SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1641

ODREĐIVANJE BAKTERIJSKIH GENA REZISTENTNIH NA ANTIBIOTIKE

Marko Žagar

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1641

ODREĐIVANJE BAKTERIJSKIH GENA REZISTENTNIH NA ANTIBIOTIKE

Marko Žagar

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

ZAVRŠNI ZADATAK br. 1641

Pristupnik:	**Marko Žagar (0036542479)**
Studij:	Elektrotehnika i informacijska tehnologija i Računarstvo
Modul:	Računarstvo
Mentorica:	izv. prof. dr. sc. Mirjana Domazet-Lošo

Zadatak: **Određivanje bakterijskih gena rezistentnih na antibiotike**

Opis zadatka:

U okviru ovoga završnog rada potrebno je proučiti problem bakterijske rezistencije na antibiotike te metode koje se koriste za određivanje rezistentnih gena: metode koje se temelje na usporedbi sljedova i metode strojnog učenja. Odabrane metode za određivanje rezistentnih gena implementirati korištenjem programskog jezika Python te testirati na javno dostupnom skupu podataka. Dobivene rezultate potrebno je usporediti s postojećim rješenjima.

Rok za predaju rada: 14. lipnja 2024.

Ovim se putem zahvaljujem svima koji su pridonijeli uspješnoj realizaciji završnog rada, ali i svima koji su me podržali tijekom prijediplomskog studija na Fakultetu elektrotehnike i računarstva.

Posebno se zahvaljujem mentorici na njezinom strpljenju, stručnim savjetima i konstantnoj podršci bez koje ovaj rad ne bi bio moguć.

Sadržaj

1. Uvod

Posljednjih nekoliko godina svjedočimo napretku i sve široj upotrebi umjetne inteligenciju u svim segmentima života [1]. Koristi se za napredno i precizno prevođenje jezika, optimiranje ruta pri navigaciji, brzo i sigurno prepoznavanje lica, ispravljanje i doradu tekstova, napredna pretraživanja, komunikaciju s ljudima i druge razne primjene.

Osim umjetne inteligencije, došlo je do napretka i u područjima poput sekvenciranja gena. Sekvenciranje nove generacije [4] je omogućilo brzo, precizno i jeftino određivanje nukleinskih baza u genetskom kodu. Zbog novih tehnologija sekvenciranja, pojavio se veliki skup podataka koje je moguće iskoristiti za istraživanje bioloških pojava poput rezistencije na antibiotike.

U ovom radu sam proučio što je to rezistencija na antibiotike i usporedio metode pristupa identifikaciji i klasifikaciji rezistencije. Nakon toga sam pobliže objasnio metodu identifikacije i klasifikacije pomoću umjetne inteligencije. Koristio sam javno dostupan jezični model za stvaranje reprezentacija gena te sam u programskom jeziku *Python* izgradio model strojnog učenja za identifikaciju i klasifikaciju rezistencije.

2. Opis problema

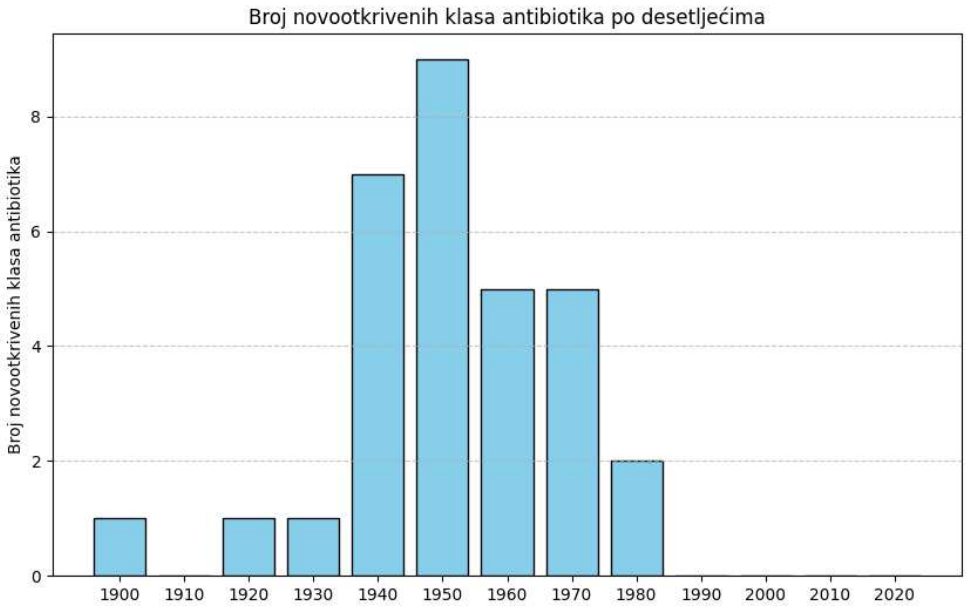
Jedan od većih problema u medicini i zdravstvu je rezistencija bakterija na antibiotike [2]. Rezistencija predstavlja veliki medicinski problem zato što onemogućuje primjenu određenih tretmana. Posljedica rezistencije su produžene hospitalizacije, veće cijene liječenja te veća smrtnost od bakterijskih infekcija. Osim u medicini, problem rezistencije je prisutan u drugim područjima poput prehrambene industrije te ekologije.

Problem rezistencije se u prošlosti nije isticao zato što su novootkriveni antibiotici nadomjestili lošije funkcioniranje starih antibiotika. No, otkrića novih antibiotika su sve rjeđa (Sl. 2.1), a udio rezistentnih bakterija je sve veći. Jako se malo tvrtki upušta u istraživanje novih antibiotika zbog velike cijene istraživanja i velikog rizika od neuspjeha plasiranja antibiotika na tržište. Zbog toga je broj novootkrivenih antibiotika jako malen [5]. Dodatno, ako novi antibiotik prođe rigorozno testiranje, neće biti financijski isplativ zato što će ga doktori prepisivati u minimalnim količinama na kratke periode liječenja kako se ne bi pojavila nova rezistencija.

Zbog toga je potrebno prikupiti informacije o postojanju i vrsti rezistencije te iskoristiti znanje koje imamo o tim rezistencijama te na temelju njega izabrati prikladan tretman te maksimizirati učinkovitost lijekova.

Rezistencija je uzrokovana određenim genima koji imaju sposobnost brze mutacije i širenja među bakterijama. Predviđanje rezistencije je vrlo zahtjevno zbog složenih i raznolikih mehanizama rezistencije te brze i nepredvidive evolucije bakterija. Klasični biološki pristupi identifikaciji i klasifikaciji rezistencije poput *Kirby-Bauer disk difuzijskog testa* [3] su spori, zahtjevni i skupi zbog čega ih nije moguće primijeniti u rješenjima koja zahtijevaju brzu identifikaciju i klasifikaciju rezistencije.

Sekvenciranje nove generacije [4] svojom cijenom i brzinom sekvenciranja gena rješava probleme klasičnog pristupa. Genetski slijed sekvenciran iz bakterija se može analizirati naprednim alatima s pomoću kojih je moguće saznati informacije o rezistenciji organizma na antibiotike.



Sl. 2.1: Broj novootkrivenih klasa antibiotika po desetljećima [5]

3. Rezistencija na antibiotike

Otkriće antibiotika je jedno od najvećih postignuća u medicini. Već prilikom dodjele Nobelove nagrade za otkriće antibiotika 1945. godine, Sir Alexander Fleming upozorio je na mogućnost razvoja bakterijske rezistencije na antibiotike ako se oni ne budu pravilno koristili. Bakterije postižu rezistenciju na različite načine te ju na različite načine šire među drugim bakterijama.

3.1. Mehanizmi rezistencije

Bakterije mogu stvoriti rezistenciju na više načina [6]:

1. ograničavanjem ulaska antibiotika u bakteriju (npr. selektivno propuštanje antibiotika kroz vanjsku membranu kod gram-negativnih bakterija),
2. izbacivanjem antibiotika iz bakterijske stanice s pomoću specijalno prilagođenih pumpi kroz staničnu stijenku (npr. pumpe bakterije *Pseudomonas aeruginosa* za izbacivanje raznih antibiotika),
3. modifikacijom antibiotika kako bi on izgubio svoju funkcionalnost (npr. stvaranje enzima karbapenemaza kod bakterije *Klebsiella pneumoniae*),
4. izmjenom receptora na koje se vežu antibiotici (npr. gen *mcr-1* kod bakterije *Escherichia coli* stvara komponentu na staničnoj stijenci kako se antibiotik kolistin ne bi mogao vezati),
5. stvaranjem novih staničnih procesa koji će nadomjestiti one onemogućene utjecajem antibiotika.

