalu te filter na X, Y, i Z osima akcelerometra i filter za brzinu na kvadrat za žiroskop.



Slika 27. Konfiguracija stabla odlučivanja i filtera (Izvor: autorski rad)

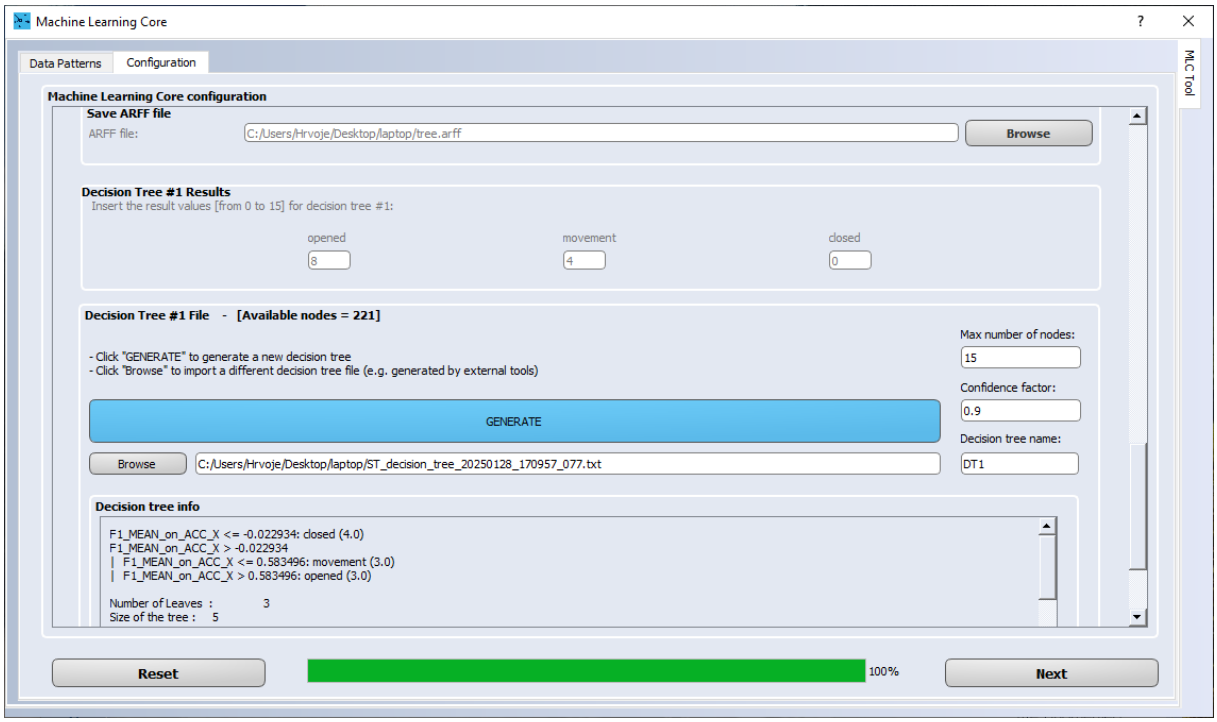
Sljedeći korak je odabir značajki koje smatramo da su bitne za stablo odlučivanja po kojima se najbolje razlikuju stanja. U ovom slučaju na slici 21. vidljivo je da su najveće razlike između stanja na X i Z osima akcelerometra te Y osi žiroskopa. Prema tome odabrane su

značajke medijan za X i Z osi akcelerometra i Y os žiroskopa, energija na Y osi žiroskopa, vrh do vrha (eng. *Peak to peak*) na X i Z osi akcelerometra i Y osi žiroskopa, minimum i maksimum na X i Z osi akcelerometra i Y osi žiroskopa.



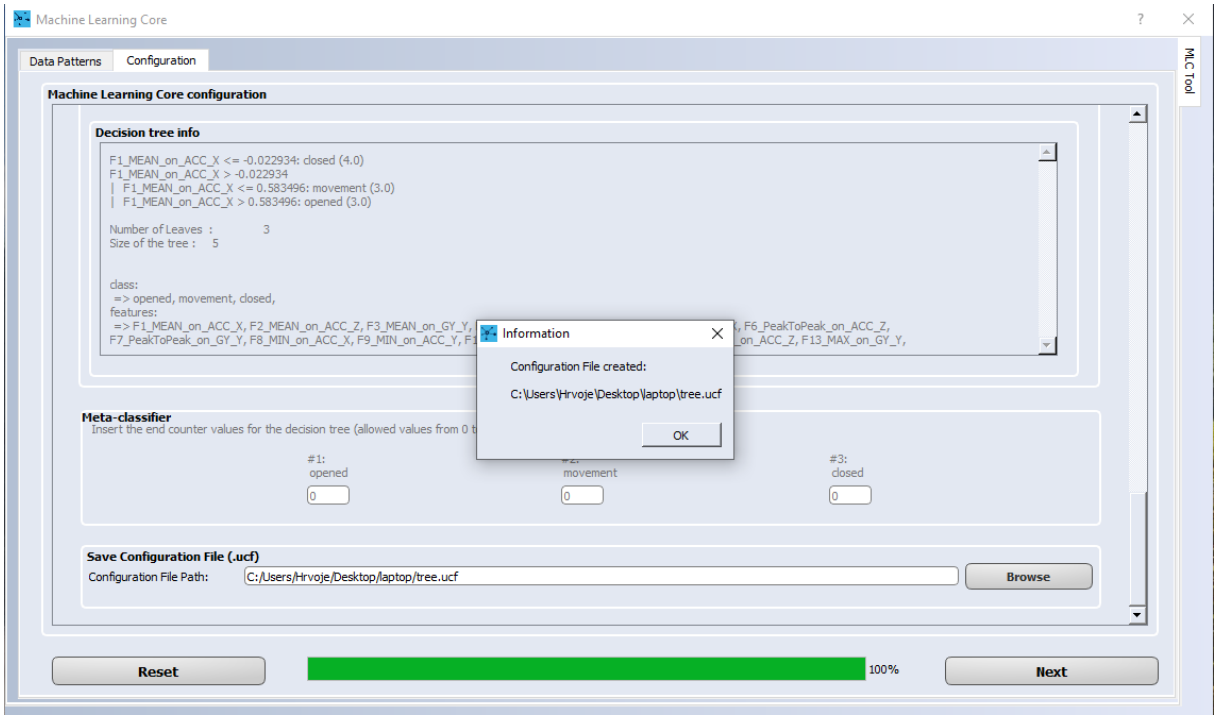
Slika 28. Odabir značajki za stablo odlučivanja (Izvor: autorski rad)

Nakon odabira značajki generira se arff datoteka koja sadrži konfiguraciju za odabrane značajke i relevantne podatke. Zatim se odabiru vrijednosti izlaza za svako detektirano stanje koje će biti upisane u registar rezultata jezgre za strojno učenje prilikom detekcije tog stanja. Odabrane su vrijednosti 8 za otvoreno stanje poklopca, 4 za pokret otvaranja i zatvaranja poklopca, i 0 za zatvoreno stanje poklopca. Na kraju ovog koraka generira se stablo odlučivanja formatu čitljive tekstualne datoteke.



