

Stvaranje novih peptida upotrebom modela varijacijskog autoenkodera

Ronta, Antonio

Master's thesis / Diplomski rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:190:387312>

Rights / Prava: [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2024-10-21**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Engineering](#)



SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Diplomski sveučilišni studij računarstva

Diplomski rad

**Stvaranje novih peptida upotrebom modela
varijacijskog autoenkodera**

Rijeka, studeni 2023.

Antonio Ronta
0069078024

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Diplomski sveučilišni studij računarstva

Diplomski rad

**Stvaranje novih peptida upotrebom modela
varijacijskog autoenkodera**

Mentori:

doc. dr. sc. Goran Mauša

Rijeka, studeni 2023.

Antonio Ronta
0069078024

Umjesto ove stranice umetnuti zadatak
za završni ili diplomski rad

Izjava o samostalnoj izradi rada

Izjavljujem da sam samostalno izradio ovaj rad.

Rijeka, studeni 2023.

Antonio Ronta

Zahvala

Zahvaljujem se mentoru doc. dr. sc. Goranu Mauši i asistentu Marku Njirjaku na pruženoj pomoći prilikom izrade rada, stručnosti te velikom strpljenju i ljudskosti pomoću koje su me motivirali za izradu rada. Vrata njihovih ureda uvijek su bila otvorena.

Također, zahvaljujem se svojoj obitelji, prijateljima i djevojci na neizmjerljivoj podršci u trenucima kada mi je bila najpotrebnija. Najveće hvala Onomu kojem je sve moguće.

Sadržaj

Popis slika	viii
Popis tablica	x
Popis isječaka programskog koda	x
1 Uvod	1
1.1 Kemija peptida	2
2 Metodologija	5
2.1 Autoenkoderi	5
2.1.1 Smanjenje dimenzionalnosti	5
2.1.2 Općenito o autoenkoderima	8
2.1.3 Autoenkoderi u svrhu generativnog procesa	11
2.1.4 Varijacijski autoenkoderi	13
2.1.5 Regularizacija	15
2.2 Rekurentne neuronske mreže	18
2.3 Priprema i obrada podataka	22
2.3.1 One-hot enkodiranje	24

Sadržaj

3 Implementacija VAE modela	26
3.1 Enkoder	26
3.2 Dekoder	28
3.3 VAE model	29
3.3.1 Funkcija gubitka	29
3.3.2 Treniranje modela	30
4 Rezultati	31
4.1 Analiza dekodiranih peptida	37
5 Zaključak	48
Bibliografija	50
Sažetak	53

Popis slika

1.1	Svih 20 aminokiselina koje se pojavljuju u ljudskom organizmu, grupirane prema svojim poznatim svojstvima	3
2.1	Ilustracija principa redukcije dimenzionalnosti s enkoderom i deko- derom	6
2.2	Ilustracija tehnike PCA koja traži najbolji linearni podprostor koris- teći linearnu algebru	8
2.3	Ilustracija autoenkodera s njegovom funkcijom gubitka	9
2.4	Ilustracija veze između linearnog autoenkodera i tehnike PCA	10
2.5	Smanjujući dimenzionalnost, želimo zadržati strukturu koja postoji među podacima	11
2.6	Ilustracija principa rada dekodiranja točaka nasumično uzorkovanih iz latentnog prostora	12
2.7	Prikaz lošeg latentnog prostora iz kojeg se dobivaju vrijednosti bez značaja	13
2.8	Prikaz razlike treniranja autoenkodera i VAE	14
2.9	Prikaz varijacijskog autoenkoder	15
2.10	Prikaz razlike dobro reguliranog latentnog prostora i lošeg	17
2.11	Ilustracija RNN-a	19
2.12	Ilustracija LSTM-a	20
2.13	Podjela peptida po vrsti u DRAMP bazi	22

Popis slika

2.14	Raspodjela peptida u početnom skupu	23
3.1	Prikaz slojeva enkodera	27
3.2	Prikaz slojeva dekodera	29
4.1	Prikaz gubitaka tijekom treninga	32
4.2	Prikaz ukupnog gubitka kroz trening	33
4.3	Trening podaci enkodirani u latentni prostor	34
4.4	Testni podaci enkodirani u latentni prostor	35
4.5	Rekonstruirani trening podaci enkodirani u latentni prostor	36
4.6	Rekonstruirani testni podaci enkodirani u latentni prostor	36
4.7	Usporedba distribucija dužina novih peptida sa trening peptidima .	38
4.8	Usporedba dužina novih peptida sa trening skupom	39
4.9	Razlika dužina duljih novih peptida sa trening skupom	39
4.10	Razlika dužina kraćih novih peptida sa trening skupom	40
4.11	Usporedba distribucija dužina novih peptida sa testnim peptidima .	41
4.12	Usporedba dužina novih peptida sa testnim skupom	42
4.13	Razlika dužina kraćih novih peptida sa testnim skupom	42
4.14	Rezultati sličnosti ovisno o indeksu - novi peptidi s trening peptidima	44
4.15	Distribucija sličnosti - novi peptidi s trening peptidima	44
4.16	Rezultati sličnosti ovisno o indeksu - novi peptidi s testnim peptidima	45
4.17	Distribucija sličnosti - novi peptidi sa testnim peptidima	45

Popis tablica

2.1	Enkodirane aminokiseline koristeći <i>One-hot</i>	24
2.2	Enkodirani peptid 'RWRW'	25
4.1	Vrijednosti gubitaka istreniranog VAE modela	31
4.2	Matrica zabune predviđanja aktivnosti novih peptida - trening skup	46
4.3	Matrica zabune predviđanja aktivnosti novih peptida - testni skup .	47

Poglavlje 1

Uvod

U posljednjim godinama duboko učenje primjenom generativnih modela postalo je središnja točka interesa i inovacija u različitim područjima, potkrijepljeno njihovom izvanrednom sposobnošću stvaranja realističnog sadržaja u različitim modalitetima, kao što su slike, tekstovi i zvukovi. Među istaknutim obiteljima dubokih generativnih modela, generativne suparničke mreže (*Generative Adversarial Networks*, GANs) i varijacijski autoenkoderi (*Variational Autoencoders*, VAEs) pokazali su izvanredan potencijal. Primjenom velikih skupova podataka te korištenjem precizno osmišljenih arhitektura neruonskih mreža i naprednih metoda treniranja, koriste se za stvaranje pouzdanih i kreativnih rješenja sve veće vjernosti [1].

Zbog velike popularnosti, primjene dubokih generativnih modela proširuju se izvan konvencionalnih područja generiranja medija te se sve više počinju koristiti u znanstvenim i kemijskim područjima, a s tim i kao pomoć u dizajnu novih lijekova. Nužnost brzog razvoja novih terapijskih molekula, usred rastućih troškova i produljenih vremenskih okvira koji karakteriziraju razvoj lijekova, potaknula je istraživanje računalnih strategija koje mogu ubrzati ciljanu generaciju i ispitivanje molekula s određenim terapijskim svojstvima.

Ovaj rad istražuje dizajn antimikrobnih peptida (AMP) kao odgovor na globalni porast rezistentnosti na postojeće antibiotike; prijetnju koja može godišnje odnijeti milijune života i stajati milijarde dolara do 2050. godine, na temelju sadašnjih trendova [2]. AMP-ovi predstavljaju nadu u borbi protiv porasta otpornosti

Poglavlje 1. Uvod

na antibiotike, nudeći obećavajuće alternative tradicionalnim antibioticima, zbog njihove izvanredne strukturalne raznolikosti, snažne antimikrobne aktivnosti i smanjene sklonosti razvoja rezistencije. Izazov u dizajniranju takvih terapijskih peptida leži u snalaženju u složenom prostoru molekula, kompleksnim odnosima između strukture i aktivnosti te mnogim drugim konkurirajućim ograničenjima poput aktivnosti, toksičnosti, troškovi sinteze i stabilnosti [2].

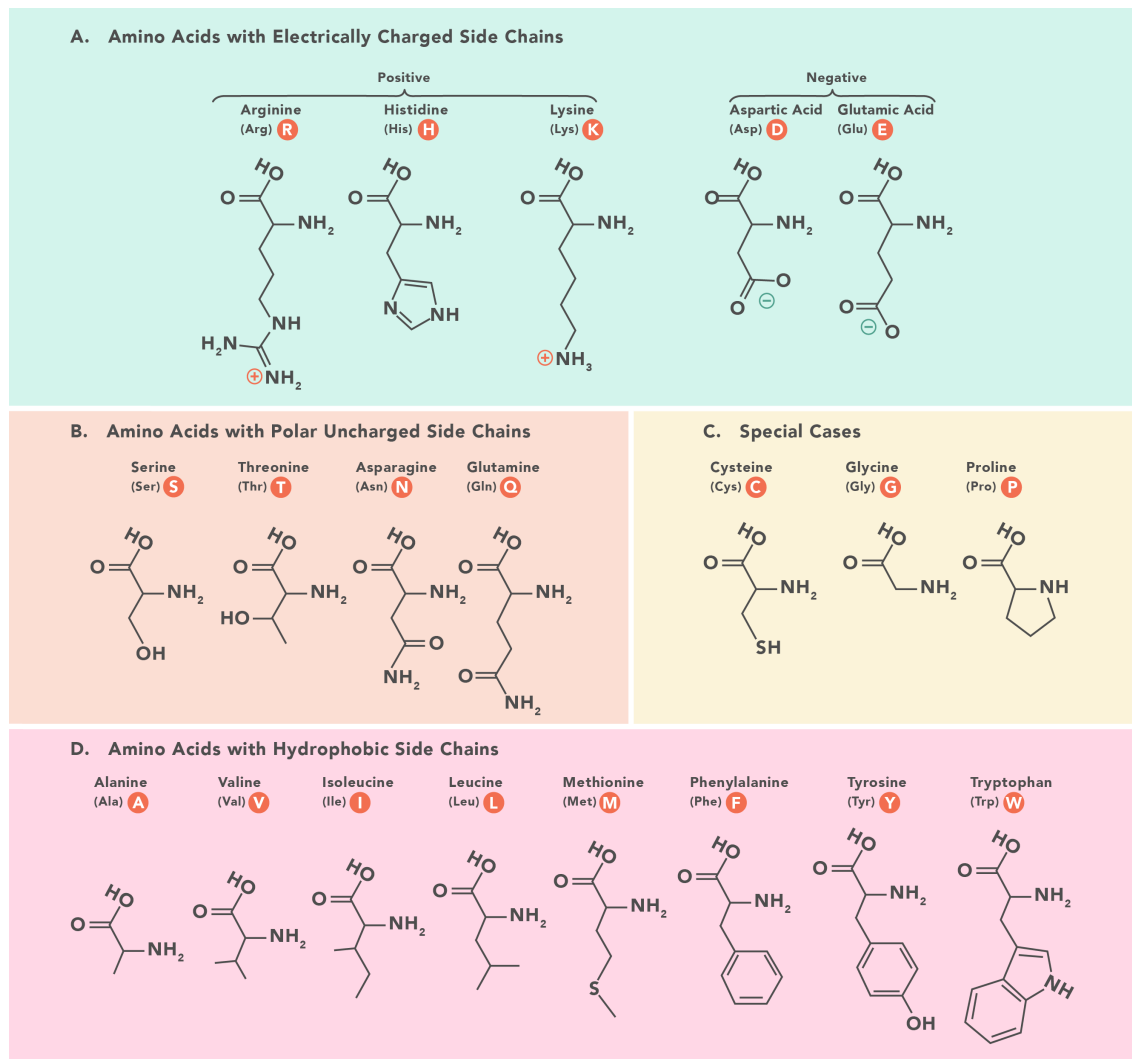
Ovaj diplomski rad izrađen je u okviru projekta „Dizajn katalitički aktivnih peptida i peptidnih nanostruktura” s oznakom UIP-2019-04-7999, a fokus je stavljen na primjenu varijacijskog autoenkodera u kontekstu dizajniranja antimikrobnih peptida. VAE modeli pružaju jedinstvenu sposobnost stvaranja kontinuiranog latentnog prostora koji olakšava generaciju novih AMP-ova, omogućava neprekidno interpoliranje između postojećih sekvenci te optimizaciju svojstava peptida. Specifičnom implementacijom ovog VAE modela, koristeći LSTM sloj rekurentnih neuronskih mreža te podešavanjem parametara slojeva i funkcija gubitaka, proučava se razumijevanje modela nad sekvencama peptida u svrhu generiranja. Kroz sustavno istraživanje mehanizama rada VAE modela i njegove karakteristične primjene, ovaj rad za cilj ima prikazati inovativni pristup u oblikovanju novih peptida.