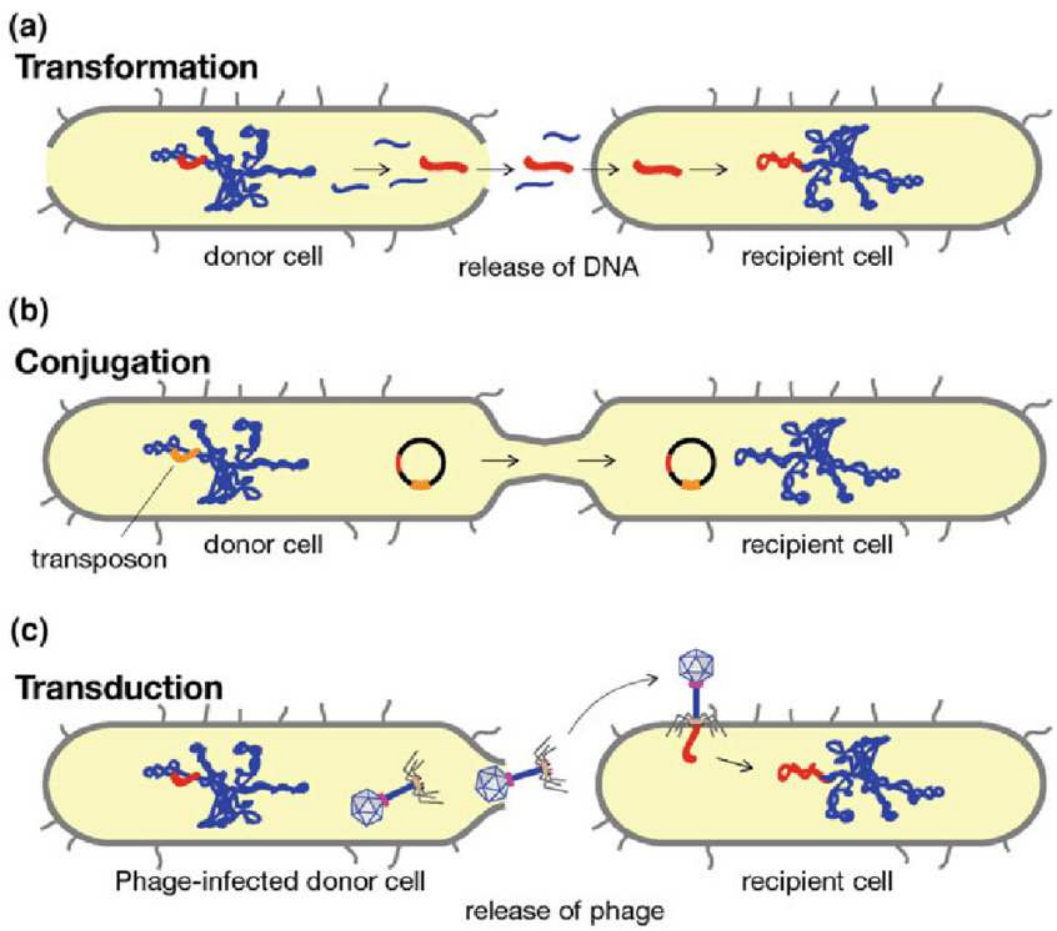
Poznavanje osnovnih mehanizama rezistencije ključno je za ovaj rad jer predstavljaju biološku osnovu modela strojnog učenja koji predviđa rezistenciju na antibiotike, treniranog na temelju sličnosti i razlika između gena.

3.2. Širenje rezistencije

Osim nasljeđivanja rezistencije s roditelja na potomke, bakterije imaju sposobnost horizontalnog prijenosa gena [7]. To znači da je moguća izmjena gena između odraslih jedinki bakterija koje nisu nužno iste vrste. Horizontalni prijenos se može postići na tri različita načina (Sl. 3.1):

- Prvi način je transdukcija u kojoj su bakteriofazi prijenosnici rezistentnih gena. Nakon što bakteriofag zarazi bakteriju i razmnoži se, u genomima potomka bakteriofaga može biti ubačen gen iz zaražene bakterije te ga mogu prilikom sljedeće zaraze ubaciti u genom druge bakterije. Zaražena bakterija će u tom slučaju biti otporna na određeni antibiotik.
- Drugi način prijenosa je konjugacija u kojoj se izmjena genetskog materijala dešava uslijed dodirivanja dvije bakterije. Mogu se prenijeti dvije vrste genetskog materijala: plazmidi (mali kružni DNA) i transpozoni (prepisani dijelovi bakterijskog kromosoma). Nakon izmjene gena, bakterija može stvoriti rezistenciju kao i kod transdukcije.
- Posljednji način je unošenje gena otpuštenog iz žive ili mrtve bakterije u okolinu. Taj način unošenja gena se zove transformacija.

Zbog čimbenika poput dinamičnog horizontalnog prijenosa gena i brze reprodukcije bakterija te sklonosti genetskim mutacijama, rezistencija se vrlo brzo širi i predstavlja veliki problem u medicini i srodnim područjima [2]. Zbog toga je potrebno pokušati spriječiti stvaranje rezistencije ispravnim korištenjem antibiotika, ali i prilagoditi korištenje antibiotika uzimajući u obzir vrste bakterijske rezistencije koje bakterija posjeduje. Model identifikacije i klasifikacije gena rezistentnih na antibiotike može pružiti informaciju o postojanju rezistencije i vrsti antibiotika na koji je bakterija rezistentna te tako pomoći u stvaranju individualiziranih tretmana.



Sl. 3.1: Mehanizmi horizontalnog prijenosa gena [7]

4. Podaci

4.1. Opis podataka

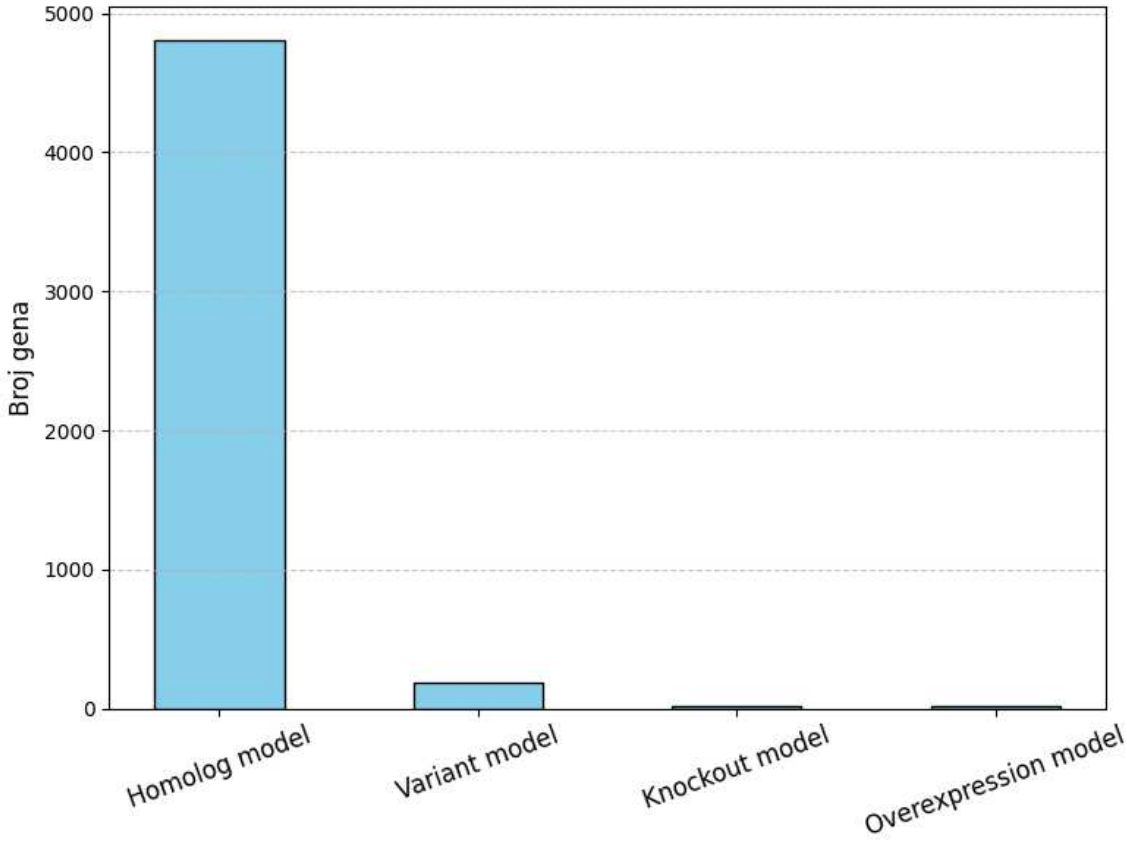
U svojem radu sam koristio podatke iz CARD (*Comprehensive Antibiotic Resistance Database*) baze podataka [8] koja objedinjuje 5,027 gena koji stvaraju rezistenciju na antibiotike (ARG, eng. *antibitoic resistance gene*) objavljenih u veljači 2024. godine. Podaci su podijeljeni na 4 skupa prema načinu na koji prisutnost i oblik gena utječu na rezistenciju:

1. *homolog model* čini većinu podataka i u njemu se nalaze geni koji svojom prisutnošću stvaraju rezistenciju,
2. *knockout model* u kojem se nalaze geni koji stvaraju rezistenciju u slučaju da ne postoje ili su inhibirani,
3. *variant model* u kojem se nalaze geni koji stvaraju rezistenciju ako se dogodi određena mutacija na njima,
4. *overexpression model* u kojem se nalaze geni koji stvaraju rezistenciju pojačanim prevođenjem u proteine.