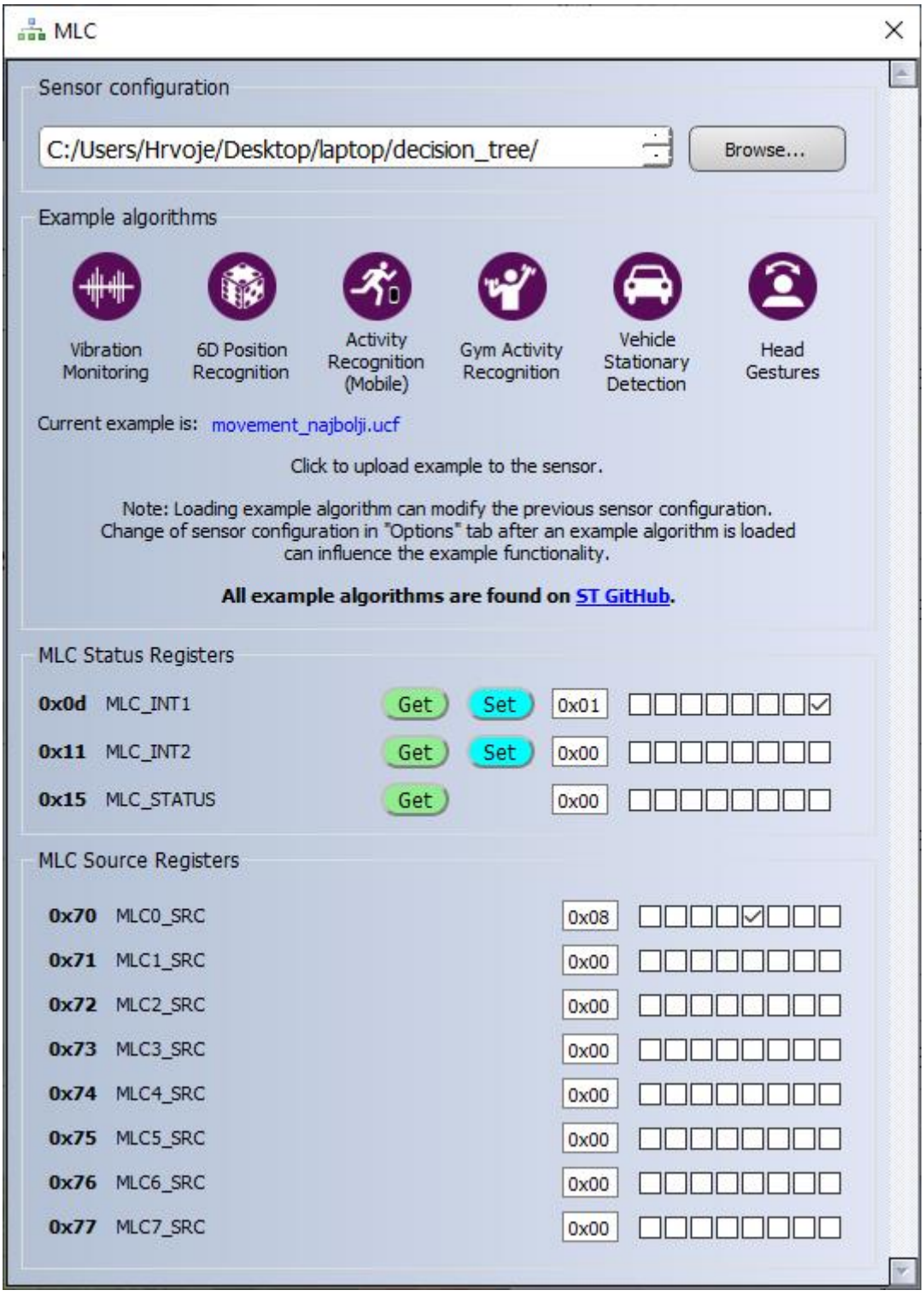
Slika 29. Odabir vrijednosti rezultata i generiranje stabla odlučivanja (Izvor: autorski rad)

Nakon što je generirano konačno stablo odlučivanja generira se ucf datoteka koja sadrži stablo odlučivanja u formatu pogodnom za konfiguriranje jezgre za strojno učenje.



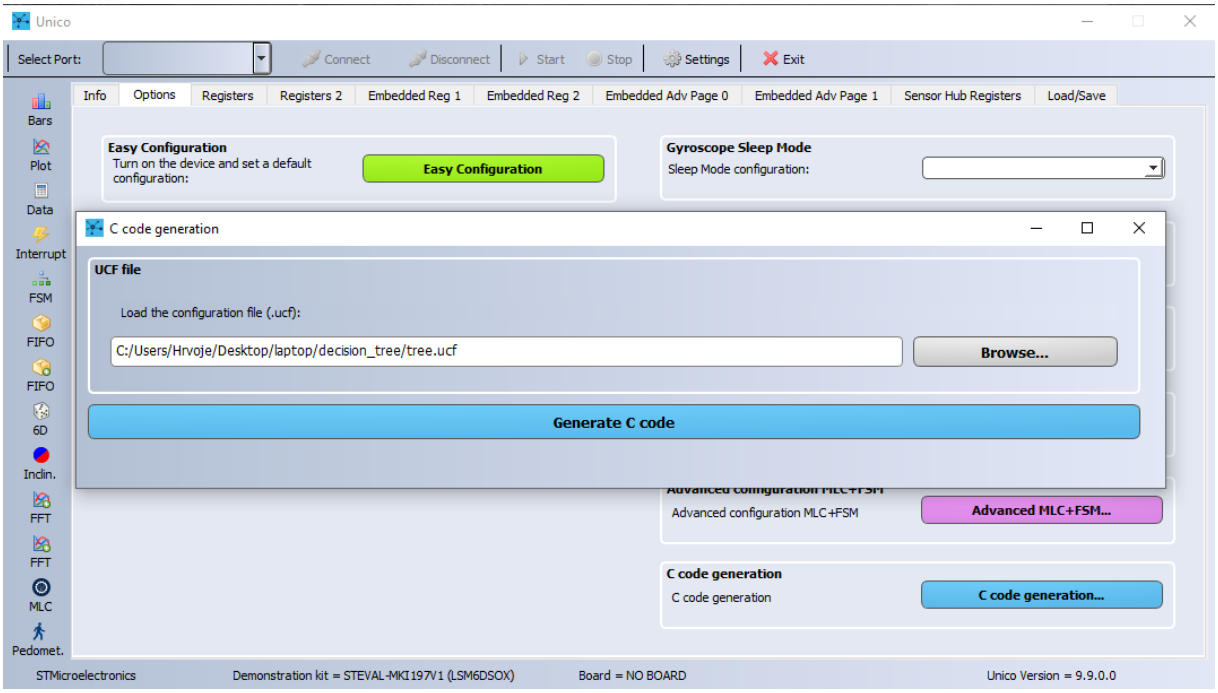
Slika 30. Generiranje ucf datoteke stabla odlučivanja (Izvor: autorski rad)