1.1 Kemija peptida

Kako bi se rad, njegova tematika i cilj bolje objasnili, potrebno je dati kratak pregled peptida, što su, od čega se sastoje te kako se kategoriziraju.

Aminokiseline predstavljaju molekularne komponente koje služe kao osnovni građivni blokovi peptida. Postoji preko 500 poznatih aminokiselina koje se pojavljuju u prirodi od kojih se samo 20 pojavljuje u ljudskom organizmu. Devet od njih je ključno za održavanje ljudskog života te ih je neophodno unositi u ljudsko tijelo putem hrane ili pića, dok se preostalih jedanaest sintetizira unutar ljudskog organizma. Uz to, u prirodi se rijetko nalaze u svom slobodnom stanju. Najčešće ih nalazimo međusobno povezane u obliku makromolekula peptida i proteina [3]. Svaka od njih specifična je na svoj način te ima različita svojstva i različite funkcije, kao što je vidljivo sa slike 1.1.

Poglavlje 1. Uvod



Slika 1.1 Svih 20 aminokiselina koje se pojavljuju u genetskom kodu, grupirane prema svojim poznatim svojstvima, preuzeto iz [4]

Neka od poznatijih svojstava aminokiselina su [4]:

- hidrofobnost - hidrofobne aminokiseline ne vole vodena okruženja, dok hidrofilne da,
- aromatičnost - aromatične aminokiseline sadrže aromatski prsten, stabilnu cikličku strukturu,

Poglavlje 1. Uvod

- naboj - aminokiselina može imati neutralan, pozitivan ili negativan naboj,
- polaritet - polarna aminokiselina sadrži neravnomjernu raspodjelu elektrona po molekuli i može imati neutralan, pozitivan ili negativan naboj, dok nepolarna aminokiselina ima ravnomjernu raspodjelu elektrona i neutralan naboj,
- alifacitet - alifatska aminokiselina je i nepolarna i hidrofobna,
- amfifilnost - amfifilna aminokiselina je lipofilna i hidrofilna.

Također, svaka aminokiselina sastoji se od 2 funkcionalne skupine [3]:

- amino skupina ($-\text{NH}_2$) - početna točka aminokiseline (N-kraj),
- karboksilna skupina ($-\text{COOH}$) - krajnja točka aminokiseline (C-kraj),
- između spomenutih skupina specifični je bočni lanac koji svakoj aminokiselini određuje njezina svojstva.

Peptidi su kratki lanci aminokiselina međusobno povezani preko peptidnih veza, gdje se peptidna veza formira između N-kraja jedne aminokiseline i C-kraja prethodne aminokiseline u lancu. Manji se peptidi (do 20 aminokiselina) nazivaju oligopeptidima; spoj dviju aminokiselina naziva se dipeptid, triju aminokiselina tripeptid, četiriju aminokiselina tetrapeptid, itd. Polipeptidi su duži, nerazgranati peptidni lanci duljine do 50 aminokiselina. Ako polipeptidni lanac sadrži više od 50 aminokiselina, poznat je kao protein [5].

Kao što se razlikuju aminokiseline, razlikuju se i peptidi. Peptidi se svrstavaju u različite skupine na temelju njihove funkcije i izvora. Neki od njih uključuju gljivične peptide, biljne peptide, bakterijske ili antibiotske peptide, peptide cjepiva, peptide otrova itd.

Poglavlje 2

Metodologija

U ovom dijelu napravljen je pregled važnih principa vezanih za smanjenje dimenzionalnosti i autoenkodere koji će biti koristan za razumijevanje samih *Variational Autoencoders (VAEs)*. Zatim će biti pokazano zašto se autoenkodere ne mogu koristiti za generiranje novih podataka te će kao rješenje tog problema biti predstavljeni varijacijski autoenkodere, na čijoj osnovi je implementiran ovaj generativni model.

2.1 Autoenkodere

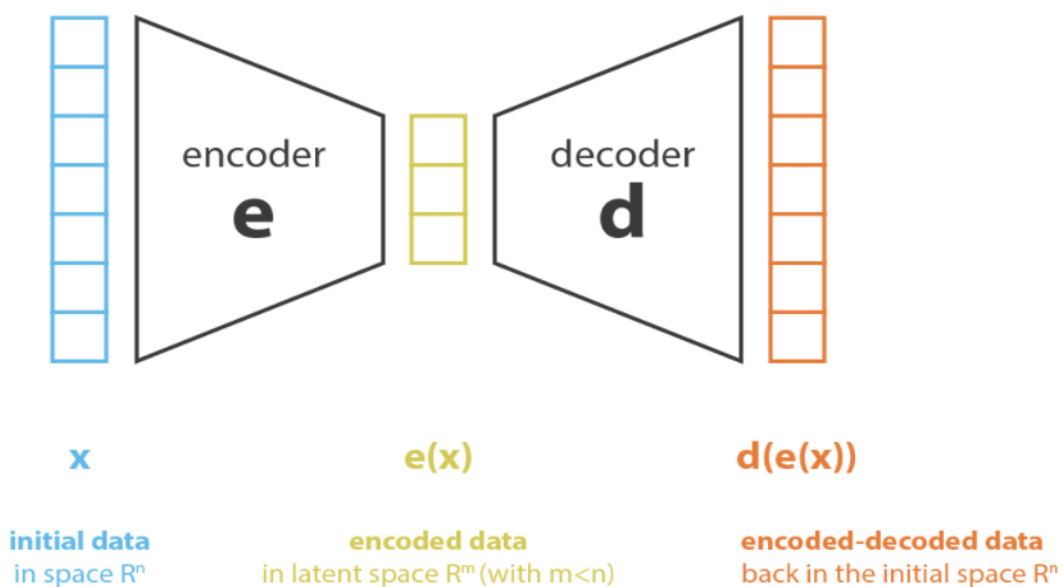
2.1.1 Smanjenje dimenzionalnosti

U strojnom učenju smanjenje dimenzionalnosti je proces smanjenja broja značajki koje opisuju neki skup podataka. Smanjenje broja značajki vrši se odabirom, (eng. *selection*) postupkom kojim su sačuvane samo neke postojeće značajke ili ekstrakcijom (eng. *extraction*) postupkom kojim se smanjeni broj novih značajki stvara na temelju skupa starih značajki. Ovi postupci vrlo su korisni u mnogim situacijama koje zahtijevaju podatke niske dimenzionalnosti poput vizualizacije podataka, pohrane podataka te zadataka koji zahtijevaju veliku komputacijsku moć računala. Iako postoji mnogo različitih metoda smanjenja dimenzionalnosti, u nastavku će biti predstavljeni principi koje koristi većina metoda [6].

Enkoder je proces s kojim dobivamo reprezentaciju "novih značajki" iz repre-

Poglavlje 2. Metodologija

zentacije "originalnih značajki" (odabirom ili ekstrakcijom), dok je dekodeo obrnuti gore navedeni proces. S tako definiranim pojmovima enkodera i dekodera smanjenje dimenzionalnosti se može protumačiti kao kompresija podataka gdje enkoder komprimira podatke iz početnog prostora u kodirani prostora, tzv. latentni, vektorski prostor, dok ih dekodeo dekomprimira iz latentnog prostora u početni. Ovisno o početnoj distribuciji podataka, dimenziji dobivenog latentnog prostora i definiciji enkodera - dekodera, ova kompresija može biti s gubitkom, što znači da se dio informacija gubi tijekom procesa kodiranja i ne može se povratiti tijekom dekodiranja [6, 1].



Slika 2.1 Ilustracija principa redukcije dimenzionalnosti s enkoderom i dekoderom, preuzeto iz [1]

Pronalazak najboljeg enkoder - dekodeo para, glavni je cilj metode redukcije dimenzionalnosti kojim se ostvaruje [1]:

- zadržavanje maksimalne količine informacija prilikom enkodiranja,
- minimalan broj pogrešaka prilikom rekonstrukcije podataka dekodiranjem.

Poglavlje 2. Metodologija

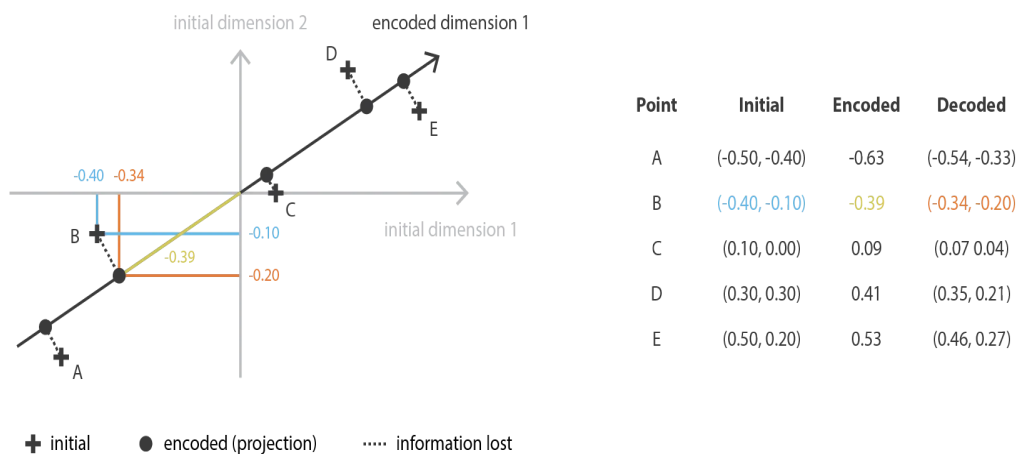
Ako skupinu enkodera i dekodera koje razmatramo označimo redom E i D, tada se problem smanjenja dimenzionalnosti može zapisati:

$$(e^*, d^*) = \underset{(e,d) \in E \times D}{\operatorname{argmin}} \epsilon(x, d(e(x))) \quad (2.1)$$

gdje $(x, d(e(x)))$ definira mjeru pogreške rekonstrukcije između ulaznih podataka x i enkodiranih - dekodiranih podataka $d(e(x))$. Ovaj izraz u osnovi kvantificira razliku između originalnih ulaznih podataka i njihove rekonstrukcije nakon procesa enkodiranja i dekodiranja [1]. Do kraja rada, dimenzija početnog (dekodiranog) prostora biti će označavana sa n dok će dimenzija latentnog (enkodiranog) prostora biti označavana sa m .

S ciljem uspostavljanja temelja za kasnije uspoređivanje tehnike analize glavnih komponentata (eng. *Principal components analysis (PCA)*) i autoenkodera, ukratko će biti objašnjene osnove tehnike PCA. Osnovna ideja tehnike PCA je izgraditi nove neovisne značajke kao linearnu kombinaciju starih značajki tako da projekcije podataka na podprostor definiran novim značajkama budu što bliže početnim podacima u smislu euklidske udaljenosti. Drugim riječima, tehnika PCA traži najbolji linearni podprostor početnog prostora opisanog ortogonalnom osnovom novih značajki, tako da je pogreška aproksimacije podataka njihovom projekcijom na podprostoru što manja [7].