Broj gena po modelu je prikazan na grafu Sl. 4.1. *Homolog model* sadrži većinu podataka što znači da se u većini slučajeva rezistencija postiže prisutnošću određenog gena.

Svaki skup podataka dolazi u dvije vrste zapisa. Prva vrsta je FASTA proteinski zapis, dok je druga vrsta FASTA nukleotidni zapis. Za identifikaciju i klasifikaciju rezistencije važna je funkcija proteina, ali ne i mutacije koje ne utječu na funkciju gena pa su u ovom radu podaci korišteni u FASTA proteinskom zapisu.

U FASTA proteinskom zapisu se za svaki gen, osim njegovog proteinskog slijeda nalaze i ostale informacije poput naziva proteina, vrste bakterije iz koje je sekvenciran te identifikator ontologije (ARO, eng. a*ntibiotic resistance ontology*). S pomoću tog identifikatora možemo saznati informacije poput obitelji ARG-a, klasu antibiotika na koji je rezistentan, mehanizam rezistencije i dr.



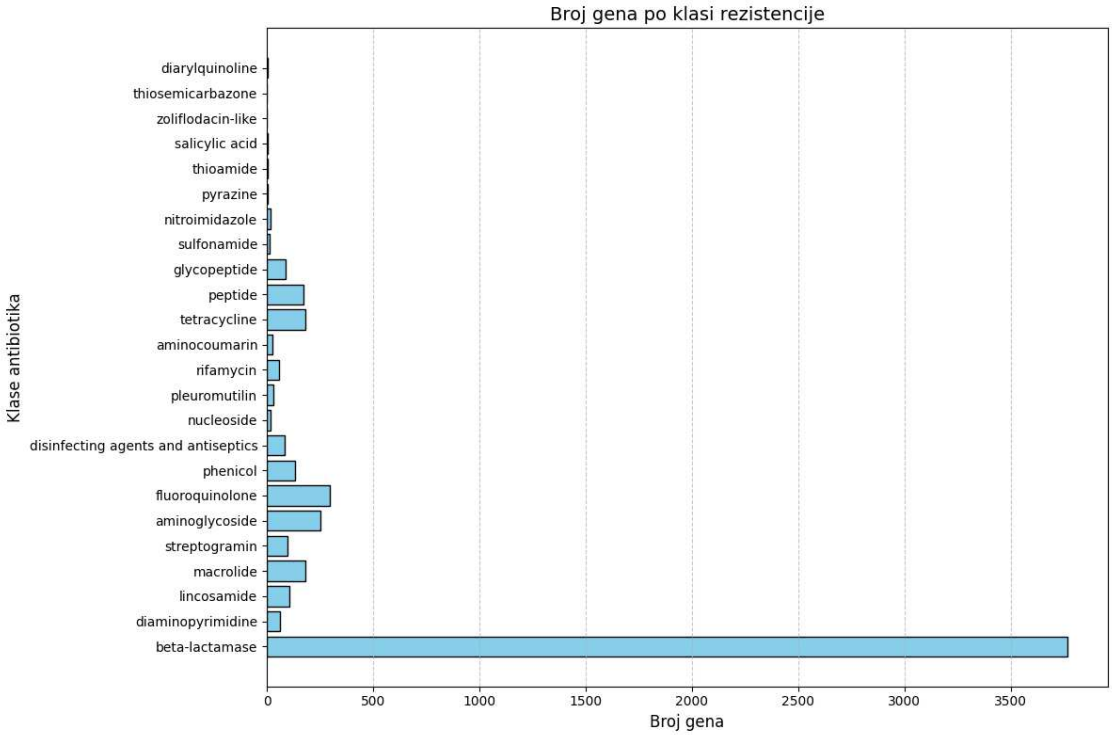
Sl. 4.1: Broj gena unutar svakog skupa podataka u CARD bazi podataka [8]

4.2. Priprema podataka

Za treniranje modela identifikacije ARG-a sam osnovni skup podataka proširio skupom gena koji ne stvaraju rezistenciju. Te podatke sam preuzeo s UniProt baze podataka [10] koristeći napredno pretraživanje s upitom: *„Taxonomy: Bacteria (eubacteria)* AND NOT *Keyword: Antibiotic resistance* AND *Reviewed: Yes”* Iz tog sam skupa zatim nasumično odabrao jednak broj gena koliki se nalazi u skupu rezistentnih gena (5,027).

Za treniranje modela klasifikacije ARG-a sam grupirao specifične vrste antibiotika u zajedničke klase antibiotika koje sam iščitao iz popisa klasa i potklasa antibiotika [12]. Klase koje nisam mogao grupirati, a imale su premalen broj podataka sam uklonio iz skupa. Nakon obrade podataka, ostalo je 4,876 gena.

Na grafu Sl. 4.2 je prikazan broj ARG-a po klasi antibiotika nakon što sam grupirao i obradio podatke. Najviše zapisa se odnosi na beta-laktamske antibiotike. Velika razlika u brojnosti ima smisla zato što su beta-laktamski antibiotici najčešće prepisivani antibiotici (udio na tržištu do 65 %) i imaju značajan broj podvrsta [9]. Za 5.8 % antibiotika je zabilježena rezistencija na više od jedne klase antibiotika. U grafu su ti geni brojani u svakoj klasi rezistencije.



Sl. 4.2: Broj gena po klasi antibiotika na koji je rezistentan iz CARD baze [8] nakon grupiranja po klasama antibiotika

4.3. Podaci u referentnom radu

Rad „*PLM-ARG: antibiotic resistance gene identification using a pretrained protein language model*” [13] (u daljnjem tekstu: „referentni rad”) ću koristiti za usporedbu rezultata identifikacije i klasifikacije. Podatke koje su koristili za klasifikaciju su objedinili iz 6 baza podataka (*CARD [8], ResFinder [33], MEGARes [31], AMRFinderPlus [32], ARGMiner [30], HMD-ARG-DB [34]*) koje objedinjuju 28,579 ARG-a. Za identifikaciju su podatke preuzeli isključivo s UniProt baze podataka koja objedinjuje 2,280 ARG-a te isto toliko nasumično odabranih bakterijskih gena koji ne stvaraju rezistenciju. Najnoviji podaci korišteni u referentnom radu su podaci iz CARD baze podataka objavljeni u svibnju 2022. godine.

U referentnom radu za klasifikaciju ARG-a koriste 4.4 puta veći skup podataka među kojima se nalazi skup iz CARD baze podataka koja je u ovom radu jedini izvor gena koji stvaraju rezistenciju. S druge strane podaci koje sam ja koristio za identifikaciju su 2.8 puta veći te se ne preklapaju s podacima korištenim u referentnom radu.

5. Metode

Postoje dvije metode s pomoću kojih možemo identificirati i klasificirati rezistencije na temelju proteinskih sljedova.

Prva metoda je uspoređivanje sljedova željenog gena s poznatim genima rezistencije te odlučivanje na temelju sličnosti sljedova [11]. Taj pristup se može ostvariti s pomoću bioinformatičkih alata kao što je BLAST [14]. BLAST je heuristički pristup poravnanju sljedova zbog čega ima veliku brzinu rada i sposobnost usporedbe velikog skupa podatka. Algoritam uspoređuje upitni slijed s bazom podataka sljedova tako da se upitni slijed dijeli u riječi zadane duljine prema kojima se određuje sličnost sa sljedovima iz baze podataka. Nakon toga se sekvence proširuju u oba smjera kako bi se pronašla najbolja poravnanja. Poravnati segmenti se zatim ocjenjuju na temelju matrice sličnosti te se određuje statistička značajnost.

Drugi pristup je korištenje jezičnog modela za transformiranje proteinskog slijeda u numeričke reprezentacije te modela nadziranog strojnog učenja za identifikaciju i klasifikaciju rezistencije pošto su klase rezistencije poznate. Prednost pristupa sa strojnim učenjem je što model ne promatra samo sličnost sljedova, već uzima u obzir redoslijed i kontekst određenih uzoraka u genu. Na primjer, malena mutacija na jednom mjestu unutar gena može biti puno značajnija od velike mutacije na nekom drugom mjestu unutar gena.

U ovom radu sam koristio ESM-1b [26] proteinski jezični model i strojno učenje kao glavnu metodu istraživanja. Jezični model sam iskoristio za stvaranje reprezentacijskih vektora, dok sam strojno učenje iskoristio za identifikaciju i klasifikaciju gena.