Kako bi jednostavno provjerili valjanost modela i detekciju stanja moguće je u alatu Unicleo odabrati kraticu *MLC* te uvesti generiranu ucf datoteku koja se zatim šalje na senzor te se rezultati i vrijednosti registara prikazuju na grafičkom sučelju. Ovime je model testiran i spreman za korištenje. Na slici 31. vidljivo je da model dobro detektira stanja na primjeru otvorenog stanja poklopca.



Slika 31. Testiranje stabla odlučivanja (Izvor: autorski rad)

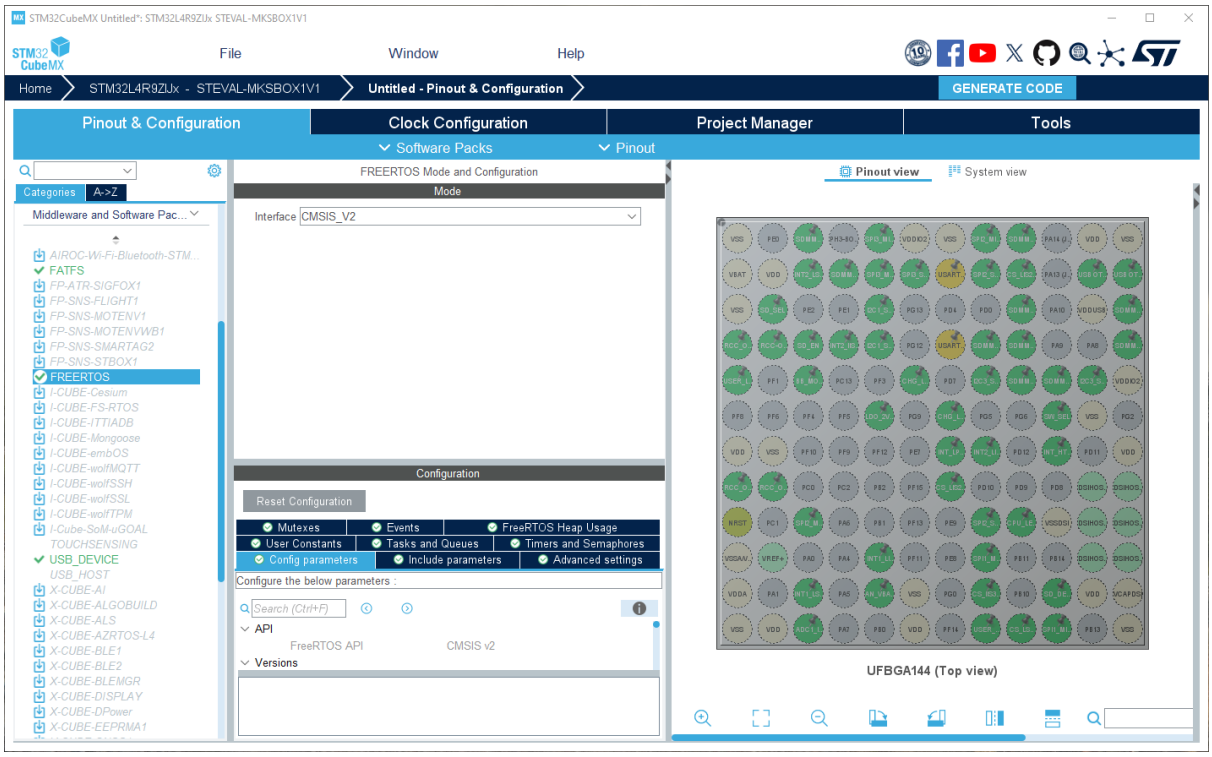
Za korištenje stabla odlučivanja u aplikaciji potrebno je još ucf datoteku pretvoriti u *header* datoteku koja sadrži stablo odlučivanja u obliku C polja koje se može programski zapisati na senzor s pomoću I2C ili SPI sučelja. U alatu Unico odabire se kratica *Options* na kojoj se nalazi opcija generiranja C koda iz ucf datoteke.



Slika 32. Generiranje C koda iz ucf datoteke (Izvor: autorski rad)

6.2. Aplikacija za detekciju stanja

Projekt za aplikaciju za detekciju stanja generiran je s pomoću STM32CubeMX alata. Ovaj alat služi za jednostavnu grafičku konfiguraciju STM32 mikrokontrolera i procesora te generiranje odgovarajućeg inicijalizacijskog C koda. Prvi korak je odabir mikrokontrolera ili kompleta za razvoj putem grafičkog sučelja i ugrađene baze podataka. Nakon što je odabran odgovarajući hardver moguće je interaktivno konfigurirati postavke za cijeli sustav poput postavki sata, perifernih uređaja, i općih ulaza i izlaza. Na kraju se pokreće generiranje koda koji odgovara odabranim postavkama što generira inicijalizacijski C kod za mikroprocesor i svu odabranu periferiju. [61]



Slika 33. Sučelje programa STM32CubeMX (Izvor: autorski rad)

Nakon što je projekt generiran moguće je koristiti generirane funkcije za interakciju s hardverom kao što je paljenje i gašenje ulaza i izlaza, čitanje i pisanje u dostupne registre, slanje i primanje podataka s konfigurirane periferije i konfiguracija dostupnih senzora. Ulazna korisnička funkcija u aplikaciju generirana je u main.c datoteci pod nazivom StartDefaultTask budući da se koristi FreeRTOS kao aplikacijsko okruženje. Prije poziva same ulazne funkcije u korisnički kod pokreće se main funkcija koja inicijalizira i konfigurira uređaj prema postavkama odabranim kod generiranja projekta. Korisnički kod koji je dodan u projekt nalazi se u Drivers/LSM6DSOX direktoriju gdje je smješten upravljački kod za pisanje i čitanje na SPI sučelje i datoteka koja sadrži stablo odlučivanja u C formatu kao datoteka zaglavlja.