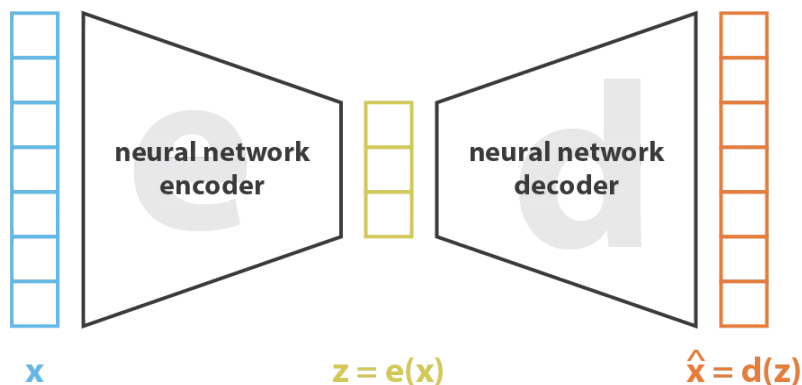
Poglavlje 2. Metodologija



Slika 2.2 Ilustracija tehnike PCA koja traži najbolji linearni podprostor koristeći linearnu algebru, preuzeto iz [7]

2.1.2 Općenito o autoenkoderima

Osnovna ideja autoenkodera sastoji se od postavljanja enkodera i dekodera kao neuronskih mreža kako bi se istrenirala najbolja shema kodiranja - dekodiranja pomoću iterativnog procesa optimizacije. U svakom koraku iteracije, arhitektura autoenkodera, gdje enkoder prethodi dekoderu, prima ulazne podatke. Enkoder zatim komprimira ulazne podatke u kompaktni oblik, koji se potom dekodira kako bi se rekonstruirali originalni podaci. U ovom procesu, generirani dekodirani izlaz uspoređuje se s originalnim podacima kako bi se kvantificirala razlika ili pogreška rekonstrukcije. Na kraju iteracije, dobivena pogreška prenosi se unatrag kroz arhitekturu autoenkodera, igrajući ključnu ulogu u prilagodbi težina neuronske mreže. Ovaj postupak ažuriranja težina omogućuje autoenkoderu da prilagodi svoje interne reprezentacije kako bi što bolje rekonstruirao originalne podatke. Ova dinamika treniranja omogućuje autoenkoderu da postupno poboljšava sposobnost rekonstrukcije tijekom iteracija, što rezultira efikasnijim predstavljanjem i obradom ulaznih podataka [1]. Cjelokupna arhitektura autoenkodera stvara usko grlo za podatke s kojim se osigurava da samo strukturirani dio informacija na kraju bude rekonstruiran.

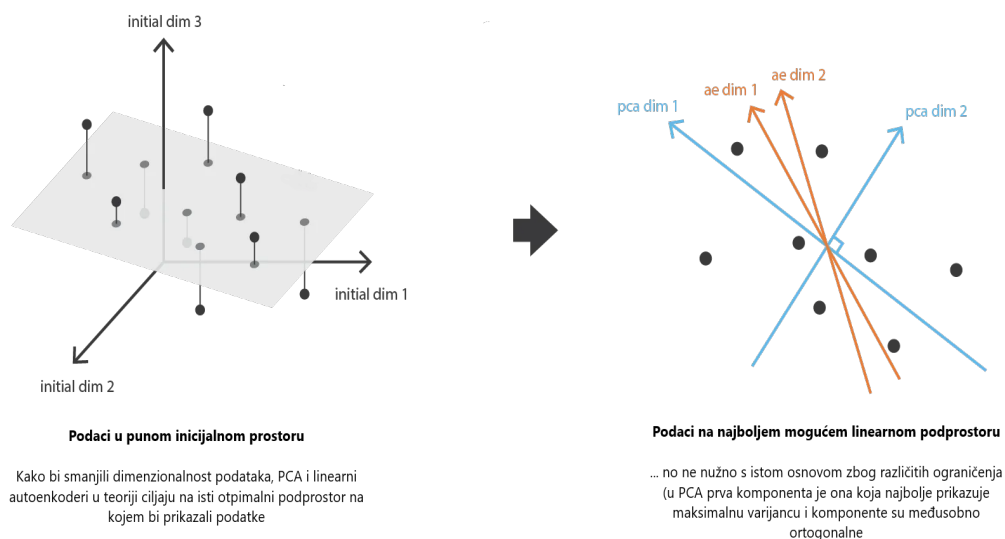


$$\text{loss} = \| \mathbf{x} - \hat{\mathbf{x}} \|^2 = \| \mathbf{x} - \mathbf{d}(\mathbf{z}) \|^2 = \| \mathbf{x} - \mathbf{d}(\mathbf{e}(\mathbf{x})) \|^2$$

Slika 2.3 Ilustracija autoenkodera s njegovom funkcijom gubitka, preuzeto iz [1]

Linearni autoenkodere u svojoj arhitekturi imaju enkoder i dekoder kao linearne transformacije koje se mogu izraziti kao matrice. U takvoj situaciji vidi se veza s tehnikom PCA koja traži najbolji linearni podprostor na koji će projicirati podatke sa što je moguće manjim gubitkom informacija. Matrice enkodiranja i dekodiranja dobivene tehnikom PCA definiraju jedno od rješenja koje se postiže gradijentnim spuštanjem. Također, nekoliko osnova za opisivanje istog optimalnog podprostora može se odabrati sa nekoliko parova enkodera i dekodera koji će isto tako rezultirati sa optimalnom pogreškom rekonstrukcije. Štoviše, za linearne autoenkodere nove značajke koje su dobivene na kraju ne moraju biti nezavisne jer nema ograničenja ortogonalnosti u neuronskim mrežama [1].

Poglavlje 2. Metodologija

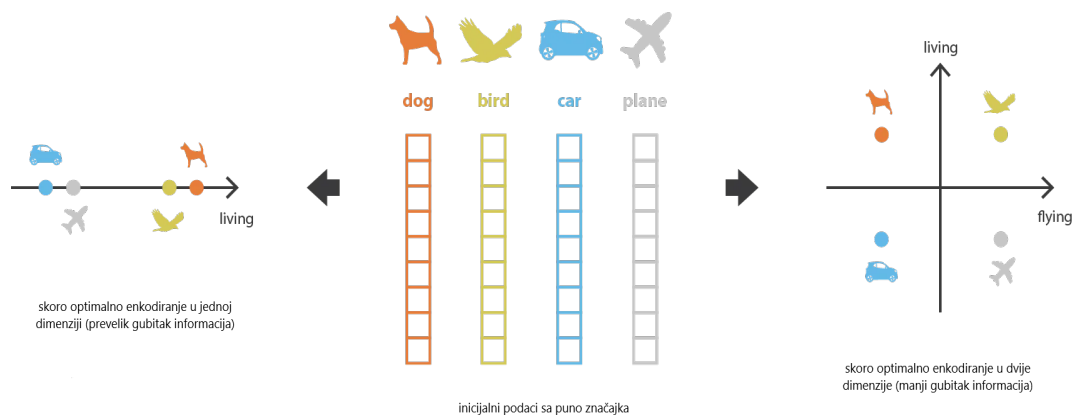


Slika 2.4 Ilustracija veze između linearnog autoenkodera i tehnike PCA, preuzeto iz [1]

Postoje dva velika izazova kod redukcije dimenzionalnosti koristeći autoenkodere [1]:

- smanjenje dimenzionalnosti bez gubitka rekonstrukcije često ima svoju cijenu: nedostatak interpretabilnih i iskoristivih struktura u latentnom prostoru, odnosno nedostatak pravilnosti i regularnosti,
- svrha smanjenja dimenzionalnosti nije samo smanjiti dimenziju skupa podataka, već smanjiti te pritom zadržati glavni dio informacija o početnoj strukturi podataka.

Zbog ovih izazova, dimenzije latentnog prostora i „dubina” autoenkodera koji definiraju stupanj i kvalitetu kompresije, postaju ključni elementi. Kao takvi, istodobno zahtijevaju precizno definiranje i prilagodbu te uzimaju u obzir konačnu svrhu primjene. Kontrola ovih parametara postaje važan faktor kako bi se postigla željena ravnoteža između učinkovite redukcije dimenzionalnosti i očuvanja informacija u latentnom prostoru. Ovaj pristup osigurava da autoenkoder bude prilagođen specifičnim zahtjevima zadatka, omogućavajući optimalnu izvedbu u različitim implementacijama.



Slika 2.5 Smanjujući dimenzionalnost, želimo zadržati strukturu koja postoji među podacima, preuzeto iz [1]

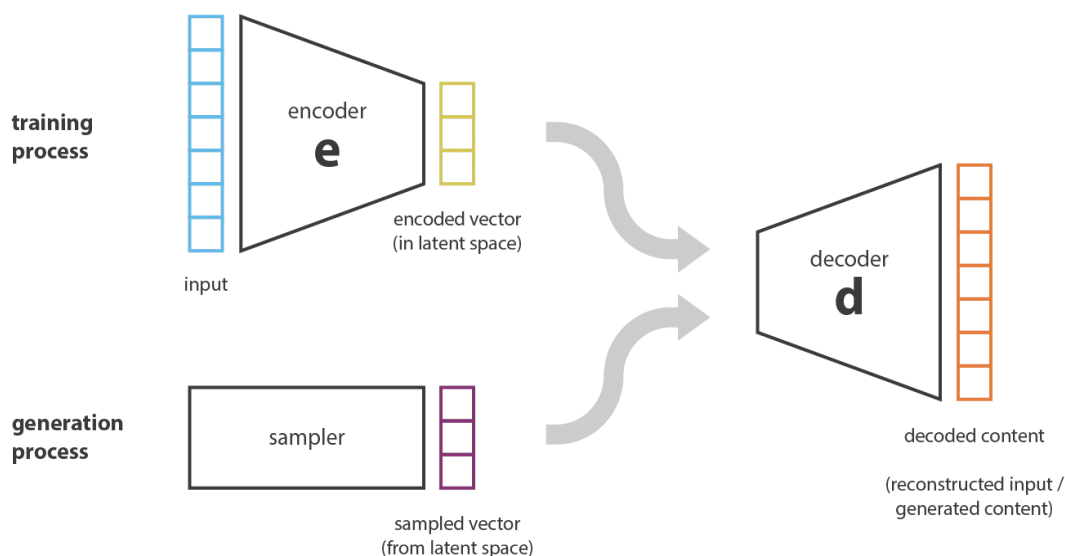
2.1.3 Autoenkoderi u svrhu generativnog procesa

Do sada je razjašnjena prvotna ideja autoenkodera koja je vezana za redukciju dimenzionalnosti te njihova arhitektura i poteškoće koje dolaze s njom. U nastavku biti će napravljena poveznica sa glavnim ciljem ovog rada: generiranjem novih sadržaja; te će se približe definiranti limitacije autoenkodera.

U trenutku kada je autoenkoder istreniran, još uvijek nije u stanju generirati novi sadržaj na konstruktivan način. Može se pretpostaviti da je latentni prostor dovoljno regularan, te nasumičnim odabirom uzeti točku iz tog prostora i dekodirati ju kako bi se dobio neki novi sadržaj. Na taj način dekodek bi se ponašao više manje kao generator generativne suparničke mreže [1].

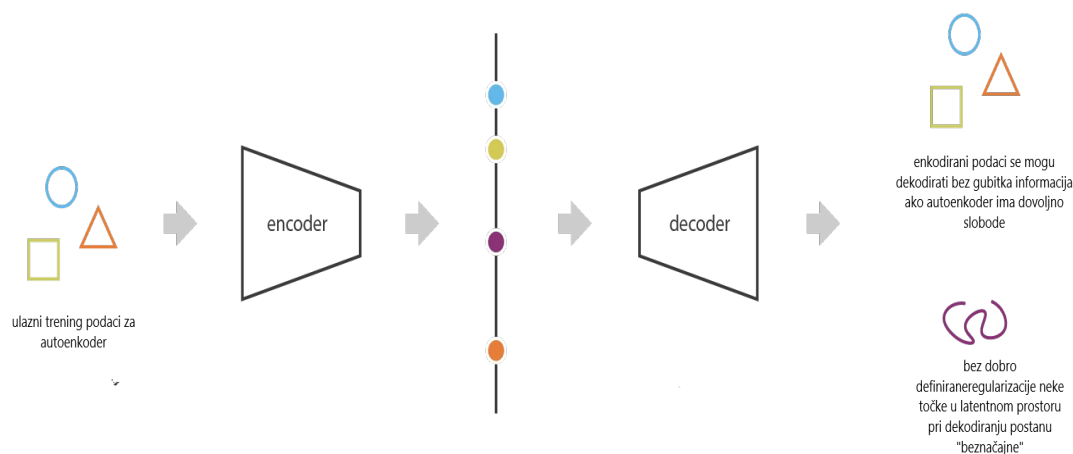
Međutim, kao što je ranije navedeno, regularnost latentnog prostora ovisi o puno faktora kao što su: distribucija podataka u početnom prostoru, dimenzija latentnog prostora, sama arhitekturi enkodera itd. Isto tako, s klasičnim autoenkoderima ne može se garantirati da će biti dobiven organizirani latentni prostor, a s tim i kompatibilan generativni proces za kreiranje novih sadržaja koristeći dekodek [1].