6. Strojno učenje

Strojno učenje [15] je grana umjetne inteligencije u kojoj treniramo model da nauči uzorke i pravilnosti iz zadanog skupa podataka te da na temelju njih donosi zaključke o novim i neviđenim uzorcima. Postoji nekoliko grana strojnog učenja:

- Nadzirano učenje je grana strojnog učenja kojem je cilj na temelju značajki predvidjeti ciljnu vrijednost. Ako je ciljna vrijednost kontinuirana radi se o regresiji, a ako je ciljna vrijednost diskretna, radi se o klasifikaciji. Identifikacija je slučaj klasifikacije između dvije klase.
- Nenadzirano učenje nema označene podatke za razliku od nadziranog učenja. Cilj modela nenadziranog učenja jest uočiti sličnosti između uzorka i grupirati ih u logične cjeline.
- Podržano učenje se temelji na interakciji s okruženjem i primanju povratnih informacija u obliku nagrada i kazni. Model se trenira tako da maksimizira kumulativnu nagradu kroz seriju akcija.
- Polunadzirano učenje je slično nadziranom učenju, osim što nisu poznate sve ciljne vrijednosti. Uzorci bez ciljnih vrijednosti služe za postizanje bolje uspješnosti modela.

U ovom radu je korišteno isključivo nadzirano učenje pošto su za sve podatke poznate ciljne vrijednosti.

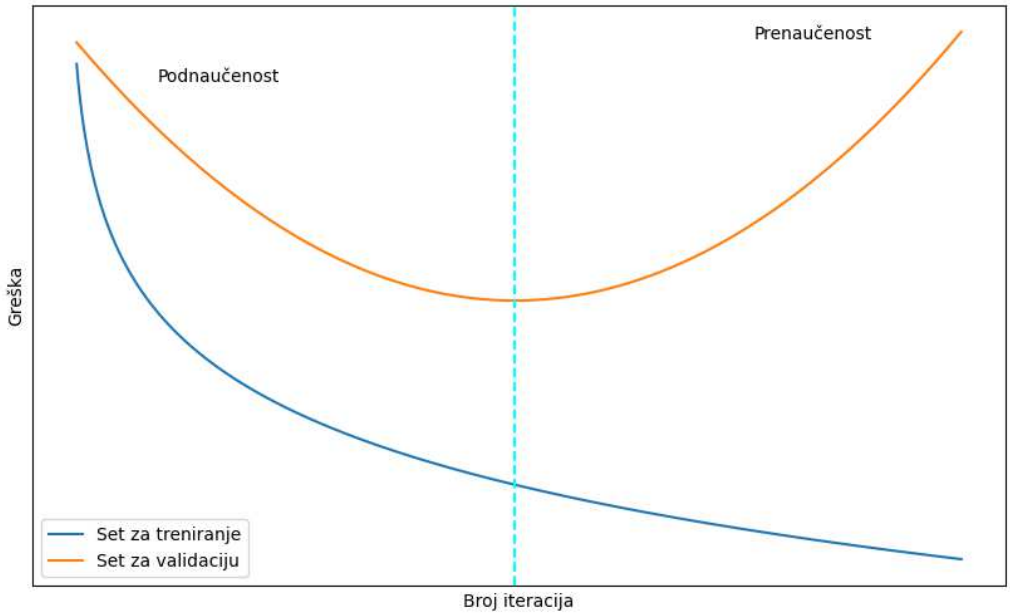
Izgradnja modela se odvija u nekoliko faza. Prva faza je podjela skupa podataka na skup za treniranje i testiranje (obično u omjeru 80 % naprema 20 % ili u omjeru 70 % naprema 30 %). Model se zatim trenira na skupu za treniranje gdje kroz iteracije, model podešava težine i prilagođava se podacima. Zatim slijedi testiranje u kojem se na ulaz modela stavljaju podaci iz skupa za testiranje. Bilježi se uspješnost svake predikcije te se na kraju ispisuje ukupna uspješnost modela.

Podjela skupa na skup za treniranje i skup za testiranje se još naziva i metoda izdvajanja. To je najjednostavnija metoda unakrsne validacije i ona se provodi kako bi se izbjegla prenaučenost modela [16]. Prenaučenost je problem kada se model previše prilagodi podacima te izgubi sposobnost generalizacije na nove, neviđene podatke. Navedena podjela je najjednostavniji način uočavanja prenaučenosti zato što se za testiranje koriste podaci koje model nikada nije vidio prilikom treniranja.

Na slici Sl. 6.1 je prikazan odnos pogreške i broja iteracija prilikom treniranja modela. Na označenoj graničnoj vrijednosti model je optimalno istreniran zato što je tu najmanja greška na skupu za testiranje i model najbolje generalizira. Ako se model nastavi trenirati, greška na skupu za testiranje će rasti, a model će postati prenaučen.

Prenaučenost se generalno može izbjeći na nekoliko načina:

1. *Rano zaustavljanje* je prekidanje treniranja u trenutku kada model postiže najbolje rezultate.
2. *Treniranje s više podataka* će spriječiti prenaučenost jer će model biti primoran generalizirati.
3. *Odabir značajki* koje najviše doprinose točnosti modela će smanjiti složenost modela, a time i vjerojatnost da se model prenauči.
4. *Unakrsna provjera* je metoda u kojoj se podaci dijele u više skupova te se kroz iteracije podaci treniraju i testiraju na različitim skupovima.
5. *Balansiranje podatka* se provodi kada postoji neravnoteža u količini podataka različitih klasa te se može umjetnim putem povećati jedan skup, odnosno smanjiti drugi.
6. *Regularizacija* je uvođenje dodatnih informacija koje sprječavaju da model postane presložen te time smanjuju prenaučenost.



Sl. 6.1: Odnos pogreške modela u odnosu na broj iteracija treniranja [17]

6.1. Stabla odluke

Stabla odluke [17] su neparametarska metoda strojnog učenja. Sastoje se od čvorova koji odgovaraju značajkama, grana koje odgovaraju vrijednostima značajke te listova koji odgovaraju klasifikacijskim odlukama. Primjer se stavlja na korijen stabla te se slijedno donose odluke na temelju čvorova i vrijednosti značajka primjera. Kada stablo dosegne list, primjer se klasificira oznakom lista.

Prilikom izgradnje stabla, u svakom trenutku nam je potreban kriterij za odabir najpovoljnije značajke za grananje. Želimo da svaka sljedeća značajka što bolje diskriminira između primjera. Kriteriji s pomoću kojih možemo postići željeno svojstvo su informacijska dobit, Gini index ili Hi-kvadrat test [17].

Prednost stabla odluke je jednostavnost izgradnje, interpretacije i vizualizacije. S druge strane, nedostatak stabla odluke je sklonost prenaučenosti. Prenaučenost se kod stabla odluke sprječava ograničavanjem rasta stabla prije postizanja savršene klasifikacije na skupu za učenje ili naknadnim podrezivanjem prenaučenog stabla. Postoje dvije vrste kriterija za određivanje optimalne dubine stabla. Intrinzični kriteriji su unaprijed zadana dubina, trenutak kada broj primjera u čvoru padne ispod zadane vrijednosti, trenutak kada je pad entropije manji od zadanog praga. Ekstrinzični kriterij je pad točnosti na skupu podataka za provjeru.

6.2. XGBoost

XGBoost (*Extreme Gradient Boosting*) [18] je ansambl model koji koristi stabla odluke kao osnovne modele te *gradient boosting* kao mehanizam optimizacije tih osnovnih modela.

Gradient boosting je optimizacijska metoda koja serijski gradi modele. Svaki sljedeći model pokušava optimizirati gubitak prethodnog koristeći metodu gradijentnog spusta. Svaki se model trenira na rezidualima prošlog modela te tako pokušava predvidjet gdje će prošli model zakazati. Konačni izlaz se izračunava skaliranim zbrajanjem predikcija svih stabala odlučivanja.

XGBoost ima ugrađene regularizacijske parametre za sprječavanje prenaučenosti koji uključuju podrezivanje stabla i penaliziranje previše složenih modela. Omogućuje paralelnu obradu podataka te učinkovito korištenje računalnih resursa.

7. Duboko učenje

Duboko učenje [20] je grana strojnog učenja koja koristi duboke neuronske mreže. Duboke neuronske mreže su neuronske mreže [21] s tri ili više sloja. Razlika između strojnog i dubokog učenja je prvenstveno u tome što model dubokog učenja može učiti na sirovim, neobrađenim podacima poput slike ili teksta, dok model strojnog učenja zahtjeva podatke formatirane u skup značajki. Nedostatak dubokog učenja je što zahtjeva veću količinu podataka te je treniranje modela zahtjevnije.

Model dubokog učenja stvara predikcije tako da sirove podatke provlači kroz nelinearne slojeve gdje svaki sloj transformira podatke prethodnog sloja te ih propagira u sljedeći, više apstraktan sloj. Taj proces se zove propagacija unaprijed. Prvi i posljednji sloj se nazivaju vidljivi slojevi, dok se ostali nazivaju skriveni slojevi.