Upravljački kod za senzor sastoji se od konstanti koje definiraju oznake za registre i vrijednosti koje će se koristiti u aplikacijskom dijelu koda, enumeraciju za stanje poklopca, i funkcija lsm6dsox_spi_read, lsm6dsox_spi_write, lsm6dsox_load_ucf koje koriste generirani kod za interakciju s hardverom za pisanje i čitanje na SPI sučelje.

Funkcija lsm6dsox_spi_read služi za čitanje vrijednosti s odabranog registra i kao parametre prima registar koji se želi pročitati te pokazivač na spremnik u koji će se spremiti pročitana vrijednost. Da bi se izvršilo čitanje registra potrebno je prvo spustiti pin *CS - Chip Select* senzora kako bi se on odabrao, a u vrijednosti registra koja je proslijeđena funkciji potrebno je postaviti MSB na 1 kako bi se odabrala opcija čitanja. [58] Ove operacije izvode

se koristeći generirane funkcije HAL_GPIO_WritePin i HAL_SPI_TransmitReceive. Izvorni kod ove funkcije je sljedeći:

```
HAL_StatusTypeDef lsm6dsox_spi_read(uint8_t reg, uint8_t *data)
{
    uint8_t tx_buffer[2];
    uint8_t rx_buffer[2];

    // Set MSB pin to 1 for read
    tx_buffer[0] = reg | 0x80;
    tx_buffer[1] = 0x00;

    HAL_GPIO_WritePin(CS_LSM6DSOX_GPIO_Port, CS_LSM6DSOX_Pin,
GPIO_PIN_RESET);

    HAL_StatusTypeDef status = HAL_SPI_TransmitReceive(&hspi1, tx_buffer,
rx_buffer, 2, HAL_MAX_DELAY);

    HAL_GPIO_WritePin(CS_LSM6DSOX_GPIO_Port, CS_LSM6DSOX_Pin,
GPIO_PIN_SET);

    *data = rx_buffer[1];

    return status;
}
```

Funkcija lsm6dsox_spi_write služi za upisivanje vrijednosti u odabrani registar i kao parametre prima registar u koji se želi pisati te vrijednost koja će se upisati. Da bi se izvršilo pisanje registra potrebno je prvo spustiti pin *CS - Chip Select* senzora kako bi se on odabrao, a u vrijednosti registra koja je proslijeđena funkciji potrebno je postaviti MSB na 0 kako bi se odabrala opcija pisanja. [58] Ove operacije izvode se koristeći generirane funkcije HAL_GPIO_WritePin i HAL_SPI_Transmit. Izvorni kod ove funkcije je sljedeći:

```
HAL_StatusTypeDef lsm6dsox_spi_write(uint8_t reg, uint8_t data)
{
    uint8_t tx_buffer[2];

    // Set MSB pin to 0 for write
    tx_buffer[0] = reg & 0x7F;
    tx_buffer[1] = data;
```

```
    HAL_GPIO_WritePin(CS_LSM6DSOX_GPIO_Port, CS_LSM6DSOX_Pin,
GPIO_PIN_RESET);


    HAL_StatusTypeDef status = HAL_SPI_Transmit(&hspi1, tx_buffer, 2,
HAL_MAX_DELAY);


    HAL_GPIO_WritePin(CS_LSM6DSOX_GPIO_Port, CS_LSM6DSOX_Pin,
GPIO_PIN_SET);


    return status;
}
```

Zadnja funkcija u kodu je lsm6dsox_load_ucf koja služi za slanje generiranog stabla odlučivanja koje je u obliku C polja na senzor koristeći ranije spomenute funkcije za pisanje na SPI sučelje. Ova funkcija iterira kroz polje koje sadrži parove registara i njihovih vrijednosti te ih zapisuje na SPI sučelje kako bi konfigurirala jezgra za strojno učenje senzora. Kod ove funkcije je sljedeći:

```
void lsm6dsox_load_ucf()
{
    size_t ucf_size = sizeof(movement) / sizeof(ucf_line_t);
    size_t i;
    for (i = 0; i < ucf_size; i++)
    {
        uint8_t reg = movement[i].address;
        uint8_t value = movement[i].data;


        lsm6dsox_spi_write(reg, value);
    }


    HAL_Delay(10);
}
```