Poglavlje 2. Metodologija



Slika 2.6 Ilustracija principa rada dekodiranja točaka nasumično uzorkovanih iz latentnog prostora, preuzeto iz [1]

Povrh toga, kao što je već objašnjeno, glavna svrha klasičnih autoenkodera je enkodiranje i dekodiranje sa što manjim gubitkom informacija prilikom rekonstrukcije, ne uzimajući u obzir pravilnost latentnog prostora. Tako da, kada bi postojao autoenkoder dovoljno snažan da bi mogao primiti ulazne podatke veličine N i enkodirati ih na način da se svaka točka kodira kao stvarna vrijednost te dekodira bez gubitka informacija, dogodilo bi se pojačano pretreniranje. Važno je naglasiti da će neke točke latentnog prostora nakon dekodiranja sigurno davati sadržaj koji nije dovoljno zadržao strukturu i smisao originalnog podatka, čineći sadržaj takvih točaka besmislenim [1].



Slika 2.7 Prikaz lošeg latentnog prostora iz kojeg dobivamo vrijednosti bez značaja, preuzeto iz [1]

2.1.4 Varijacijski autoenkoderi

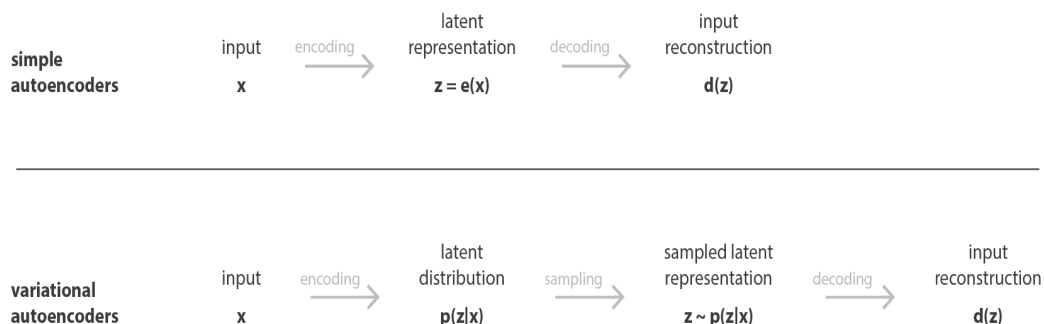
Varijacijski autoenkoderi (VAEs) mogu se definirati kao autoenkoderi čije je treniranje regulirano, kako bi se izbjeglo prekomjerno pretreniranje i osiguralo da latentni prostor ima dobra svojstva koja omogućuju smisleni generativni proces. Slično arhitekturi klasičnog autoenkodera, arhitektura varijacijskog autoenkoder sastavljena je od enkodera i dekodera te je istrenirana sa svrhom smanjenja pogreške rekonstrukcije između enkodiranih i dekodiranih podataka i početnih podataka. Međutim, kako bi se uvela regulacija latentnog prostora uvodi se modifikacija procesa enkodiranja i dekodiranja na način da se umjesto kodiranja ulaza kao jedne točke, ulaz enkodira kao distribucija preko latentnog prostora [1]. Kod VAEs model enkodera se ponekad naziva i model prepoznavanja (eng. recognition model) dok se model dekodera naziva generativni model (eng. generative model) [8].

Model varijacijskog autoenkodera trenira se po sljedećim koracima [1]:

1. ulaz se kodira kao distribucija preko latentnog prostora,
2. točka iz latentnog prostora uzorkuje se iz dobivene distribucije,
3. uzorkovana točka se dekodira te se shodno tome pogreška rekonstrukcije može

izračunati,

4. pogreška rekonstrukcije povratno se širi unazad kroz mrežu.



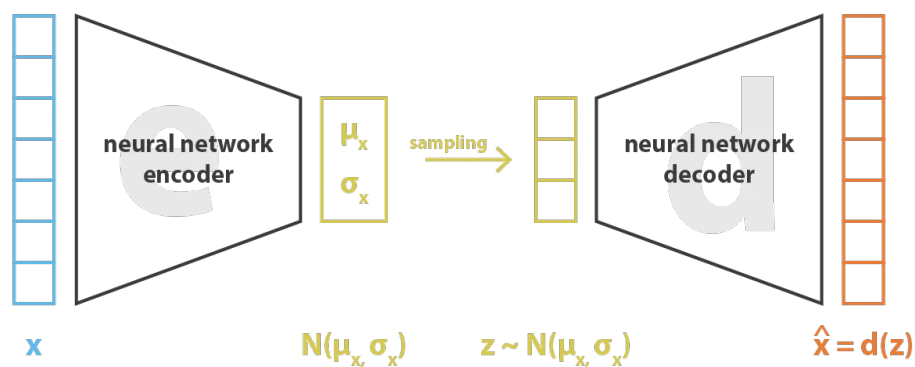
Slika 2.8 Prikaz razlike treniranja autoenkodera i VAE, preuzeto iz [1]

U praksi, enkodirane distribucije se nastoje učiniti izgledom što bliže normalnim distribucijama, tako da enkoder vraća srednju vrijednost i matricu kovarijanci koje bi definirale parametre Gaussove distribucije. Konstruiranjem modela enkodera na način da vraća niz mogućih vrijednosti (statističku normalnu distribuciju) za razliku od jedne točke, iz kojih će biti nasumično uzorkovane vrijednosti za unos u rekonstrukcijski model dekodera, u biti se dobiva kontinuirana, glatka latentna reprezentacija prostora. To bi značilo da su vrijednosti koje su u latentnom prostoru bile bliske jedna drugoj; blizu iz razloga što imaju slična svojstva, te se ta svojstva zadržavaju i nakon rekonstrukcije [8]. U ovom slučaju cilj će biti dobiti takav latentni prostor u kojem će biti moguće jasno razgraničiti latentne reprezentacije različitih skupina peptida.

Funkcija gubitka koja se želi minimizirati tijekom treniranja VAE sastoji se od "rekonstrukcijskog izraza" (na završnom sloju), koji nastoji enkodiranje i dekodiranje učiniti što učinkovitijim, sa što manjim gubitkom informacija, i "regularizacijskog izraza" (na latentnom sloju) koji nastoji regulirati organizaciju latentnog prostora čineći distribuciju dobivenu enkoderom što bližom standardnoj normalnoj distribuciji. Taj se regularizacijski pojam izražava kao Kulback-Leiblerova divergencija

Poglavlje 2. Metodologija

(eng. *Kulback-Leibler divergence (KL)*) između dobivene distribucije i standardne Gaussove, normalne distribucije.



Slika 2.9 Prikaz varijacijskog autoenkoder, preuzeto iz [1]

Izraz funkcije gubitka kod varijacijskog autoenkodera:

$$loss = \|x - \hat{x}\|^2 + KL[N(\mu_x, \sigma_x), N(0, I)] = \|x - d(z)\|^2 + KL[N(\mu_x, \sigma_x), N(0, I)] \quad (2.2)$$

2.1.5 Regularizacija

Regularizacija je oblik regresije koji ograničava, odnosno regularizira ili smanjuje procjene koeficijenata prema nuli. Drugim riječima, ova tehnika ograničava poticaj za učenjem složenijih ili fleksibilnijih modela kako bi se izbjegao rizik od pretréniranja [9]. Regularnost koja se očekuje od dobivenog latentnog prostora kako bi se omogućio kvalitetan generativni proces kod VAEs može se izraziti kroz dva glavna svojstva [1]:

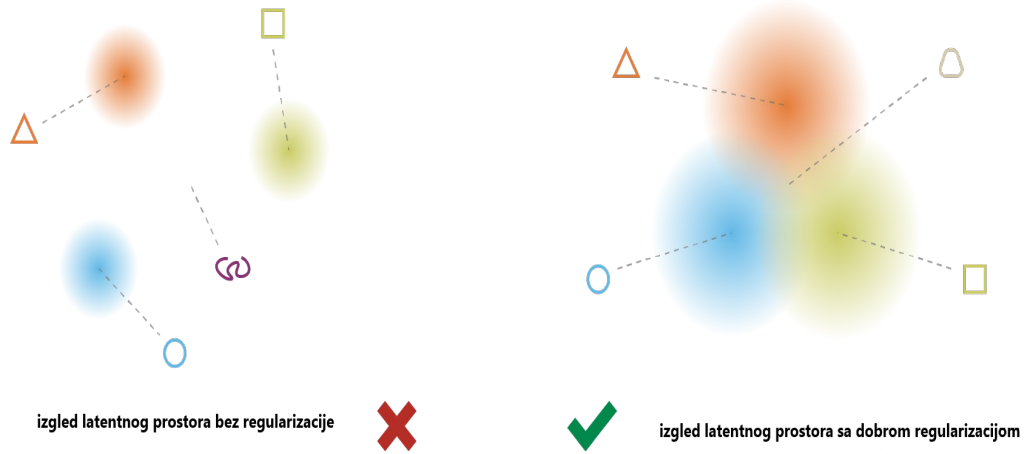
- kontinuitet (eng. continuity) - dvije bliske točke u latentnom prostoru ne bi trebale davati dva potpuno različita sadržaja nakon dekodiranja, već slična,

Poglavlje 2. Metodologija

- cjelovitost (eng. completeness) - za dobivenu distribuciju, točka uzorkovana iz latentnog prostora trebala bi dati smislen sadržaj nakon dekodiranja.

Enkodiranje ulaznog podatka kao distribuciju (kao u slučaju VAEs) nije dovoljno da se osiguraju gore navedena dva svojstva koja su prijeko potrebna za kvalitetan generativni proces. Kako bi se spriječili negativni učinci koji se mogu dobiti loše konstruiranim VAE potrebno je regularizirati i matricu kovarijanci i srednju vrijednost distribucije koju daje enkoder. U praksi se ova regularizacija provodi prisiljavanjem distribucija da budu bliske standardnoj normalnoj distribuciji. Na ovaj način zahtijeva se da matrice kovarijanci budu slične matricama identiteta, kvadratnim matricama s jedinicama na glavnoj dijagonali a ostalim vrijednostima nula, pri tom spriječavajući stvaranje točkastih (eng. *punctual*) distribucija. Također, zahtijeva se i da srednja vrijednost teži nuli, spriječavajući da kodirane distribucije budu previše udaljene jedna od druge. Ovim tehnikama kontrolira se model na način da enkodira podatke daleko jedan od drugog u latentnom prostoru i potiče se što je više moguće da se vraćene distribucije preklapaju, zadovoljavajući na ovaj način očekivane uvjete kontinuiteta i cjelovitosti. Naravno, kao i za bilo koji termin regulacije, to dolazi po cijenu veće pogreške rekonstrukcije na podacima za treniranje. Kompromis između pogreške rekonstrukcije i KL divergencije se mora namjestiti kako bi se zadovoljio zadani zadatak te dobio valjani traženi model.

Poglavlje 2. Metodologija



Slika 2.10 Prikaz razlike dobro reguliranog latentnog prostora i lošeg, preuzeto iz [1]

2.2 Rekurentne neuronske mreže

Nekolicina nedavnih istraživačkih radova koristeći se metodama dubokog učenja, za cilj je imalo poptuno iskoristiti i razumjeti podatke iz sekvencijskih baza podataka imajući ulazne podatke predstavljene sekvencijalnim nizovima. U tim radovima duboki modeli učenja sekvenci poput *Long Short-Term Memory (LSTM)* i transformatora, obučeni na skupovima podataka koji obuhvaćaju raspone sekvenci pokazali su se sposobnima učiti reprezentacije koje izvlače strukturalne i funkcionalne informacije iz istih [10]. Također, u nekoliko nedavno objavljenih primjena LSTM-ova za dizajn peptida, dobiveni su obećavajući rezultati u svrhu generiranja sekvenci aktivnih *Antimicrobial peptides (AMP)* uz eksperimentalnu potvrdu, te se samim time dokazala korisnost ove utemeljene tehnike dubokog učenja za generiranje novih AMP-ova [11].

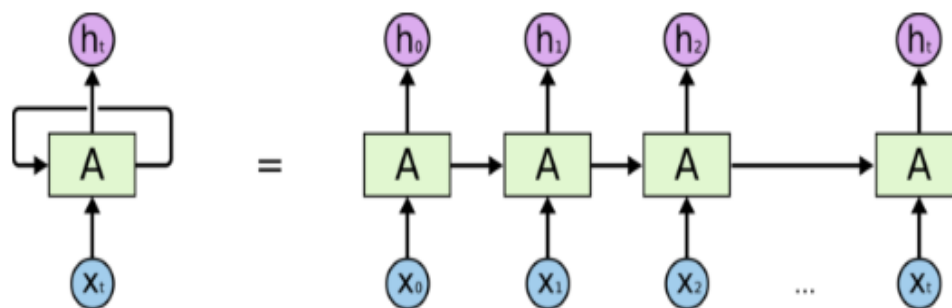
Nastavno na te radove i rezultate istih, za arhitekturu VAE modela korištenog u ovom radu kao glavni sloj u implmentaciji enkodera i dekodera korišten je *LSTM Recurrent Neural Network (RNN)*. Primjenom ovog sloja u implmentaciji model će za ulaz moći primiti visoko dimenzionalne reprezentacije sekvenci te će model, to jest dekodera modela, moći generirati nove sekvence bazirane na naučenim informacijama, zadržavajući informacije ulaznih, u kojima će poredak aminokiselina u ulaznim sekvencama peptida imati najveći utjecaj na način na koji će model učiti „razumijeti” dani skup podataka. [11].