Propagacija unatrag je algoritam u kojemu je cilj minimizirati grešku prilikom propagacije unaprijed tako da se svakom neuronu krenuvši od izlaznog sloja podese težine i pristranost. Postoje i drugi načini učenja modela poput genetskog algoritma [22].

Neke od arhitektura dubokog učenja su [23]:

- *Višeslojni perceptron* je arhitektura dubokog učenja gdje su ulazi u svaki neuron izlazi neurona prethodnog sloja pomnoženi s težinom te ulaz pristranosti. Na svaki sloj osim posljednjeg se primjenjuje aktivacijska funkcija.
- *Konvolucijska neuronska mreža* ima dva specifična sloja. Konvolucijski sloj služi za filtriranje bitnih informacija s ulaza te sloj sažimanja koji smanjuje dimenzionalnost podataka. Koristi se za obrade slika, videa i obradu prirodnog jezika.
- *Povratna neuronska mreža* sadrži povratne veze između i unutar slojeva što modelu omogućuje pamćenje stanja. To je posebno korisno za obradu vremenski ovisnih podataka. Primjer primjene ove arhitekture su prepoznavanje govora i rukopisa.
- *Arhitektura sa slojevima pažnje* sadrži slojeve pažnje koji su bitni za pružanje konteksta određenom dijelu ulaza. Tako se prepoznaju bitni dijelovi ulaza te njihovi odnosi. Primjer takve arhitekture je *transformer* arhitektura [27] koja se koristi u naprednim jezičnim modelima.

7.1. Transformer arhitektura

U radu *Attention is all you need* [27] je predstavljena transformer arhitektura, model dubokog učenja. Izvorno je namijenjena za obradu prirodnog jezika, no našla je svoju primjenu i u područjima poput računalnog vida, analize zvuka i bioinformatici [35][26]. Trenutno najpoznatiji alat koji koristi transformer arhitekturu je ChatGPT [28] razvijen od tvrtke OpenAI.

Transformer arhitektura [24] se može sastojati od kodera, dekodera ili kombinacije oba elementa (Sl. 7.1). Koder pretvara ulazne podatke poput teksta u numeričke reprezentacije, dok dekoder pretvara numeričke reprezentacije u izlazne podatke. Alati za klasifikaciju ili identifikaciju će koristiti samo koder dio arhitekture, dok će alati za generiranje izlaza koristiti samo dekoder dio arhitekture. Alati koji koriste oba elementa mogu na primjer imati ulogu prevođenja ili sažimanja teksta.

Podaci prvo dolaze na sloj ugrađivanja ulaza (na slici *input embedding*). U ovom sloju se ulaz podijeli na niz tokena koji se zatim pretvaraju u numeričke reprezentacije. Najjednostavniji način jest pretvaranje tokena u redni broj tokena. Niz vrijednosti se zatim dopunjuje ili skraćuje na unaprijed zadanu duljinu dodavanjem određenih tokena ili brisanjem tokena. Vrijednosti tokena se zatim transformiraju u vektore

Sljedeći sloj je pozicijsko kodiranje. U ovom sloju se vektorima dobivenim iz prethodnog sloja dodaju vektori jednakih dimenzija čije su vrijednosti funkcije sinusa i kosinusa različitih frekvencija. Tako se tokenima dodaje informacija o poziciji unutar slijeda. Bez ovog sloja, model ne bi mogao razlikovati sljedove s istim, ali različito poredanim tokenima.

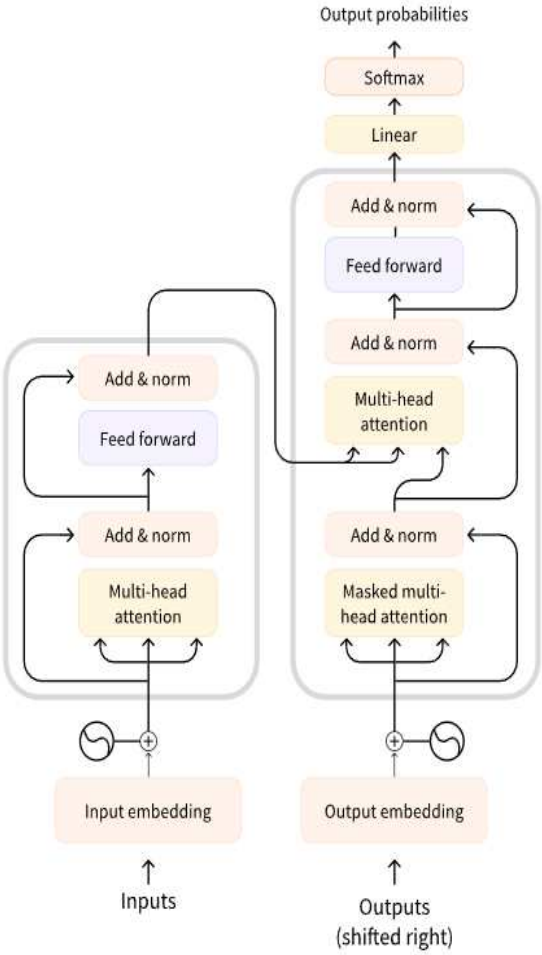
Podaci zatim ulaze u koder te se prosljeđuju na sloj *multi-head attention.* Taj sloj se sastoji od više glava koje nezavisno računaju izlaze s pomoću mehanizma pažnje. Mehanizam pažnje je mehanizam prilikom kojeg se uzimaju u obzir odnosi između tokena u slijedu. Drugim riječima, svakom od tokena se pridjeljuje kontekst. Rezultati svih glava se zatim objedinjuju, zbrajaju s podacima koji su bili ulaz u sloj te prosljeđuju dalje.

Podaci se zatim normaliziraju kako bi se smanjila varijanca podataka između slojeva te omogućio stabilan gradijent. Na taj se način olakšava učenje modela te model postiže bolje rezultate.

Sljedeći sloj je *unaprijedna mreža* (na slici *feed forward*). U ovom sloju se koristi nelinearna aktivacijsku funkcija koja omogućuje modelu da uči složene odnose između

podataka. Najčešće korištena aktivacijska funkcija u transformerima je ReLU funkcija. Nakon ovog sloja ponovno slijedi zbrajanje s podacima koji su bili ulaz u sloj i normalizacija podataka.

Izlaz iz kodera su numeričke reprezentacije ulaznog slijeda u kojima su skriveni odnosi i značenje tokena unutar ulaznog slijeda. Arhitektura dekodera je slična arhitekturi kodera, no model dubokog učenja u ovom radu koristi samo koder dio arhitekture pa neću ulaziti u detalje dekoder arhitekture.



Sl. 7.1: Trasformer arhitektura [25]

8. Implementacija

8.1. Programsko okruženje

Za implementaciju programskog rješenja sam koristio programski jezik *Python. Python* se ističe zbog jednostavne i interpretabilne sintakse zbog koje je vrlo jednostavno programirati u njemu. Osim toga, *Python* je vrlo svestran te sadrži bogatu kolekciju biblioteka.

Ovo su neke od poznatih biblioteka koje sam koristio prilikom izrade rješenja [36]:

1. *Matplotlib* je biblioteka za kreiranje grafova i vizualizacija.
2. *Pandas* je biblioteka koja nudi strukture podataka poput podatkovnih okvira koji omogućuju jednostavno rukovanje tabličnim podacima.
3. *NumPy* je biblioteka koja sadrži mnoštvo funkcija za napredne i optimizirane matematičke izračune.
4. *ESM* je javno dostupan set funkcija koji omogućuje pokretanje Esm-1b proteinskog jezičnog modela, a uključuje i rukovanje s FASTA datotekama [26].
5. *Scikit-learn* je biblioteka za strojno učenje koja nudi alate za modeliranje, pretprocesiranje, podjelu i evaluaciju podatka.
6. *XGBoost* je biblioteka sa XGBoost modelom strojnog učenja.

8.2. Jezični model

Proteinski slijed u FASTA zapisu gena je zapisan kao niz slova koja predstavljaju aminokiseline. Kako bih takav oblik informacije proslijedio na ulaz modela strojnog učenja, potrebno je transformirati podatke u niz brojčanih vrijednosti koje sadrže informacije o odnosima i kontekstu aminokiselina u slijedu. Stvoreni slijed nazivamo reprezentacijski vektor.

Za stvaranje reprezentacijskog vektora sam koristio javno dostupan proteinski jezični model ESM-1b [26] koji koristi transformer arhitekturu od 33 sloja sa 650 milijuna parametara te je treniran na skupu od približno 250 milijuna proteinskih sekvenci. Reprezentacije stvorene ESM modelima su predviđene za određivanje strukture proteina, prepoznavanje funkcionalnih regija u proteinu, klasifikaciju proteina i slično. Alat koristi samo koder dio arhitekture transformera, prima FASTA proteinski zapis, a vraća reprezentacijski vektor.