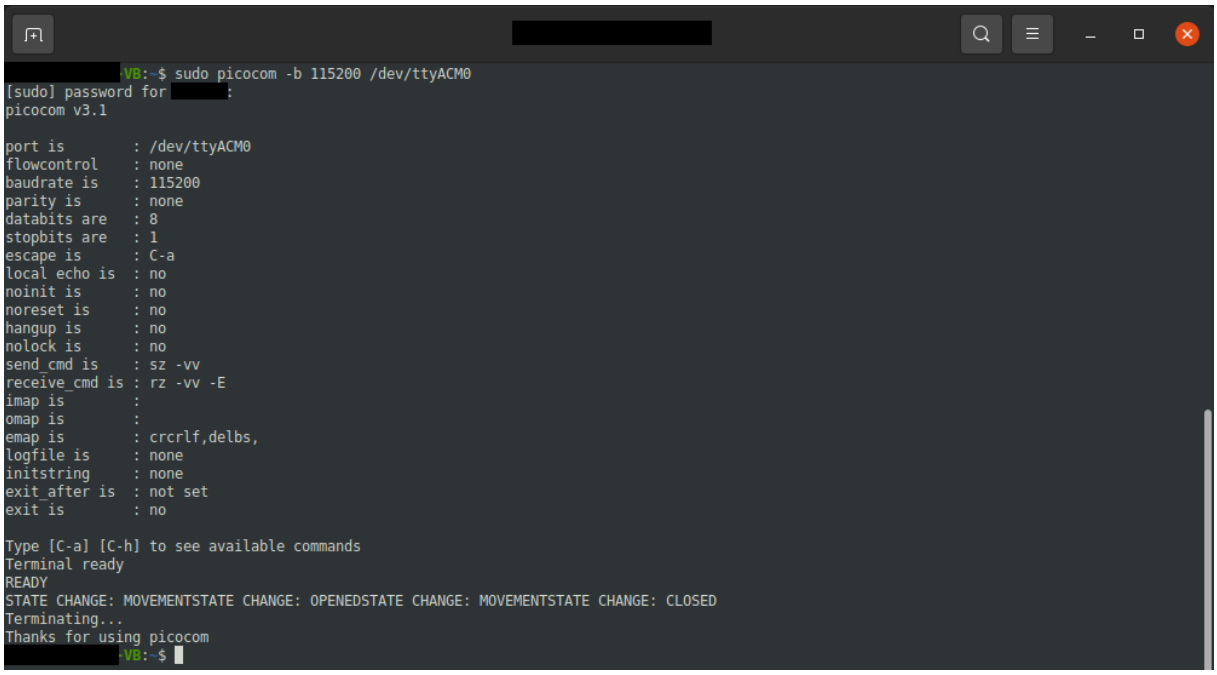
Nakon što su definirane funkcije za rad sa senzorom slijedi opis glavne funkcije koja se izvodi pri pokretanju aplikacije. Funkcija na početku inicijalizira USB sučelje putem generirane funkcije MX_USB_DEVICE_Init kako bi se podaci mogli slati sa senzora. Zatim se aplikacija nakratko pauzira koristeći funkciju osDelay kako bi se dala mogućnost korisniku da pripremi grafičku aplikaciju za prikaz stanja senzora koja će biti opisana u sljedećem potpoglavlju. Nakon pauze pali se zelena lampica na pločici kompleta za razvoj kao vizualni indikator da je aplikacija pokrenuta. Najprije se poziva ranije definirana funkcija lsm6dsox_load_ucf kako bi se jezgra za strojno učenje konfigurirala i pripremila za detekciju stanja. Nakon što je jezgra spremna ulazi se u beskonačnu petlju koja s pomoću SPI sučelja čita registar u koji senzor sprema rezultate stabla odlučivanja i ako se stanje promijenilo šalje se poruka s novim stanjem

na USB sučelje uređaja koju zatim može primiti vanjska aplikacija. Da bi se mogao pročitati registar s rezultatima stabla odlučivanja potrebno je prvo uključiti pristup tako da se u registru FUNC_CFG_ACCESS postavi MSB koji označava uključivanje pristupa ugrađenim funkcijama i registrima. Kod glavne funkcije je sljedeći:

```
void StartDefaultTask(void *argument)
{
  MX_USB_DEVICE_Init();

  // Prepare device
  osDelay(10000);
  HAL_GPIO_WritePin(USER_LED_GPIO_Port, USER_LED_Pin, GPIO_PIN_SET);
  usb_print("READY\r\n");

  lsm6dsox_load_ucf();

  enum State lastState = Closed;
  enum State mlc_output = Closed;

  /* Infinite loop */
  for (;;)
  {
    // Enable embedded functions to read MLC register
    lsm6dsox_spi_write(FUNC_CFG_ACCESS_REG, FUNC_CFG_ACCESS_SET);
    // Read output for decision tree 0
    lsm6dsox_spi_read(MLC0_SRC_REG, &mlc_output);
    // Disable embedded functions after read
    lsm6dsox_spi_write(FUNC_CFG_ACCESS_REG, FUNC_CFG_ACCESS_RESET);

    if (mlc_output != lastState)
    {
      switch (mlc_output)
      {
      case 0:
        usb_print("STATE CHANGE: CLOSED");
        break;
      case 4:
        usb_print("STATE CHANGE: MOVEMENT");
        break;
      case 8:
        usb_print("STATE CHANGE: OPENED");
```

```
        break;
      default:
        usb_print("STATE CHANGE: UNKNOWN");
        break;
      }
      lastState = mlc_output;
    }
  }
  osDelay(10);
}
```

Prije upotrebe grafičke aplikacije za prikaz stanja senzora moguće je testirati rad aplikacije tako da se koristeći neku aplikacije za čitanje serijskih sučelja otvori veza prema sučelju na koje je spojen uređaj. Za testiranje korištena je aplikacija picocom koja standardno dolazi na Ubuntu Linux distribuciji dok je uređaj na sučelju /dev/ttyACM0.

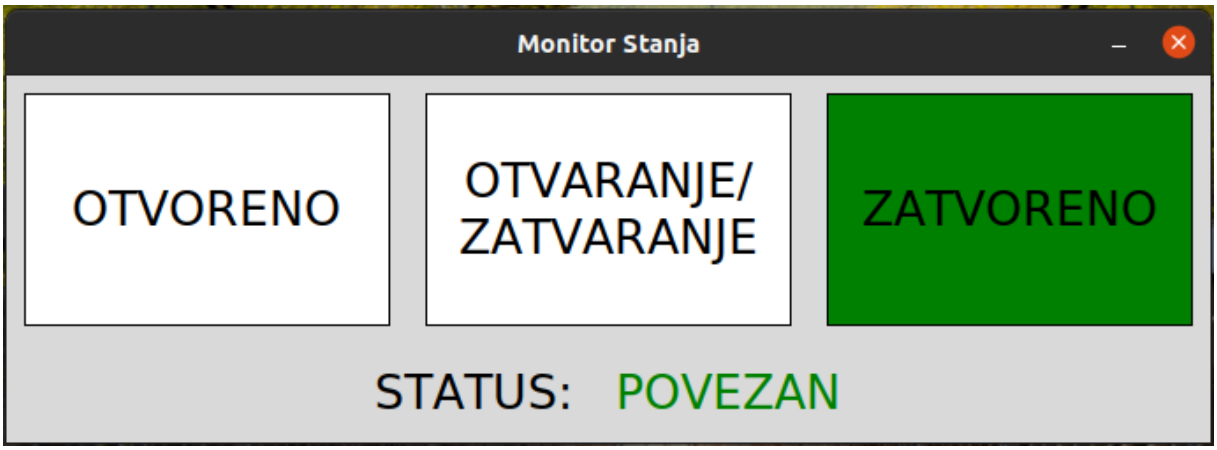


Slika 34. Testiranje aplikacije u terminalu (Izvor: autorski rad)

6.3. Aplikacija za prikaz stanja senzora

Aplikacija za prikaz stanja izrađena je koristeći Python programski jezik uz korištenje modula Tkinter za izradu grafičkog sučelja i modula serial za povezivanje i čitanje serijskog sučelja na koje je spojen uređaj. Aplikacija kontinuirano čita poruke sa serijskog sučelja /dev/ttyACM0 i ažurira korisničko sučelje na temelju poruka koje šalje senzor. Poruke koje šalje senzor su u formatu „STATE CHANGE: <novo stanje>“ gdje novo stanje može biti jedna

od vrijednosti CLOSED, OPENED, MOVEMENT, ili UNKNOWN. Na grafičkom sučelju prikazana su tri prozora od kojih svaki označava jedno moguće stanje. Kad se detektira stanje povezano s jednim od okvira on poprima zelenu boju kako bi se označilo da je detektirano odgovarajuće stanje. Ako se primi nepoznato stanje, svi prozori označavaju se žutom bojom. Statusna oznaka na dnu pokazuje je li uređaj spojen ili isključen. Zasebna dretva upravlja čitanjem serijskog sučelja u beskonačnoj petlji.