Rekurentna (rekurzivna) neuronska mreža (RNN) generalizacija je neuronske mreže s unaprijednim prijenosom (usmjernim prijenosom u jednom smjeru) te internom memorijom. Kao takva, rekurentna je po prirodi jer izvodi istu funkciju za svaki ulaz podataka, dok izlaz iz trenutnog ulaza ovisi o izračunu prethodnog ulaza. Nakon što proizvede izlaz, kopira ga i šalje ga natrag u rekurentnu mrežu. Za donošenje odluke u svakom koraku učenja u obzir uzima trenutni ulaz te izlaz naučen iz prethodnog koraka [12].

Za razliku od neuronskih mreža s unaprijednim prijenosom, RNN-ovi mogu koristiti svoje interno stanje odnosno memoriju za obradu nizova ulaznih podataka. Korištenje memorije ih čini primjenjivima za zadatke poput prepoznavanja nesegmentirane pisane riječi ili prepoznavanja govora. Za razliku od ostalih tipova neuronskih

Poglavlje 2. Metodologija

mreža, gdje su svi ulazi neovisni jedni o drugima, u slučaju RNN neuronskih mreža svi ulazi su međusobno povezani [13].



An unrolled recurrent neural network.

Slika 2.11 Ilustracija RNN-a [13]

Prednosti rekurentnih neuronskih mreža su [13]:

- RNN može modelirati sekvence podataka na način da svaki uzorak može smatrati ovisnim o prethodnima,
- može se koristiti zajedno sa konvolucijskim slojevima kako bi se proširila učinkovitost prepoznavanja susjednih piksela.

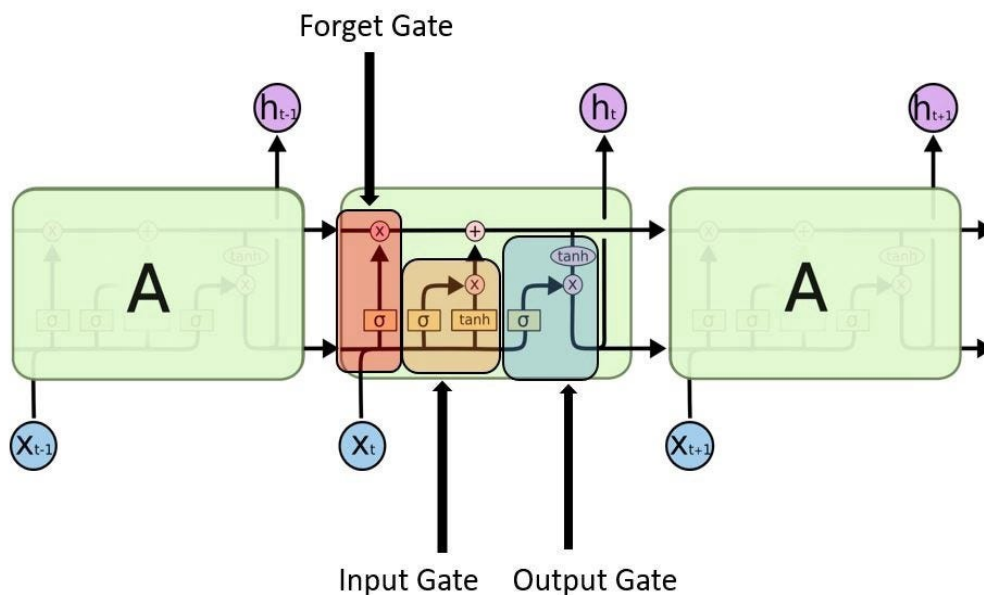
Mane rekurentnih neuronskih mreža su [13]:

- problemi s nestajanjem i eksplozivanjem gradijenta,
- uspješno treniranje RNN-a može biti vrlo zahtjevan zadatak,
- ne mogućnost obrade vrlo dugih sekvenci ako se koriste tangentne hiperboličke funkcije (*tanh*) ili *relu* kao funkcije aktivacije.

Neuronske mreže s dugoročnom i kratkoročnom memorijom (LSTM) predstavljaju modificiranu verziju rekurentnih neuronskih mreža, u kojima je olakšano pamćenje prethodno obrađenih podataka. U takvim mrežama riješen je problem nestajućeg gradijenta, karakteristične mane kod RNN mreža. Iste su pogodne za klasificiranje, obradu i predviđanje vremenskih serija s nepoznatim trajanjem vremenskih zastoja.

Poglavlje 2. Metodologija

Model LSTM-a najčešće se trenira korištenjem algoritma "širenjem unatrag" (eng. back-propagation). Prikaz pojednostavljene ilustracije modela vidljiv je na slici 2.12.



Slika 2.12 Ilustracija LSTM-a [13]

Tri su glavna vrata koja čine arhitekturu LSTM neuronske mreže [13]:

1. vrata za ulaz (eng. Input Gate) - služe za otkrivanje koji podatak iz ulaza treba koristiti za izmjenu memorije. Sigmoidna funkcija odlučuje koje vrijednosti će propustiti (0 ili 1), dok tangenta hiperbolička funkcija dodjeljuje težine vrijednostima koje su prošle kroz vrata, odlučujući o njihovoj važnosti na skali između $[-1, 1]$,
2. vrata za zaborav (eng. Forget Gate) - služe za otkrivanje koje pojednosti treba odbaciti koristeći sigmoidnu funkciju, uzorkovana točka se dekodira te se shodno tome pogreška rekonstrukcije može izračunati,
3. vrata za izlaz (eng. Output Gate) - služe za otkrivanje izlaznih podataka

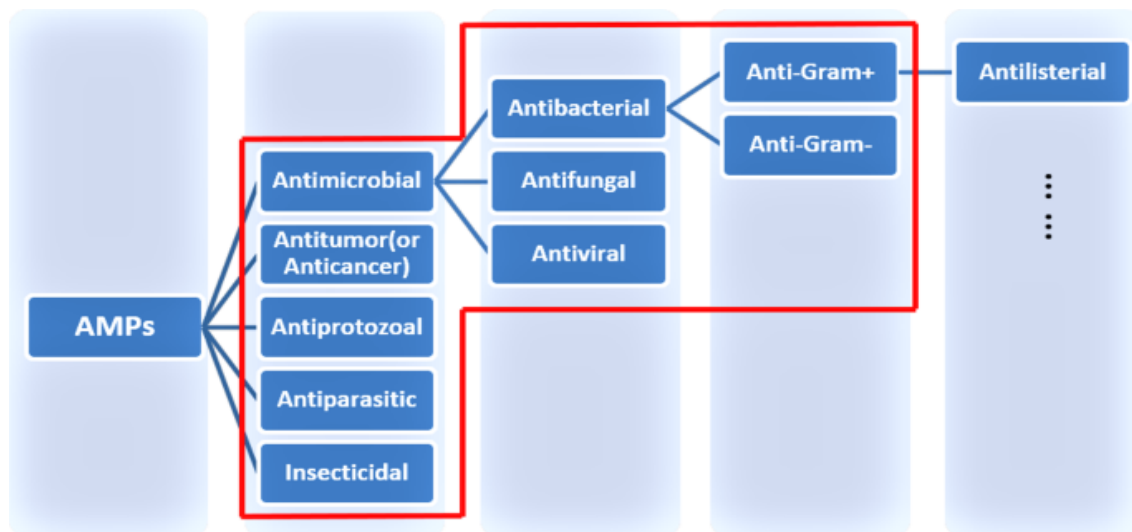
Poglavlje 2. Metodologija

služeći se ulaznim podacima i memorijom. Sigmoidna funkcija odlučuje koje vrijednosti propustiti (0 ili 1), a tangentna hiperbolička funkcija dodjeljuje težine vrijednostima koje prolaze kroz vrata, određujući njihovu važnost na skali između $[-1, 1]$ nakon čega se pomnože s izlazom sigmoidne funkcije.

2.3 Priprema i obrada podataka

U implementaciji, kao glavni izvor podataka i informacija o postojećim antimikrobnim peptidima korištena je baza podataka DRAMP 2.0 (eng. *Data Repository of Antimicrobial Peptides*). DRAMP predstavlja skup podataka otvorenog pristupa s raznolikim informacijama o antimikrobnim peptidima i njihovim karakteristikama, uključujući sekvence, strukture, svojstva, fizikalno-kemijske karakteristike, patentne podatke, kliničke informacije te izvore literature o tvrtkama koje provode istraživanja, uključujući predklinička i klinička ispitivanja [14].

DRAMP 2.0 trenutno sadrži ukupno 22,499 unosa, među kojima su 6,105 općenita antimikrobna peptida (koji uključuju prirodne i sintetske AMP-ove), 16,110 antimikrobnih peptida s patentnim informacijama te 96 antimikrobna peptida u fazi razvoja lijekova (bilo u prekliničkoj ili kliničkoj fazi) [14].

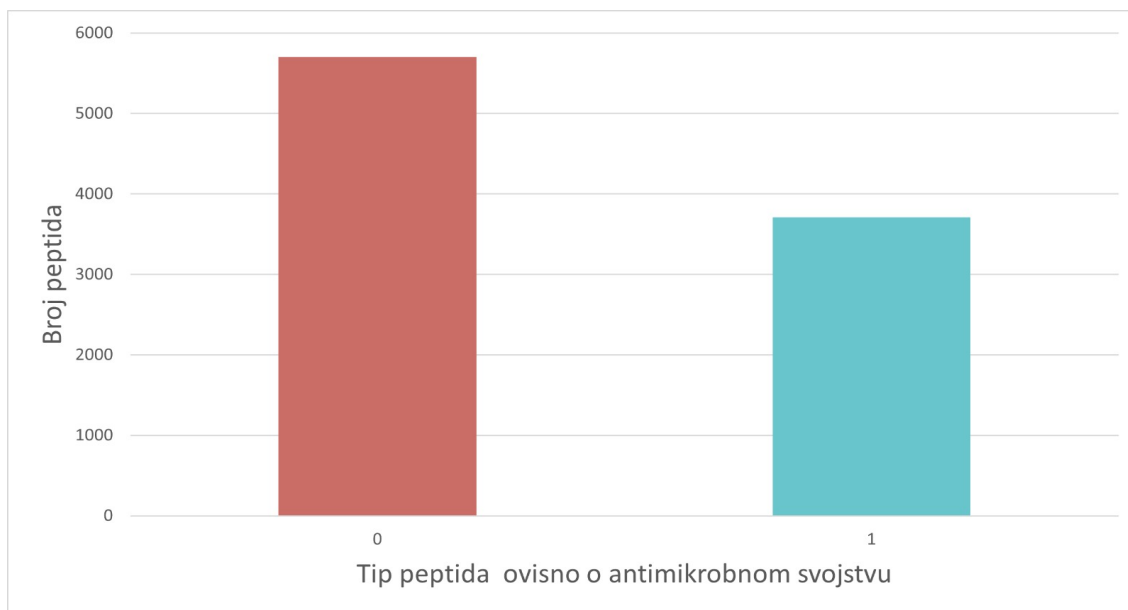


Slika 2.13 Podjela peptida po vrsti u DRAMP bazi [15]

Za početni skup podataka u ovom radu korišten je skup sastavljen od pozitivnih peptida (peptida sa antimikrobnim svojstvom) te negativnih peptida (peptida bez antimikrobnog svojstva). Peptidi sa antimikrobnim svojstvom uzeti su iz DRAMP baze dok su peptidi bez antimikrobnog svojstva preuzeti iz UniProt baze [16]. U oba slučaja, peptidi su sadržavali maksimalno pedeset aminokiselina što znači da je mak-

Poglavlje 2. Metodologija

simalna duljina peptida bila pedeset. Sa slike 2.14 vidljiva je raspodjela pozitivnih i negativnih peptida u početnom skupu, gdje su sa "1" označeni pozitivni, a sa "0" negativni peptidi. Ukupan broj peptida iznosio je 9,409 od kojih je 3,708 pozitivnih a 5,701 negativnih.