Stvaranje reprezentacija uz pomoć jezičnog modela računalno je zahtjevno, stoga sam koristio vanjske računalne resurse. Koristio sam računalo SUPEC sa 16 procesorskih jezgri te 64 GB radne memorije koje je dio usluge *Napredno računanje* Sveučilišnog računskog centra u Zagrebu [29].

8.3. Strojno učenje

U svom radu sam konstruirao dva modela strojnog učenja. Prvi model sam izgradio za identifikaciju gena rezistentnih na antibiotike, dok drugi model klasificira vrstu rezistencije antibiotika.

Pri identifikaciji sam objedinio sve skupove rezistentnih modela te dodao skup nerezistentnih gena iz UniProt baze. Podaci rezistentnih i nerezistentnih gena su jednako formatirani, osim što je za rezistentne gene ne postoji ARO index. Za taj skup podataka sam s pomoću proteinskog jezičnog modela Esm-1b stvorio reprezentacijske vektore. Skupu podataka sam zatim dodao ciljni stupac koji odgovara vrijednosti nula za nerezistentne gene, a jedinici za rezistentne gene. Definirao sam XGBoost model strojnog učenja te podijelio skup podataka na skup za treniranje i testiranje. Istrenirao sam model nad skupom za treniranje te sam proveo evaluaciju nad skupom za testiranje.

Pri klasifikaciji sam koristio samo one gene iz objedinjenog skupa koji stvaraju rezistenciju. Jedan gen može stvarati rezistenciju na više antibiotika pa su ciljne vrijednosti za neke gene bile višestruke. Prvi pristup tom problemu je nasumično uzimanje jednog antibiotika iz ciljnog stupca za gene s više ciljnih klasa te odbacivanje ostalih. Drugi pristup koji je pokazao bolje rezultate je korištenje funkcije *MultiOutputClassifier* iz *Scikit-learn* biblioteke koja koristi *One-Vs-Rest* strategiju pri klasifikaciji. *One-Vs-Rest* strategija gradi binarni model za svaku od klasa u kojem ispituje je li primjer pripada toj klasi. Izlaz ukupnog modela je lista rezultata pojedinačnih modela.

9. Evaluacija

Postoje četiri glavne mjere uspješnosti s pomoću kojih validiramo uspješnost modela. To su točnost, preciznost, odziv i F1 vrijednost. Za računanje vrijednosti tih mjera uspješnosti su nam važne informacije o ishodima klasifikacije. Ispravno pozitivni (eng. *true positive,* oznaka TP) su slučajevi gdje je model predvidio pripadnost klasi te primjer zaista pripada toj klasi. Ispravno negativni (eng. *false negative*, oznaka FN) su slučajevi gdje je model predvidio da primjer ne pripada klasi što je zaista istina. Lažno pozitivni (eng. *false positive*, oznaka FP) su slučajevi gdje je predviđena pripadnost klasi, ali primjer ne pripada klasi. Lažno negativni (eng. *false negative,* oznaka FN) su slučajevi gdje je predviđena pripadnost klasi što nije istina.

Matrica zabune jest tablični prikaz ishoda klasifikacije. Ona prikazuje točne i netočne klasifikacije koje je model napravio. Vrijednosti na padajućoj dijagonali su brojevi ispravnih predviđanja te klase dok sve ostale ćelije predstavljaju broj pogrešnih predviđanja između dviju pripadajućih klasa. Čim se više vrijednosti nalazi na padajućoj dijagonali, to je model bolji.

Točnost se definira kao udio točno klasificiranih primjera u skupu svih primjera (8.1). Preciznost se definira kao udio ispravno pozitivnih u skupu svih predviđenih pozitivnih primjera (8.2). Odziv je udio ispravno pozitivnih u skupu svih pozitivnih primjera (8.3). Specifičnost je udio ispravno negativnih u skupu svih negativnih primjera (8.4)

$$točnost = \frac{TP + TN}{TP + TN + FP + FN} \tag{8.1}$$

$$preciznost = \frac{TP}{TP + FP} \tag{8.2}$$

$$odziv = \frac{TP}{TP + FN} \tag{8.3}$$

$$specifičnost = \frac{TN}{TN + FT} \tag{8.4}$$

F1-vrijednost je mjera uspješnosti koja predstavlja harmonijsku sredinu između preciznosti i odziva (8.5). Ova mjera osigurava da i preciznost i odziv daju dobre rezultate kako bi se izbjeglo manipuliranje rezultatima. S parametrom β se odabire hoće li se dati naglasak na preciznost ili odziv.

$$F1\,vrijednost = (1 + \beta^2)\frac{preciznost \cdot odziv}{\beta^2\,preciznost + odziv} \tag{8.5}$$

ROC krivulja (eng. *Receiver Operating Characteristic*) je još jedna grafička metoda za prikazivanje uspješnosti modela. Prikazuje odnos odziva i specifičnosti za različite pragove donošenja odluka. AUC (eng. *Area Under the Curve*) je površina ispod ROC krivulje te predstavlja vjerojatnost točne klasifikacije. Što je graf udaljeniji od rastuće dijagonale to bolje diskriminira između klasa što povlači i da je AUC vrijednost veća.

10. Rezultati

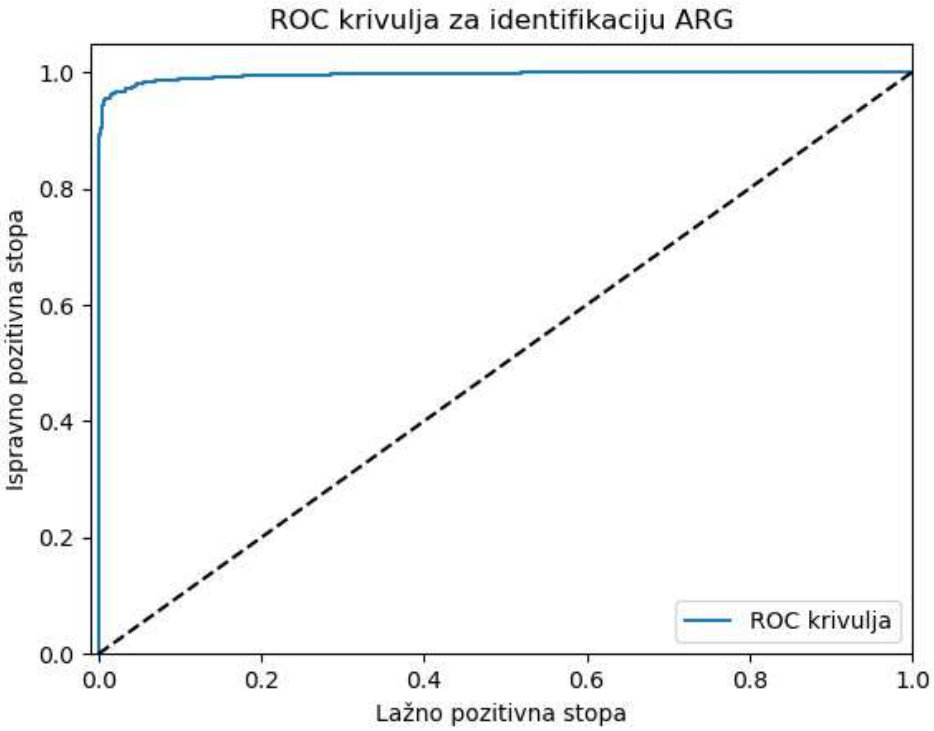
Identifikacija ARG-a je iz prvog pokušaja pokazala odlične rezultate gdje sve mjere uspješnosti prelaze vrijednost od 0.96. ROC krivulja prikazuje kako model odlično diskriminira između dviju klasa (Sl. 10.1) što potvrđuje površina ispod ROC krivulje s vrijednosti 0.97. Iz matrice zbunjenosti (Sl. 10.2) također možemo vidjeti da model odlično diskriminira između gena koji stvaraju rezistenciju i gena koji ju ne stvaraju, no možemo vidjeti da model ima oko tri puta više lažno negativnih, nego lažno pozitivnih vrijednosti što znači da moramo biti oprezni ako model klasificira gen kao nerezistentan.

Klasifikacija je s druge strane davala različite rezultate tijekom izgradnje. Najlošije rezultate je davao model gdje sam uzimao nasumične ciljne vrijednosti za podatke s više ciljnih klasa. Niti jedna mjera uspješnosti u tom slučaju nije prelazila 0.86. Sve mjere uspješnosti su prešle prag od 0.87 nakon što sam počeo koristiti *MultiOutputClassifier metodu* za predviđanje višestrukih klasa rezistencije. Dodatno poboljšanje rezultata sam postigao grupiranjem klasa antibiotika u logične grupe. Rezultati u tom trenutku su prikazani u drugom stupcu tablice Tablica 10.1. Najbolje rezultate sam postigao kada sam iz skupa podataka izbacio sve podatke koji imaju više ciljnih klasa. Ti podaci su prikazani u trećem stupcu iste tablice. Na matrici zbunjenosti (Sl. 10.3) se vidi da model odlično diskriminira između klasa rezistencije unatoč velikoj razlici u brojnosti klasa.