Slika 35. Grafičko sučelje aplikacije za prikaz stanja (Izvor: autorski rad)

7. Zaključak

Internet stvari jedna je od najtransformativnijih tehnologija 21. stoljeća te je pojam s kojim se svakodnevno susrećemo. Gotovo svi svakodnevni predmeti danas se međusobno mogu povezati i dijeliti podatke bilo da se radi kućanskim uređajima, automobilima, strojevima ili običnim prekidačima. Korištenjem jeftinih i pristupačnih uređaja, usluga u oblaku, podataka, analitike i mobilne tehnologije, fizički objekti sada mogu prikupljati i dijeliti podatke uz minimalnu ljudsku intervenciju. Takav spoj fizičkog i digitalnog svijeta omogućuje im da djeluju samostalno kako bi regulirali neki sustav ili obavili rad. IoT pruža ljudima jednostavniji život i rad omogućujući pametnija i učinkovitija rješenja. Danas se koristi u svim granama industrije gdje pomaže poduzećima bolje upravljanje poslovnim procesima, pruža im pristup većem broju podataka i informacija na temelju kojih mogu donositi bolje poslovne odluke. Omogućuje jednostavan pristup informacijama bilo kada, bilo gdje i na bilo kojem uređaju u realnom vremenu. Zbog povećanja broja povezanih uređaja bitno je razmotriti i potencijalne rizike kao što su sigurnosni rizici, zakonske regulative, upravljanje velikom količinom uređaja, ranjivosti sustava i nepostojanja univerzalnog standarda za interoperabilnosti.

Strojno učenje i umjetna inteligencija također je tehnologija koja je sve više prisutna u svakodnevnom životu. Strojno učenje široko je primjenjivo u mnogim industrijama i nalazi primjenu u različitim područjima kao što su prepoznavanje slike i govora, obrada prirodnog jezika, sustavi preporuka, otkrivanje prijevara, optimizacija portfelja, glasovni asistenti, autonomna vozila i automatizacija zadataka. Sposobnost učenja iz podataka i poboljšanja tijekom vremena čini strojno učenje nevjerojatno moćnim alatom za rješavanje kompleksnih poslovnih problema. Zbog velike količine podataka koju generira IoT spoj ove dvije tehnologija pruža veliku priliku i mogućnosti rješavanja problema na dosad neviđene načine. Omogućuje uređajima da analiziraju podatke u stvarnom vremenu, prepoznaju obrasce i donose autonomne odluke bez potrebe za stalnim nadzorom čovjeka. Time se postiže optimizacija procesa, smanjuju troškovi i povećava efikasnost u različitim industrijama. Dodatno, s razvojem tehnologija umjetne inteligencije na rubu sve više analitike može se odvijati direktno na IoT uređajima, čime se smanjuje potreba za slanjem podataka u oblak i omogućuje brža i sigurnija obrada informacija. Ovaj spoj umjetne inteligencije i Interneta stvari postaje ključan za razvoj pametnih sustava koji su samostalniji, prilagodljiviji i inteligentniji nego ikada prije.

U ovom radu uspješno je prikazana mogućnost rješavanja stvarnog poslovnog problema na takav način upotrebom senzora s ugrađenim strojnim učenjem. Uspješno je prikazano kako se takav senzor može koristiti za detekciju promjene stanja poklopca na cisterni za gorivo kako bi se zamijenio postojeći uređaj. Na ovaj način bitno se pojednostavljuje kompleksnost potrebnog hardvera uređaja, proces instalacije i kalibracije za detekciju. Ovaj

primjer daje samo uvid u funkcionalnosti senzora čiji je puni potencijal daleko veći. Integracijom dodatnih senzora u model, primjenom u stvarnim uvjetima i kreiranjem više modela stabla odlučivanja uređaj s ovakvim senzorom mogao bi se koristiti za detekciju bilo koje druge aktivnosti u raznim industrijama npr. praćenje gibanja vozila, praćenje brodova, detekcija vibracija/nagiba u industrijskim postrojenjima ili bilo koje druge aktivnosti koja se može detektirati s pomoću dostupnih senzora. Ovakva rješenja ne samo da poboljšavaju efikasnost praćenja i nadzora, već i omogućuju smanjenje troškova održavanja i povećanje sigurnosti u industrijskim i logističkim sustavima.

Popis literature

[1] K. Yasar i A. S. Gillis, »Internet of things (IoT),« [Mrežno]. Available: https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT. [Pokušaj pristupa 18 Studeni 2024].

[2] »What is IoT?,« Oracle, [Mrežno]. Available: https://www.oracle.com/in/internet-of-things/. [Pokušaj pristupa 18 Studeni 2024].

[3] P. Kostek, »mobile-magazine.com,« 10 Lipanj 2023. [Mrežno]. Available: https://mobile-magazine.com/articles/retelling-iots-space-history-from-apollo-missions-to-da. [Pokušaj pristupa 19 Studeni 2024].

[4] J. Lomberg, »The Apollo Guidance Computer – History's Unsung Hero,« 24 Srpanj 2019. [Mrežno]. Available: https://www.powersystemsdesign.com/articles/the-apollo-guidance-computer-historys-unsung-hero/135/14863. [Pokušaj pristupa 19 Studeni 2024].

[5] »Internet of things,« Wikipedia, [Mrežno]. Available: https://en.wikipedia.org/wiki/Internet_of_things. [Pokušaj pristupa 19 Studeni 2024].

[6] »The History of IoT: How This Technology Is Evolving,« 9 Rujan 2024. [Mrežno]. Available: https://www.cogniteq.com/blog/history-iot-how-technology-evolving. [Pokušaj pristupa 19 Studeni 2024].

[7] »A history of the Internet of Things,« [Mrežno]. Available: https://power.h5mag.com/future_power_technology_feb24/timeline-internet-of-things. [Pokušaj pristupa 19 Studeni 2024].