Slika 2.14 Raspodjela peptida u početnom skupu

Nadalje, gore navedeni početni skup peptida zajedno sa odgovarajućim oznakama aktivnosti podijeljen je na dva skupa: trening skup i testni skup. U ovom slučaju, 20% početnog skupa korišteno je za testiranje dok je preostalih 80% korišteno za trening. Također, kako bi se osigurala ponovljivost rezultata prilikom izvođenja ovog postupka, iskorišten je nasumični faktor prilikom podijele što pridonosi reproduktivnosti implementacije i omogućuje usporedbu rezultata kroz više iteracija.

Važno je istaknuti da će kao ulazni podaci VAE modelu biti predane samo sekvence peptida bez oznaka aktivnosti, kako bi glavni fokus prilikom treniranja bio upravo na redoslijedu i poretku aminokiselina u peptidu. Međutim, oznake aktivnosti biti će izuzetno korisne prilikom vizualizacije gdje će omogućiti prikaz peptida različitih aktivnosti u odgovarajućim bojama pružajući vizualni uvid u njihove karakteristike.

2.3.1 One-hot enkodiranje

U odluci za upotrebu *one-hot* enkodiranih ulaznih podataka (peptida) u varijacijskim autoenkoderima koji koriste LSTM sloj, ključni faktor leži u potrebi očuvanja prikaz sekvencijskih podataka, na način koji je pogodan za rad s rekurentnim neuronskim mrežama. U ovom slučaju, *one-hot* enkodiranje pruža jasno i diskretno prikazivanje različitih aminokiselina pojedinog peptida te omogućava modelu da ne pretpostavlja nikakvu unaprijed definiranu relaciju između istih. Ovim pristupom osigurava se zadržavanje informacija vezanih za ulazne sekvence koje sačinjavaju jedan peptid na kvalitetan način, olakšavajući učenje i generiranje novih pomoću LSTM sloja u okviru VAE-a.

U priloženoj tablici mogu se pronaći enkodirane reprezentacije aminokiselina koje se pojavljuju u ljudskom organizmu. Uključen je i dodatni simbol koji predstavlja prazan prostor ('#' - korišteni znak).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
A	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
#	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Tablica 2.1 Enkodirane aminokiseline koristeći *One-hot*

Poglavlje 2. Metodologija

Koristeći se ovom tablicom, peptid koji se sastoji od niza aminokiselina 'RWRW', kao takav u enkodiranom obliku izgledao bi na sljedeći način.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Tablica 2.2 Enkodirani peptid 'RWRW'

Istodobno, kako bi se smanjila vjerojatnost pojave pogrešaka te omogućilo modelu da konzistentno obrađuje ulaze, pazeći na redoslijed i uzajamne odnose između sekvenca aminokiselina predstavljenih *one-hot* enkodiranim zapisom; na svaki *one-hot* enkodirani peptid nadodan je onoliki broj praznih prostora koliko je potrebno da čine dužinu peptida jednaku pedeset. Tako se usklađivanjem dužina svih peptida na dužinu najdužeg peptida, postigla uniformnost i dosljednost ulaznih podataka.

Primjer finalnog enkodiranog peptida:

RWRW#####

Poglavlje 3

Implementacija VAE modela

Prilikom implementacije modela strojnog učenja korišten je programski jezik Python. U svrhu strojnog učenja korišten je Keras, vrlo popularna knjižnica s otvorenim kodom koja pruža praktično sučelje za izradu i treniranje neuronskih mreža. Ono što čini Keras posebnim jest njegova sposobnost apstrakcije nad složenim operacijama, omogućujući korištenje pozadinske knjižnice TensorFlow na jednostavniji i efikasniji način. S navedenim knjižnicama olakšan je sam proces implementacije varijacijskog autoenkodera u okviru ovog istraživanja. U nastavku detaljnije će biti razjašnjena implementacija zasebno enkodera i dekodera.

3.1 Enkoder

Enkoder kao sastavni dio VAE modela, za osnovni zadatak ima komprimirati ulazne podatke u prostor manje dimenzionalnosti. U ovoj implementaciji enkoder se sastoji od nekoliko ključnih slojeva za ekstrakciju informacija. Glavni sloj enkodera je sloj LSTM. Koristi se za obradu sekvenca podataka te zadržavanje i ažuriranje informacija s obzirom na prethodne vremenske korake. Ovaj sloj obrađuje ulazne podatke, omogućujući modelu naučiti i pamtit i njihove vremenske uzorke ili zavisnosti, u ovom slučaju veze između *one-hot* enkodiranih aminokiselina peptida.

Nakon LSTM sloja slijedi gusti (eng *dense*) sloj. Gusti slojevi su potpuno povezani slojevi u neuronskim mrežama koji pomažu u transformaciji i agregaciji značajki

Poglavlje 3. Implementacija VAE modela

koje su izvučene iz prethodnih slojeva. Ovaj sloj transformira izlaz iz LSTM-a sloja u vektor, koji predstavlja komprimirane značajke ulaznih podataka.

Konačno, izlazni dijelovi enkodera obuhvaćaju dvije grane: slojeve koji generiraju `z_mean` i `z_log_sigma`. Ovi slojevi su odgovorni za stvaranje parametara distribucije latentnog prostora pri čemu `z_mean` predstavlja srednju vrijednost distribucije, dok `z_log_sigma` predstavlja standardnu devijaciju iste. Ovi parametri su važni za reparametrizaciju latentnog prostora tako da omogućuju uzorkovanje iz istog, čineći model pogodnim za generiranje novih peptida. Na slici 3.1 prikazan je koncizan pregled enkodera, pružajući uvid u strukturu slojeva te oblike ulaznih i izlaznih podataka.

Treba napomenuti kako je definirano da će dimenzionalnost latentnog prostora kojeg stvara enkoder biti ograničena na dvije dimenzije. Iako je s ovakvim odabirom niske dimenzionalnosti očekivan veći gubitak informacija u usporedbi s latentnim prostorom veće dimenzionalnosti, dvije dimenzije nam olakšavaju vizualizaciju prostora koji će biti prikazan u dijelu rezultata. Također, kasnije testiranje gdje je broj dimenzija latentnog prostora povećavan a vizualizacija omogućena smanjivanjem dimenzija pomoću PCA, nije rezultiralo značajnim razlikama.

Layer (type)	Output Shape	Param #	Connected to
input_1_enc (InputLayer)	[(None, 50, 21)]	0	[]
lstm_enc (LSTM)	(None, 64)	22016	['input_1_enc[0][0]']
output_dense_enc (Dense)	(None, 32)	2080	['lstm_enc[0][0]']
dense (Dense)	(None, 2)	66	['output_dense_enc[0][0]']
dense_1 (Dense)	(None, 2)	66	['output_dense_enc[0][0]']

=====
Total params: 24,228
Trainable params: 24,228
Non-trainable params: 0

Slika 3.1 Prikaz slojeva enkodera

3.2 Dekoder

Dekoder kao komplementarna komponenta enkodera, sastavni je dio VAE modela koji koristi enkodirane informacije iz latentnog prostora kako bi rekonstruirao ulazne podatke. U ovoj implementaciji, dekodeer se sastoji od različitih slojeva za obradu i generiranje rekonstrukcije.

Prvi sloj dekodera služi kao ulaz vektora iz latentnog prostora. Ovaj vektor, u općenitoj uporabi, predstavlja komprimirane značajke koje su prethodno generirane pomoću enkodera. U ovoj implementaciji, takav vektor nije samo statički izlaz iz enkodera, već dinamički. Korištena je funkcija *sampling* kako bi se vektor dobio dinamičkim uzorkovanjem iz distribucije latentnog prostora, koristeći se izlaznim vrijednostima enkodera `z_mean` i `z_log_sigma`. Kreiranjem ulaznog vektora dekodera, dinamičkim uzorkovanjem iz distribucije latentnog prostora, potiče se model na stvaranje raznolikih primjera i generiranje rekonstrukcija koje će varirati o stvarnoj strukturi podataka, omogućivši na taj način dekodeeru generativnu svrhu.

Sljedeći sloj unutar dekodera je LSTM sloj, odgovoran za obradu uzoraka s ciljem generiranja sekvenca podataka koja će služiti kao rekonstrukcija prvotnih ulaznih podataka. Ovaj sloj omogućuje dekodeeru da pravilno rekonstruira podatke iz uzorkovanog latentnog prostora, čime se postiže kvalitetna rekonstrukcija prilikom generiranja novih primjera.

Na samom izlaznom sloju dekodeer koristi sloj za aktivaciju za koji je u ovom slučaju odabran softmax. Softmax sloj koristi se za transformiranje izlaznih vrijednosti kako bi se skalirale i kalibrirale prema odgovarajućem rasponu. U ovom slučaju, softmax se koristi u svrhu otvrđivanja (eng. *hardening*) na način da se povećava diskriminacija između vrijednosti u vektoru, s ciljem da izlazni vektor izgleda što sličnije ulaznom *one-hot* enkodiranom vektoru. Najveća vrijednost u vektoru približiti će se 1, dok će se ostale vrijednosti približiti 0, a suma svih vrijednosti u izlaznom vektoru biti će 1. Na slici 3.2 prikazan je koncizan pregled dekodera, pružajući uvid u strukturu slojeva, oblike ulaznih podataka te podataka koji se generiraju kao izlazni.

Poglavlje 3. Implementacija VAE modela

Layer (type)	Output Shape	Param #
input_1_dec (InputLayer)	[(None, 2)]	0
repeat_vector (RepeatVector)	(None, 50, 2)	0
lstm_dec (LSTM)	(None, 50, 64)	17152
time_distributed (TimeDistributed)	(None, 50, 21)	1365
activation (Activation)	(None, 50, 21)	0

=====
Total params: 18,517
Trainable params: 18,517
Non-trainable params: 0

Slika 3.2 Prikaz slojeva dekodera

3.3 VAE model

Varijacijski autoenkoder (VAE) model je strojnog učenja koji kombinira enkoder i dekodeer kako bi ostvario komprimiranje i rekonstrukciju podataka. Prilikom instanciranja VAE klase, predane su mu instance enkodera i dekodera koje su definirane kroz zasebne funkcije. U samoj VAE klasi, definirana je i gore navedena *sampling* funkcija; koja pretvara izlazne podatke enkodera u ulazne podatke dekodera dinamičkim uzorkovanjem iz distribucije latentnog prostora, povezujući enkoder i dekodeer te čineći implementaciju VAE modela dovršenim.

3.3.1 Funkcija gubitka

U ovoj VAE implementaciji, koristi se modificirana funkcija gubitka (eng. *loss function*) za evaluaciju performansi modela. Konkretno, model koristi dvije glavne komponente gubitka: gubitak rekonstrukcije i KL (Kullback-Leibler) gubitak.

Gubitak rekonstrukcije mjeri koliko dobro model rekonstruira ulazne podatke iz

Poglavlje 3. Implementacija VAE modela

latentnog prostora. Iako se često ovaj gubitak izračunava kao srednja kvadratna greška (eng. *Mean squared error (MSE)*) između izvornih ulaznih podataka i rekonstruiranih podataka sumiranih preko odgovarajućih osi, u ovom slučaju iskorišten je *Binary Crossentropy* koji se koristi kao alternativa kada su podaci binarni. Minimizacijom ovog gubitka postiže se preciznija rekonstrukcija.

Druga komponenta gubitka je KL gubitak. Ovaj gubitak mjeri koliko dobro distribucija latentnog prostora modela odgovara standardnoj normalnoj distribuciji. KL gubitak potiče model da prilagodi distribuciju latentnog prostora kako bi bila što sličnija standardnoj normalnoj distribuciji.

U konačnici, ukupan gubitak modela sastoji se od kombinacije gubitka rekonstrukcije i KL gubitka. Cilj minimizaciju ukupnog gubitka tijekom treninga putem gradijentnog spusta kako bi se postigla što bolja rekonstrukcija podataka i odgovarajuća distribucija latentnog prostora. U analizi rezultata biti će prikazano ponašanje oba dva gubitka kroz epohe treniranja te njihove početne i finalne vrijednosti.