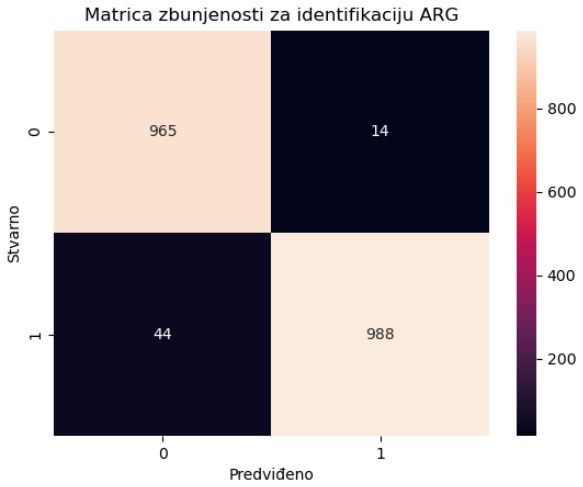
U referentnom radu su za stvaranje reprezentacija, klasifikaciju i identifikaciju ARG-a koristili iste metode kao u ovom radu. Sve mjere uspješnosti kod identifikacije osim preciznosti su u mojem modelu bolje od mjera u referentnom radu (Tablica 10.2). Bolju uspješnost modela pri identifikaciji ARG-a sam postigao zbog korištenja 4.4 puta većeg skupa podataka. S druge strane, mjere uspješnosti pri klasifikaciji su bolje u referentnom radu gdje su koristili 2.8 puta veći skup podataka.

Modeli za klasifikaciju ARG-a u referentnom radu i ovom radu pokazuju vrlo slične rezultate s odstupanjima od 0.02 u korist referentnog rada. Ako bi se izjednačili skupovi podataka, razlika između tih vrijednosti bi bila još manja.

Rezultati identifikacije i klasifikacije pokazuju da bi se ovaj model mogao dobro uklopiti u praktičnu primjenu. Liječnici bi s velikom sigurnošću mogli zaključiti koje tretmane ne smiju prepisati pacijentu, a koji će im pomoći pri liječenju.



Sl. 10.1: ROC krivulja za identifikaciju ARG



Sl. 10.2: Matrica zbunjenosti za identifikaciju ARG

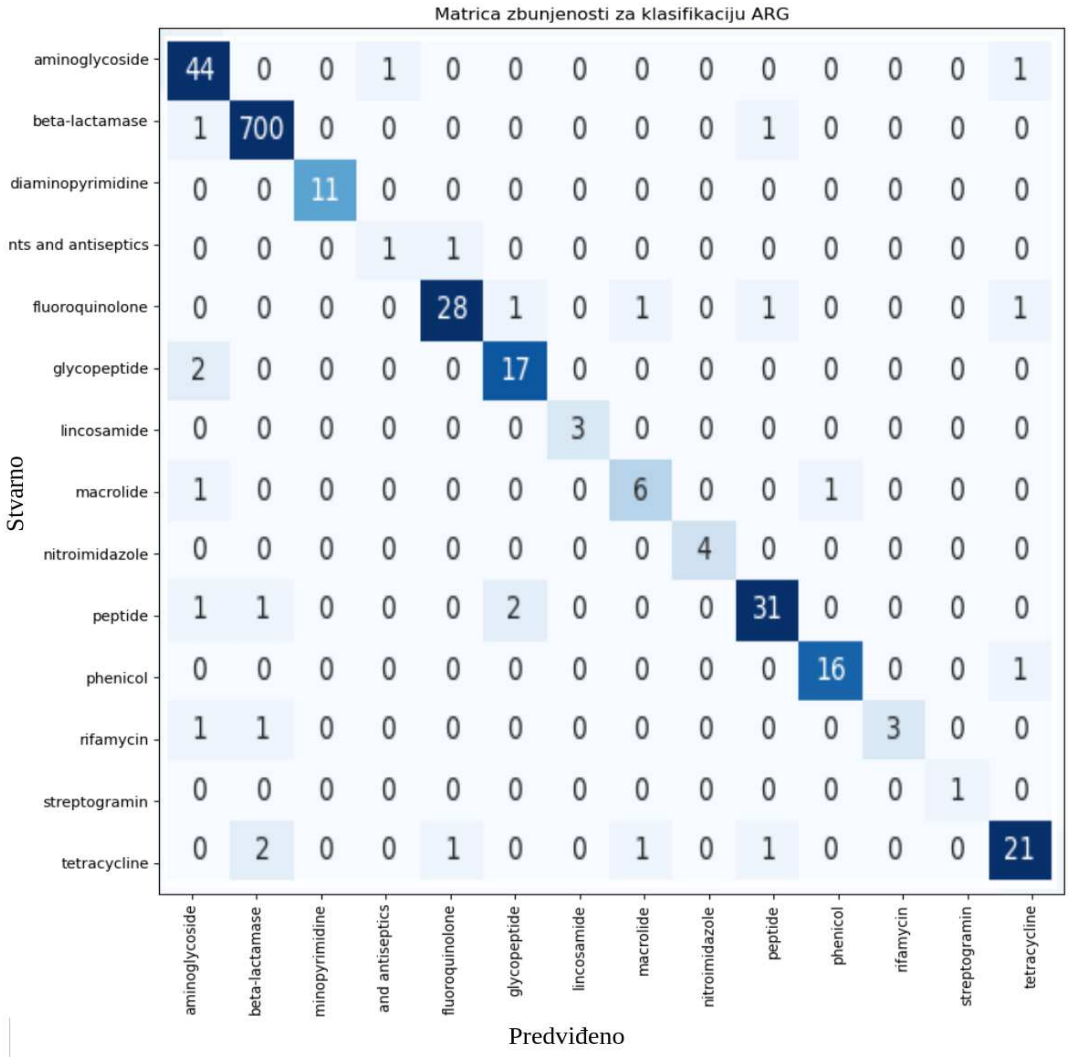
U matrici zbunjenosti vrijednost 0 predstavlja gene koji ne stvaraju rezistenciju, dok vrijednost 1 predstavlja gene koji stvaraju rezistenciju.

	Identifikacija	Klasifikacija (s genima koji pripadaju više klasa rezistencije)	Klasifikacija (bez gena koji pripadaju više klasa rezistencije)
Točnost	0.97	0.92	0.97
Preciznost	0.99	0.94	0.97
Odziv	0.96	0.9	0.97
F1 vrijednost	0.97	0.92	0.97

Tablica 10.1: Mjere uspješnosti za identifikaciju i klasifikaciju na skupu podataka navedenim u poglavlju 4.1 i metodama opisanim u poglavlju 8.

	Identifikacija	Klasifikacija (s genima koji pripadaju više klasa rezistencije)
Točnost	0.91	0.99
Preciznost	1	0.99
Odziv	0.83	0.99
F1 vrijednost	0.9	0.99

Tablica 10.2: Mjere uspješnosti za identifikaciju i klasifikaciju u referentnom radu nad podacima navedenim u poglavlju 4.3 te istim metodama kao u ovom radu.



Sl. 10.3: Matrica zbunjenosti za klasifikaciju ARG nakon uklanjanja primjera s više klasa rezistencije.

Zaključak

U ovom radu sam proučavao kako se može riješiti problem identifikacije i klasifikacije gena koji stvaraju rezistenciju. Za to sam koristio proteinske sljedove gena i alate strojnog i dubokog učenja.

Pristup identifikaciji i klasifikaciji gena koji stvaraju rezistenciju s pomoću proteinskog jezičnog modela i strojnog učenja je pokazao odlične rezultate. Usporedba s rezultatima rada „*PLM-ARG: antibiotic resistance gene identification using a pretrained protein language model*” [13] je pokazala da modeli koji su trenirani nad većim skupom podataka daju bolje rezultate.

XGBoost model je u ovom radu s točnosti od 0.97 i F1 vrijednosti od 0.97 identificirao gene koji stvaraju rezistenciju. Skup podataka je 4.4 puta veći od korištenog u referentnom radu gdje je XGBoost model s točnosti od 0.91 i F1 vrijednosti od 0.9 identificirao gene.

Pri klasifikaciji u ovom radu točnost iznosi 0.97, a F1 vrijednost 0.97 U referentnom radu su obje vrijednosti 0.99, a korišten je 2.8 puta veći skup podataka.

Iako su jezični modeli za generiranje odgovora i multimedije aktualna tema u široj javnosti, ovaj rad je primjer kako se oni mogu odlično primijeniti i u drugim područjima poput medicine i biologije.