[8] A. Braun, »History of IoT: A Timeline of Development,« 25 Siječanj 2019. [Mrežno]. Available: https://www.iottechtrends.com/history-of-iot/. [Pokušaj pristupa 19 Studeni 2024].

[9] M. K. Pratt, »Top 12 most commonly used IoT protocols and standards,« TechTarget, 12 Srpanj 2023. [Mrežno]. Available: https://www.techtarget.com/iotagenda/tip/Top-12-most-commonly-used-IoT-protocols-and-standards. [Pokušaj pristupa 23 Studeni 2024].

[10] »Zigbee,« Wikipedia, [Mrežno]. Available: https://en.wikipedia.org/wiki/Zigbee. [Pokušaj pristupa 23 Studeni 2024].

[11] »Zigbee mesh network tutorial | Zigbee mesh network basics,« [Mrežno]. Available: https://www.rfwireless-world.com/Tutorials/Zigbee-mesh-network-tutorial.html. [Pokušaj pristupa 23 Studeni 2024].

[12] »Near-field communication,« Wikipedia, [Mrežno]. Available: https://en.wikipedia.org/wiki/Near-field_communication. [Pokušaj pristupa 23 Studeni 2024].

[13] »Internet of Things - Wi-Fi® is an essential IoT enabler,« Wi-Fi Alliance, [Mrežno]. Available: https://www.wi-fi.org/discover-wi-fi/internet-things. [Pokušaj pristupa 23 Studeni 2024].

[14] »5G,« Wikipedia, [Mrežno]. Available: https://en.wikipedia.org/wiki/5G. [Pokušaj pristupa 23 Studeni 2024].

[15] »Bluetooth Low Energy,« Wikipedia, [Mrežno]. Available: https://en.wikipedia.org/wiki/Bluetooth_Low_Energy. [Pokušaj pristupa 24 Studeni 2024].

[16] »LoRa,« Wikipedia, [Mrežno]. Available: https://en.wikipedia.org/wiki/LoRa. [Pokušaj pristupa 25 Studeni 2024].

[17] »Narrowband IoT: Your Definitive Guide to NB-IoT,« 18 Rujan 2024. [Mrežno]. Available: https://www.ezurio.com/resources/blog/narrowband-iot-your-definitive-guide-to-nb-iot. [Pokušaj pristupa 25 Studeni 2024].

[18] »What is an IoT Platform?,« [Mrežno]. Available: https://www.softwareag.com/en_corporate/resources/iot/article/iot-platform.html. [Pokušaj pristupa 25 Studeni 2024].

[19] »What is Azure Internet of Things (IoT)?,« Microsoft, 19 Studeni 2024. [Mrežno]. Available: https://learn.microsoft.com/en-us/azure/iot/iot-introduction. [Pokušaj pristupa 25 Studeni 2024].

[20] EMQ Technologies Inc., »Understanding AWS IoT Core: Features, Use Cases, & Quick Tutorial,« 25 Studeni 2024. [Mrežno]. Available: https://www.iotforall.com/understanding-aws-iot-core-features-use-cases-quick-tutorial. [Pokušaj pristupa 25 Studeni 2024].

[21] »How AWS IoT works,« Amazon, [Mrežno]. Available: https://docs.aws.amazon.com/iot/latest/developerguide/aws-iot-how-it-works.html. [Pokušaj pristupa 25 Studeni 2024].

[22] »Node-RED,« Wikipedia, [Mrežno]. Available: https://en.wikipedia.org/wiki/Node-RED. [Pokušaj pristupa 25 Studeni 2024].

[23] »Node-RED About,« [Mrežno]. Available: https://nodered.org/about/. [Pokušaj pristupa 25 Studeni 2024].

[24] »Home Assistant,« Wikipedia, [Mrežno]. Available: https://en.wikipedia.org/wiki/Home_Assistant. [Pokušaj pristupa 25 Studeni 2024].

[25] »Home Assistant Documentation,« [Mrežno]. Available: https://www.home-assistant.io/docs/. [Pokušaj pristupa 25 Studeni 2024].

[26] »Microcontroller,« Wikipedia, [Mrežno]. Available: https://en.wikipedia.org/wiki/Microcontroller. [Pokušaj pristupa 26 Studeni 2024].

[27] »STM32,« Wikipedia, [Mrežno]. Available: https://en.wikipedia.org/wiki/STM32. [Pokušaj pristupa 28 Studeni 2024].

[28] »STM32 High Performance MCUs,« [Mrežno]. Available: https://www.st.com/en/microcontrollers-microprocessors/stm32-high-performance-mcus.html. [Pokušaj pristupa 28 Studeni 2024].

[29] »STM32 Mainstream MCUs,« [Mrežno]. Available: https://www.st.com/en/microcontrollers-microprocessors/stm32-mainstream-mcus.html. [Pokušaj pristupa 28 Studeni 2024].

[30] »STM32 Ultra Low Power MCUs,« [Mrežno]. Available: https://www.st.com/en/microcontrollers-microprocessors/stm32-ultra-low-power-mcus.html. [Pokušaj pristupa 28 Studeni 2024].

[31] »STM32 Wireless MCUs,« [Mrežno]. Available: https://www.st.com/en/microcontrollers-microprocessors/stm32-wireless-mcus.html. [Pokušaj pristupa 29 Studeni 2024].

[32] »Software development tools,« [Mrežno]. Available: https://www.st.com/en/development-tools/software-development-tools.html. [Pokušaj pristupa 29 Studeni 2024].

[33] »ESP32,« [Mrežno]. Available: https://en.wikipedia.org/wiki/ESP32. [Pokušaj pristupa 1 Prosinac 2024].

[34] D. Wilcher, »A Guide for the ESP32 Microcontroller Series,« DigiKey, 29 Svibanj 2024. [Mrežno]. Available: https://www.digikey.com/en/maker/blogs/2024/a-guide-for-the-esp32-microcontroller-series. [Pokušaj pristupa 1 Prosinac 2024].

[35] »What is Machine Learning?,« 26 Svibanj 2024. [Mrežno]. Available: https://www.geeksforgeeks.org/ml-machine-learning/#what-is-machine-learning. [Pokušaj pristupa 9 Prosinac 2024].

[36] L. T. Lev Craig, »What is machine learning? Guide, definition and examples,« [Mrežno]. Available: https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML. [Pokušaj pristupa 9 Prosinac 2024].