3.3.2 Treniranje modela

Za treniranje modela odabrano je 100 epoha, gdje jedna epoha predstavlja jedan potpuni prolazak kroz skup podataka. Kroz proces treniranja, model obrađuje podatke u grupama, pri čemu je za svaku grupu odabrana veličina od 128 uzoraka (eng. *batch size*). Sa jednom grupom od 128 uzoraka predstavljen je broj uzoraka koji će se procesirati tijekom svake epohe. Ovaj broj može se prilagoditi prema specifičnim potrebama i resursima. Veći *batch size* može ubrzati trening, dok manji može pružiti stabilniju optimizaciju.

Tijekom treninga korišten je i funkcija povratnog poziva za rano zaustavljanje (eng. *EarlyStopping*). Ova funkcija omogućava rano zaustavljanje treninga ako se zadovolje određeni kriteriji. U ovom slučaju, postavljeni prag za minimalnu promjenu gubitka je 0.01 dok je vrijeme čekanja postavljeno na 5. To znači da će se treniranje zaustaviti ako se ukupna vrijednost funkcije gubitka ne smanji za više od 0.01 tijekom perioda od 5 epoha. Ovo pomaže u sprječavanju pretreniranosti modela te optimizira vrijeme tijekom treninga. Također, omogućen je povratak na najbolje težine modela ako se trening zaustavi prije nego što se postignu optimalni rezultati.

Poglavlje 4

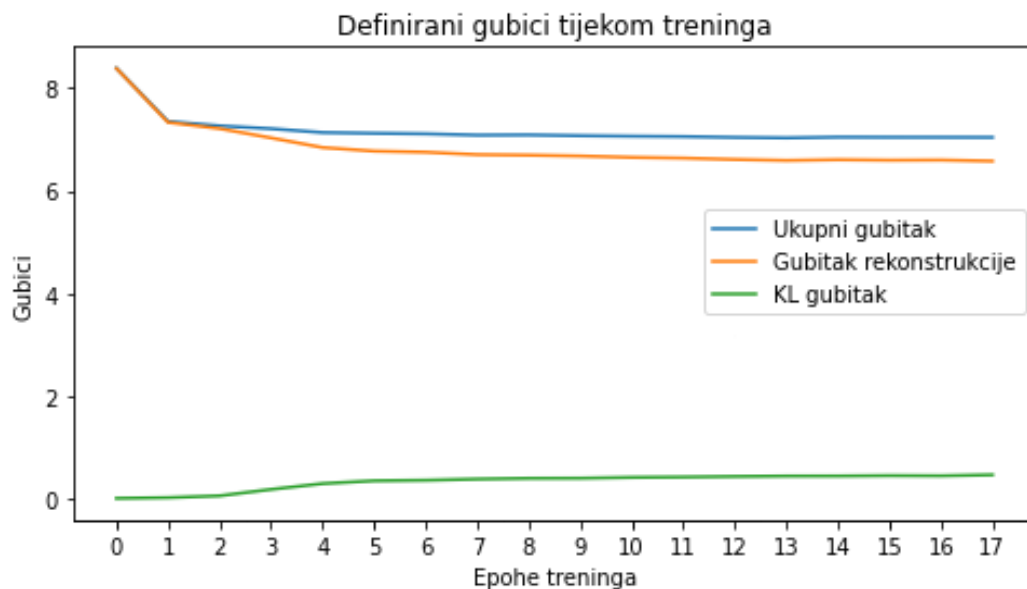
Rezultati

Uspješnost generiranja novih pepetida, kao glavna metrika ovog istraživanja, biti će predstavljena kroz analizu rezultata podijeljenu u tri dijela: dijela vezanog za vrednovanje istreniranog VAE modela, dijela vezanog za vrednovanje latentnog prostora te na dio vezan za analizu dobivenih peptida. Sva testiranja provedena su na način da su odvojeno vrednovani rezultati na skupovima podataka za treniranje i testiranje.

Nakon završetka treninga VAE modela, koji je implementiran primjenom prethodno opisanih tehnika, kao rezultat dobiven je model s određenim karakteristikama. Vrijednosti gubitka ovog modela (ukupni gubitak, gubitak rekonstrukcije i KL gubitak) mogu se detaljnije razmotriti pomoću tablice 4.1 i grafičkog prikaza 4.1.

	Ukupni gubitci	Gubitci rekonstrukcije	Gubitci KL
Početni gubitak treninga	9,050	8,386	5×10^{-3}
Završni gubitak treninga	7,003	6,579	0,463
Početni gubitak validacije	1,893	1,892	7.289×10^{-4}
Završni gubitak validacije	1,614	1,638	0,433

Tablica 4.1 Vrijednosti gubitaka istreniranog VAE modela



Slika 4.1 Prikaz gubitaka tijekom treninga

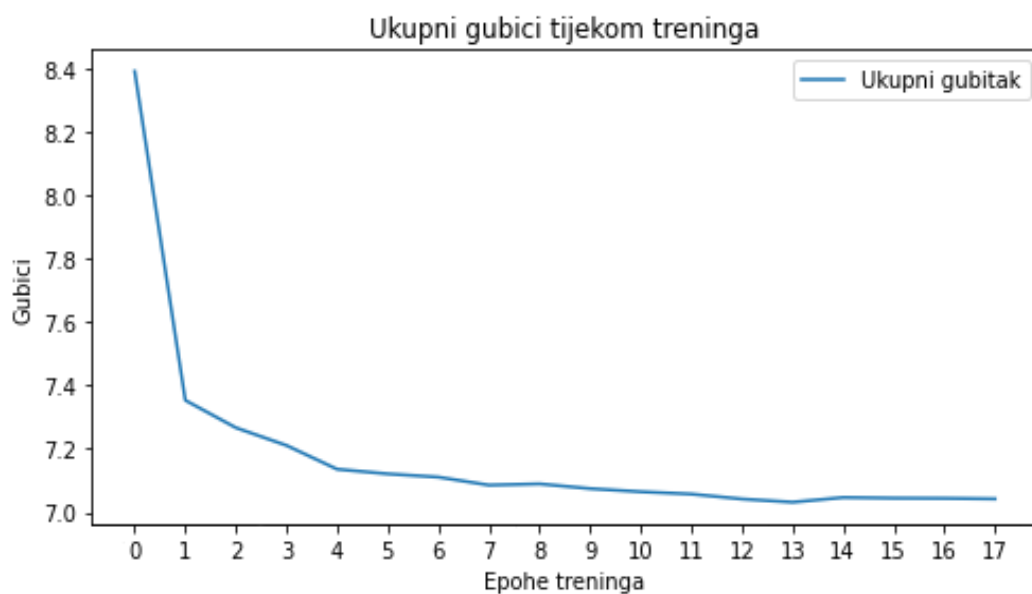
Prema podacima iz tablice 4.1 i slike 4.1 može se uočiti da su gubici rekonstrukcije i KL gubici neuravnoteženi. Početni gubitak rekonstrukcije kod treninga podataka iznosi 9,050 dok početni KL gubitak kod trening podataka iznosi 0,005. Istodobno, ukupna vrijednost gubitka rekonstrukcije kroz treniranje modela ukupno se smanjila za 2,047 dok je u slučaju KL gubitka porasla za 0,458. Dok vrijednost gubitka rekonstrukcije linearno s vremenom opada, vrijednost KL gubitka (koja je pri početku treniranja puno manja od vrijednosti gubitka rekonstrukcije) u blagom je porastu, nakon čega gotovo stagnira. Iako bi bilo optimalno imati i gubitak rekonstrukcije i KL gubitak u blagom opadadnju kroz epohe treninga te u rasponima sličnih vrijednosti (s čime bi osigurali i dobru rekonstrukciju podataka i dobra svojstva latentnog prostora) to je vrlo teško dostižno u praksi.

U pokušaju da se uravnoteži odnos među gubitcima, kroz testiranja uveden je i β faktor s kojim se pokušalo množiti KL gubitak kako bi umanjili njegov značaj te kako bi model počeo smanjivati ukupni gubitak smanjivanjem gubitka rekonstrukcije, umjesto KL gubitkom. Također, pokušana je provedba zagrijavanja (eng. *Warm-up*) KL gubitka na način da se postepeno povećava β faktor ovisno o epohi treninga

Poglavlje 4. Rezultati

u kojoj se model nalazi, ali jednako kao i s konstantnim β faktorom, nije dobiven očekivan rezultat.

Može se pretpostaviti da su ulazni podaci (*one-hot* enkodirani format), a i sam model (implementiran na gore opisan način) takvog tipa da je u fazi treniranja bilo puno lakše smanjivati ukupne gubitke smanjivanjem KL gubitka, za što se onda model i odlučuje. U takvom slučaju može se očekivati bolja kvaliteta regularnosti latentnog prostora, ali i smanjena kvaliteta rekonstrukcije novih peptida. Na slici 4.2 detaljnije je prikazan pad ukupnog gubitka kroz epohe treninga.



Slika 4.2 Prikaz ukupnog gubitka kroz trening

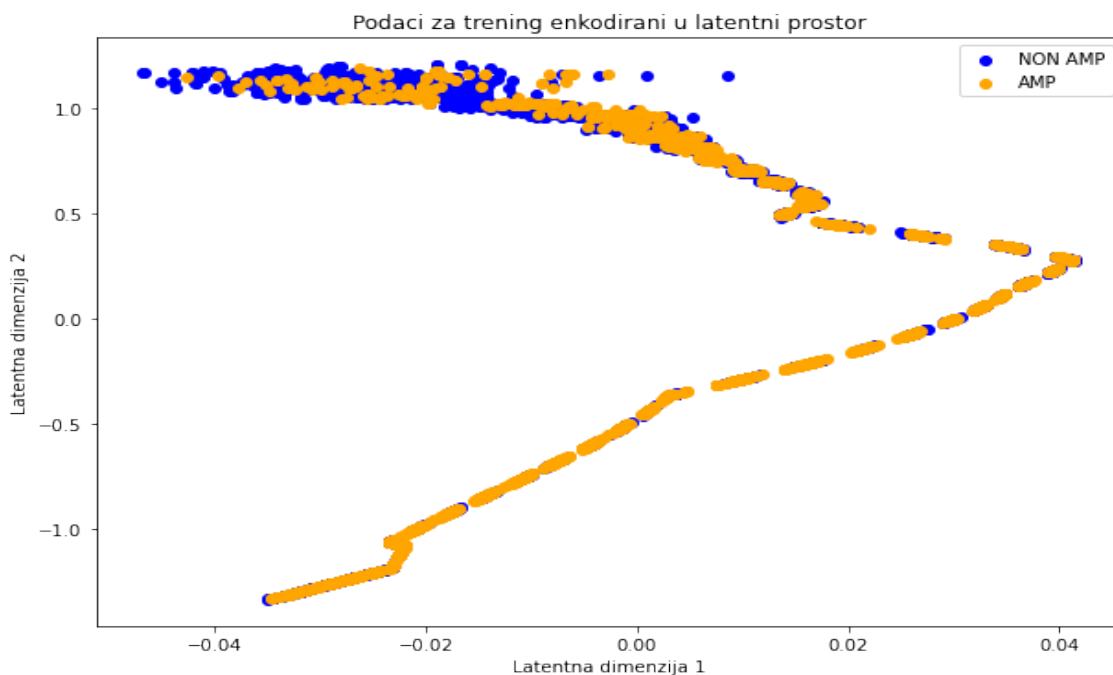
Kako je u implementaciji enkodera navedeno, izabran je dvodimenzionalan latentni prostor s kojim je olakšana vizualizacija te istodobno razumijevanje samog prostora. Kao što je vidljivo sa slika u nastavku, peptidi su prikazani točkama te obojeni prema njihovoj oznaci aktivnosti. Pozitivni peptidi (sa antimikrobnim svojstvom) označeni su žutom bojom (u legendi označeni kao "AMP") dok su negativni peptidi (bez antimikrobnog svojstva) označeni plavom bojom (u legendi označeni kao "NON AMP").