Napredni algoritmi strojnog učenja, kao što je XGBoost model, imaju izvrsne karakteristike poput velike brzine treniranja na velikim skupovima podataka i sposobnosti sprječavanja prenaučenosti.

Ovakav model bi mogao pronaći primjenu praksi. Mogao bi pomoći pri odabiru prikladnih antibiotika ili bi mogao omogućiti stvaranje individualiziranih tretmana na temelju informacija o rezistencijama.

Literatura

[1] It for all, 8 helpful examples of artificial intelligence. Poveznica: https://www.iotforall.com/8-helpful-everyday-examples-of-artificial-intelligence; Pristupljeno 13. lipnja 2024.

[2] Dr Mohsen Naghavi, Global burden of bacterial antimicrobial resistance in 2019: a systematic analysis, Lancet, 399 606 (2022).

[3] Biolabtests, Kirby-Bauer. Poveznica: https://biolabtests.com/antimicrobial-testing-methods/kirby-bauer-disk-diffusion-method/; pristupljeno 8. lipnja 2024.

[4] Yiming Zhong, Feng Xu, Jinhua Wu, Jeffrey Schubert, Marilyn M. Li, "Application of Next Generation Sequencing in Laboratory Medicine," Ann Lab Med, vol. 41, no. 1, pp. 25-43 (2021)

[5] Wellcome, It's time to fix the antibiotic market. Poveznica: https://wellcome.org/news/its-time-fix-antibiotic-market; pristupljeno 8. lipnja 2024.

[6] CDC, About Antimicrobial Resistance, (2024, travanj). Poveznica: https://www.cdc.gov/antimicrobial-resistance/about/?CDC_Aaref_Val=https://www.cdc.gov/drugresistance/about/how-resistance-happens.html; pristupljeno 28. svibnja 2024.

[7] George Vernikos and Duccio Medini, Horizontal Gene Transfer and the Role, *Evolutionary Biology: Genome Evolution, Speciation, Coevolution and Origin of Life*, 169-190 (2014, lipanj)

[8] CARD, CARD Data, (2024, veljača). Poveznica: https://card.mcmaster.ca/download; preuzeto 11. ožujka 2024.

[9] Neelanjana Pandey; Marco Cascella. "Beta-Lactam Antibiotics". (4.6. 2023.)

[10] UniProt (2024, veljača). Poveznica: https://www.uniprot.org/; preuzeto 20. svibnja 2024.

[11] Jia B, Raphenya AR, Alcock B. et al. CARD 2017: expansion and model-centric curation of the comprehensive antibiotic resistance database. Nucleic Acids Res 2017;45:D566–73.

[12] Drugs, Antibiotics 101. Poveznica: https://www.drugs.com/article/antibiotics.html; Pristupljeno 13. lipnja 2024.

[13] Wu J., Ouyang J., Qin H., Zhou J., Roberts R., Siam R., Wang L., Tong W., Liu Z., Shi T., PLM-ARG: PLM-ARG: antibiotic resistance gene identification using a pretrained protein language model, Bioinformatics, 39, 11 (2023)

[14] S F Altschul, W Gish, W Miller, E W Mers, D J Lipman, Basic local alignment search tool, *Journal of Molecular Biology*, 215, 3, (1990)

[15] IBM, What is machine learning (ML)? Poveznica: https://www.ibm.com/topics/machine-learning; Pristupljeno 10. lipnja 2024.

[16] Overfitting in machine learning. Poveznica: https://www.javatpoint.com/overfitting-in-machine-learning; Pristupljeno (10. lipnja 2024.)

[17] Bojana Dalbelo Bašić i Jan Šnajder. Strojno učenje. Uvod u umjetnu inteligenciju, FER, UNIZG (2020) Poveznica: https://www.fer.unizg.hr/_download/repository/UI-2020-10-StrojnoUcenje.pdf

[18] Tianqi, C., Carlos, G., XGBoost: A Scalable Tree Boosting System, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco (2016)

[19] Medium, Interview questions for XG Boost. Poveznica: https://medium.com/@thedatabeast/interview-questions-for-xg-boost-2d4c7b7f4bbf; Pristupljeno 11. lipnja 2024.

[20] IBM, What is deep learning? Poveznica: https://www.ibm.com/topics/deep-learning; Pristupljeno 11. lipnja 2024.

[21] IBM, What is a neural network? Poveznica: https://www.ibm.com/topics/neural-networks; Pristupljeno 11. lipnja 2024.

[22] Marko Čupić. Prirodom inspirirani optimizacijski algoritmi. Uvod u umjetnu inteligenciju, FER, UNIZG (2012) Poveznica: https://www.fer.unizg.hr/_download/repository/UI_13_PrirodomInspiriraniOptimizacijskiAlgoritmi.pdf

[23] IBM, Deep learning architectures. Poveznica: https://developer.ibm.com/articles/cc-machine-learning-deep-learning-architectures/; Pristupljeno 11. lipnja 2024.

[24] J.Alammar, The Illustrated Transformer (2018). Poveznica: https://jalammar.github.io/illustrated-transformer/; Pristupljeno 11. lipnja 2024.

[25] How do Transformers work?, Hugging Face. Poveznica: https://huggingface.co/learn/nlp-course/en/chapter1/4?fw=pt; Pristupljeno 22. svibnja 2024.

[26] Evolutionary Scale Modeling, ESM-1b (2021, lipanj). Poveznica: https://github.com/facebookresearch/esm; preuzeto 11. ožujka 2024.

[27] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I., Attention Is All You Need, *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach (2017)

[28] OpenAI, ChatGPT (2022.). Poveznica: https://chatgpt.com/

[29] Napredno računanje, Sveučilišni računski centar. Poveznica: https://www.srce.unizg.hr/napredno-racunanje; pristupljeno 20. svibnja 2024.

[30] Arango-Argoty, G.A., et al. ARGminer: a web platform for the crowdsourcing-based curation of antibiotic resistance genes. Bioinformatics 2020;36(9):2966-2973.

[31] Doster, E., et al. MEGARes 2.0: a database for classification of antimicrobial drug, biocide and metal resistance determinants in metagenomic sequence data. Nucleic Acids Res 2020;48(D1):D561-D569.

[32] Feldgarden, M., et al. Validating the AMRFinder tool and resistance gene database by using antimicrobial resistance genotype-phenotype correlations in a collection of isolates. 2019;63(11):e00483-00419.

[33] Kleinheinz, K.A., Joensen, K.G. and Larsen, M.V. Applying the ResFinder and VirulenceFinder web-services for easy identification of acquired antibiotic resistance and E. coli virulence genes in bacteriophage and prophage nucleotide sequences. Bacteriophage 2014;4(1):e27943.

[34] Li, Y., et al. HMD-ARG: hierarchical multi-task deep learning for annotating antibiotic resistance genes. Microbiome 2021;9(1).

[35] Hugging face, Vision Transformer. Poveznica;https://huggingface.co/docs/transformers/en/model_doc/vit; Pristupljeno 13. lipnja 2024.

[36] JetBrains, Python Software Foundation, Python Developers Survey 2018. Poveznica: https://www.jetbrains.com/research/python-developers-survey-2018/; Pristupljeno 13. lipnja 2024.

Sažetak

Rezistencija na antibiotike je aktualan problem u medicini. Problem je ozbiljan zbog horizontalnog prijenosa gena koji brzo širi rezistenciju između bakterija. Jedan od načina na koji se može identificirati i klasificirati rezistencija su tehnologije umjetne inteligencije poput jezičnih modela i strojnog učenja.

U ovom radu sam iskoristio skup javno dostupnih sljedova gena koji stvaraju rezistenciju iz CARD baze podataka te sam s pomoću ESM-1b proteinskog jezičnog modela stvorio numeričke reprezentacije tih gena. Zatim sam trenirao dva XGBoost modela strojnog učenja. Jednog sam trenirao da identificira, a drugog da klasificira gene koji stvaraju rezistenciju.

Model može s točnosti od 97 % identificirati uzrokuje li neki gen rezistenciju, a ako gen stvara rezistenciju model može s točnosti od 97 % odrediti na koji je antibiotik gen rezistentan. Takav model bi mogao naći mjesto u praksi kao pomoć prilikom odabira tretmana.

Summary

Antibiotic resistance is a current problem in medicine. One of the ways to identify and classify resistance is through artificial intelligence technologies such as language models and machine learning.

In this study, I utilized a publicly available dataset of resistance gene sequences from the CARD database and used a protein language model to create numerical representations of these genes. I trained two XGBoost machine learning models. One model was trained to identify, and the other to classify genes that confer resistance.

The trained model can identify with 97 % accuracy whether a gene causes resistance, and if the gene causes resistance, the model can determine with 97 % accuracy to which antibiotic the gene is resistant. Such a model could find a place in practice as an aid in selecting treatment.

Skraćenice

ARG	*Antibiotic resistance gene*	gen rezistencije na antibiotik
ARO	*Antibiotic resistance ontology*	ontologija gena rezistencije