[37] »Introduction to Deep Learning,« 26 Svibanj 2024. [Mrežno]. Available: https://www.geeksforgeeks.org/introduction-deep-learning/. [Pokušaj pristupa 9 Prosinac 2024].

[38] M. Crabtree, »What is Machine Learning? Definition, Types, Tools & More,« 8 Studeni 2024. [Mrežno]. Available: https://www.datacamp.com/blog/what-is-machine-learning. [Pokušaj pristupa 9 Prosinac 2024].

[39] G. Firican, »The history of Machine Learning,« [Mrežno]. Available: https://www.lightsondata.com/the-history-of-machine-learning. [Pokušaj pristupa 15 Prosinac 2024].

[40] »Machine learning,« [Mrežno]. Available: https://en.wikipedia.org/wiki/Machine_learning. [Pokušaj pristupa 15 Prosinac 2024].

[41] S. Tavasoli, »10 Types of Machine Learning Algorithms and Models,« 28 Listopad 2024. [Mrežno]. Available: https://www.simplilearn.com/10-algorithms-machine-learning-engineers-need-to-know-article. [Pokušaj pristupa 16 Prosinac 2024].

[42] »Linear Regression in Machine learning,« 16 Prosinac 2024. [Mrežno]. Available: https://www.geeksforgeeks.org/ml-linear-regression/. [Pokušaj pristupa 15 Prosinac 2024].

[43] »Logistic Regression in Machine Learning,« 20 Lipanj 2024. [Mrežno]. Available: https://www.geeksforgeeks.org/understanding-logistic-regression/. [Pokušaj pristupa 16 Prosinac 2024].

[44] S. Piduguralla, »Understanding the Sigmoid Function in Logistic Regression: Mapping Inputs to Probabilities,« 8 Rujan 2023. [Mrežno]. Available: https://www.linkedin.com/pulse/understanding-sigmoid-function-logistic-regression-piduguralla/. [Pokušaj pristupa 16 Prosinac 2024].

[45] »Decision Tree in Machine Learning,« 15 Ožujak 2024. [Mrežno]. Available: https://www.geeksforgeeks.org/decision-tree-introduction-example/. [Pokušaj pristupa 15 Prosinac 2024].

[46] N. Gahlawat, »Decision Trees: A Powerful Tool in Machine Learning,« 24 Lipanj 2023. [Mrežno]. Available: https://medium.com/@nidhigahlawat/decision-trees-a-powerful-tool-in-machine-learning-dd0724dad4b6. [Pokušaj pristupa 15 Prosinac 2024].

[47] »What is a Neural Network?,« 6 Prosinac 2024. [Mrežno]. Available: https://www.geeksforgeeks.org/neural-networks-a-beginners-guide/. [Pokušaj pristupa 15 Prosinac 2024].

[48] G. Paaß, »Deep Learning: How do deep neural networks work?,« The Lamarr Institute for Machine Learning and Artificial Intelligence, 21 Travanj 2021. [Mrežno]. Available:

https://lamarr-institute.org/blog/deep-neural-networks/. [Pokušaj pristupa 15 Prosinac 2024].

[49] »K-Nearest Neighbor(KNN) Algorithm,« 15 Srpanj 2024. [Mrežno]. Available: https://www.geeksforgeeks.org/k-nearest-neighbours/. [Pokušaj pristupa 16 Prosinac 2024].

[50] S. Zidi, A. Mihoub, S. Mian Qaisar i M. Krichen, »Theft Detection Dataset for Benchmarking and Machine Learning based Classification in a Smart Grid Environment,« Lipanj 2022. [Mrežno]. Available: https://www.researchgate.net/publication/360554845_Theft_Detection_Dataset_for_Benchmarking_and_Machine_Learning_based_Classification_in_a_Smart_Grid_Environment. [Pokušaj pristupa 16 Prosinac 2024].

[51] »K means Clustering – Introduction,« 29 Kolovoz 2024. [Mrežno]. Available: https://www.geeksforgeeks.org/k-means-clustering-introduction/. [Pokušaj pristupa 16 Prosinac 2024].

[52] N. Arya, »K-Means Clustering for Unsupervised Machine Learning,« 10 Studeni 2023. [Mrežno]. Available: https://www.ejable.com/tech-corner/ai-machine-learning-and-deep-learning/k-means-clustering/. [Pokušaj pristupa 16 Prosinac 2024].

[53] »Principal Component Analysis(PCA),« 10 Rujan 2024. [Mrežno]. Available: https://www.geeksforgeeks.org/principal-component-analysis-pca/. [Pokušaj pristupa 16 Prosinac 2024].

[54] M. Nashed, »Principal Component Analysis (PCA) Transformation,« [Mrežno]. Available: https://www.biorender.com/template/principal-component-analysis-pca-transformation. [Pokušaj pristupa 16 Prosinac 2024].

[55] K. Kanižaj, »MSFTv2 Projektni zadatak,« SICK Mobilisis d.o.o., Varaždin, 2023.

[56] STMicroelectronics, »STEVAL-MKSBOX1V1,« 9 Travanj 2021. [Mrežno]. Available: https://www.st.com/en/evaluation-tools/steval-mksbox1v1.html. [Pokušaj pristupa 7 Siječanj 2025].

[57] »STEVAL-MKSBOX1V1 Wireless Multi Sensor User Manual,« 2 Prosinac 2021. [Mrežno]. Available: https://manuals.plus/_st/steval-mksbox1v1-wireless-multi-sensor-manual. [Pokušaj pristupa 7 Siječanj 2025].

[58] STMicroelectronics, »LSM6DSOX,« 4 Lipanj 2024. [Mrežno]. Available: https://www.st.com/en/mems-and-sensors/lsm6dsox.html#documentation. [Pokušaj pristupa 10 Siječanj 2025].

[59] STMicroelectronics, »GUI for X-CUBE-MEMS1, motion MEMS and environmental sensor software expansion for STM32Cube,« STMicroelectronics, [Mrežno]. Available: https://www.st.com/en/development-tools/unicleo-gui.html. [Pokušaj pristupa 25 Siječanj 2025].

[60] STMicroelectronics, »MEMS evaluation kit software package for Linux, Mac OSX and Windows,« STMicroelectronics, [Mrežno]. Available: https://www.st.com/en/development-tools/unico-gui.html. [Pokušaj pristupa 25 Siječanj 2025].

[61] STMicroelectronics, »STM32Cube initialization code generator,« STMicroelectronics, [Mrežno]. Available: https://www.st.com/en/development-tools/stm32cubemx.html. [Pokušaj pristupa 28 Siječanj 2025].

Popis slika