Na slikama 4.3 i 4.4 prikazani su ulazni podaci za trening i testiranje enkodirani

Poglavlje 4. Rezultati

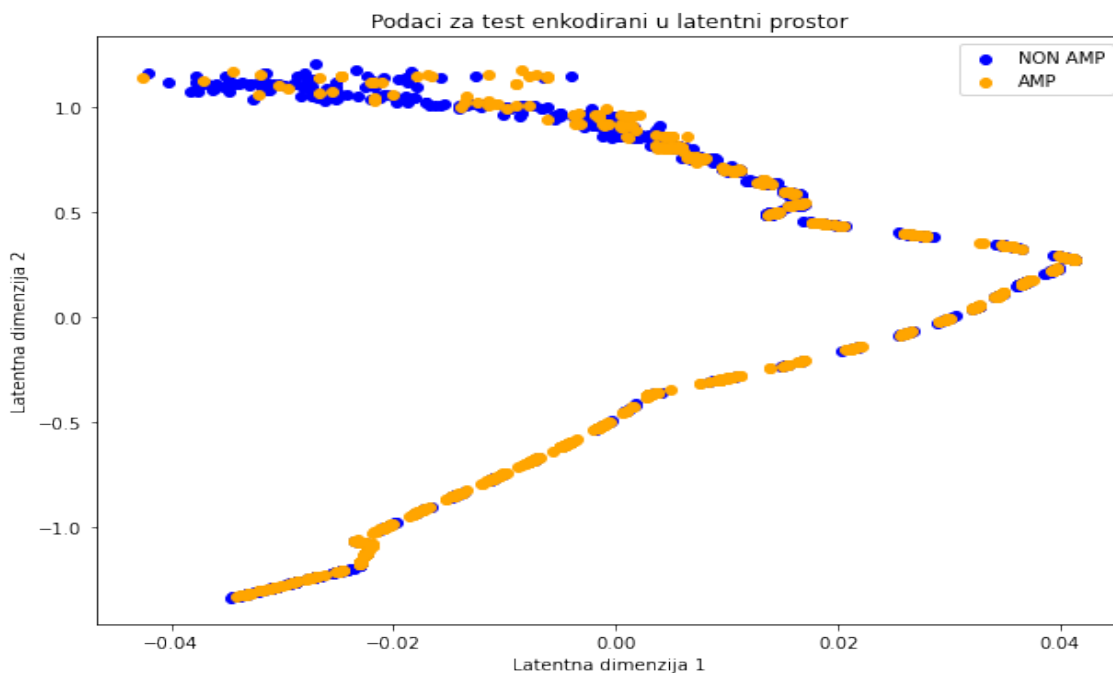
u latentni prostor. Vidljivo je da je uvjet potpunosti latentnog prostor zadovoljen s obzirom na preklapanje distribucija, te da gotovo nema neiskorištenog prostora u mjestima di je gustoća točaka najveća. Može se pretpostaviti da bi uzorkovanjem a potom dekodiranjem iz takvog prostora dobiveni peptid bio smislen te bi zadržavao karakteristike distribucija toga prostora.

Isto tako, možemo uočiti kako postoje preklapanja među peptidima različite aktivnosti nad istim prostorom. Iz tog razloga postoji sumnja da uvjet kontinuiteta latentnog prostora nije potpuno ispunjen, te da bi uzorkovanje a potom dekodiranje iz takvog prostora moglo rezultirati peptidom čije karakteristike ne bi jasno mogle ukazivati na karakteristike distribucija tog prostora. Uz to može se pretpostaviti da ulazne značajke peptida prema vrsti aktivnosti nisu bile dovoljno različite kako bi ih VAE model s preciznošću mogao razlučiti uzimajući u obzir samo redoslijed i poredak aminokiselina u peptidu.



Slika 4.3 Trening podaci enkodirani u latentni prostor

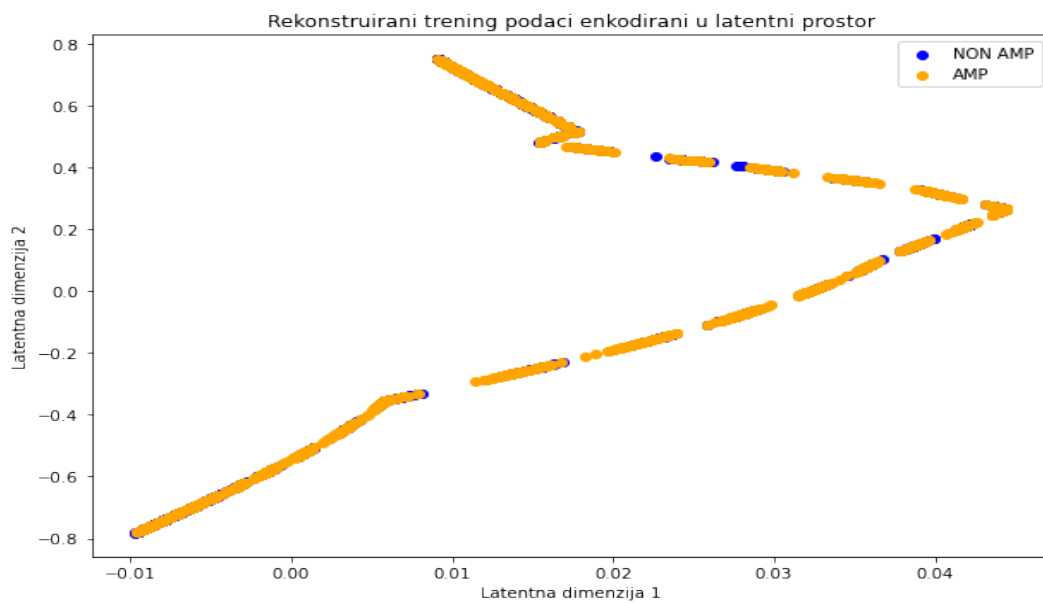
Poglavlje 4. Rezultati



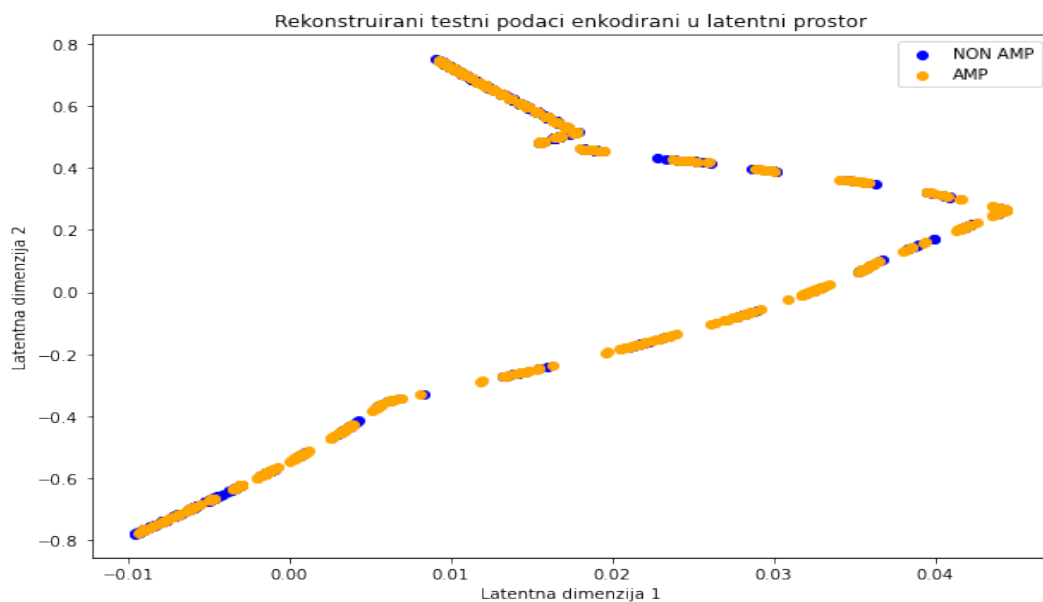
Slika 4.4 Testni podaci enkodirani u latentni prostor

Na slikama 4.5 i 4.6 prikazani su rekonstruirani podaci za trening i testiranje ponovo enkodirani u latentni prostor. Vidljivo je da prilikom dekodiranja i ponovnog enkodiranja podataka model smješta vrijednosti u isti dio latentnog prostora gdje su se nalazili prvobitno enkodirani ulazni podaci. Na taj način potvrđuje se pretpostavka da je model kroz treniranje bio više priklonjen stvaranju kontinuiranog latentnog prostora, nego zadržavanju kvalitetne rekonstrukcije ulaznih podataka.

Poglavlje 4. Rezultati



Slika 4.5 Rekonstruirani trening podaci enkodirani u latentni prostor



Slika 4.6 Rekonstruirani testni podaci enkodirani u latentni prostor

4.1 Analiza dekodiranih peptida

Prije analize uspješnosti rezultata, nekoliko ključnih aspekata biti će istaknuto radi boljeg razumijevanja. Zbog gore opisanog načina na koji je VAE model implementiran te zbog dinamičkog uzorkovanja (eng. *sampling*) vektora koji ulazi u dekođer, ali i karakteristika istreniranog modela već samo dekodiranje trening i testnog skupa ulaznih podataka rezultira novim peptidima. Kod optimalno istreniranog modela dekodiranje ulaznog trening skupa rezultira rekonstruiranim podacima identičnim onim ulaznima.

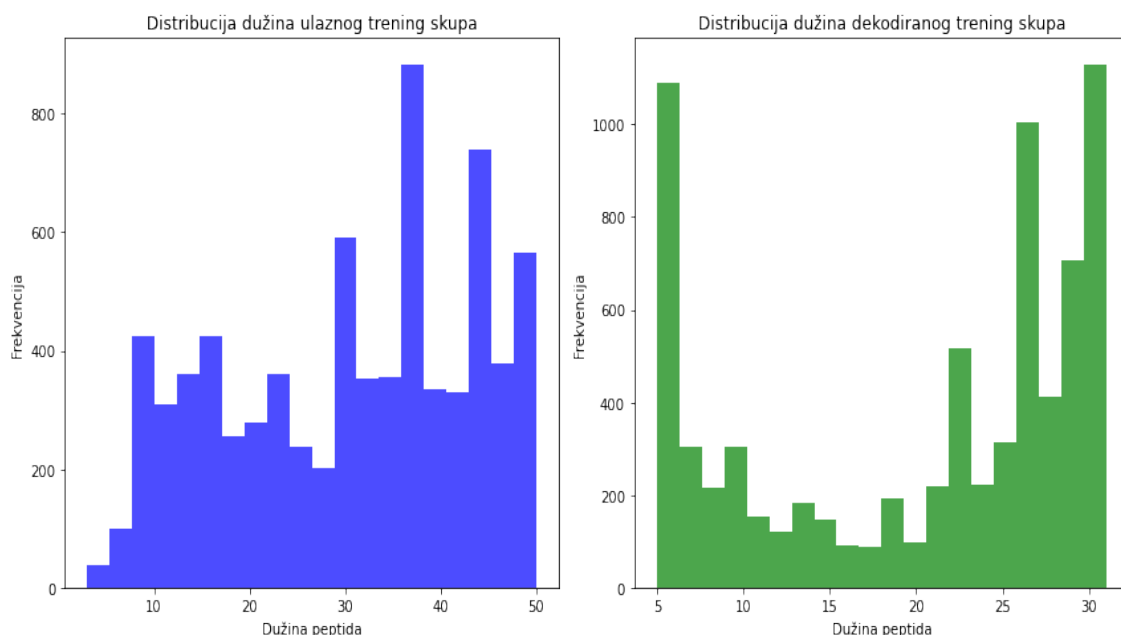
Za procjenu kvalitete novih peptida nužno je definirati odgovarajuće metrike za vrednovanje istih. Odabrane su tri metrike:

- uspoređba dužina peptida - gdje su dužine novo generiranih peptida uspoređene sa dužinama ulaznih peptida,
- uspoređba sličnosti peptida - gdje su novo generirani peptidi uspoređeni sa ulaznim peptidima,
- predviđanje oznake aktivnosti novo dobivenih peptida - koristeći nezavisni istrenirani model za klasificiranje peptidnog antimikrobnog svojstva.

Treba napomenuti, da će pri svakoj analizi prvo biti rađene uspoređbe, sličnosti i predikcije novo generiranih peptida nastalih dekodiranjem trening skupa, a potom novo generiranih peptidi nastalih dekodiranjem testnog skupa peptida.

Na slici 4.7, prikazana su dva grafa koja pokazuju razliku distribucija dužina ulaznih peptida namijenjenih za trening te novo generiranih peptida iz tog skupa. Iz grafova se može vidjeti da se općeniti oblik distribucija dužina zadržao s tim da su se prosječne vrijednosti dužina smanjile. Srednja vrijednost dužine ulaznih trening peptida iznosila je 30,47 dok je srednja vrijednost dužine novih peptida iznosila 20,19.

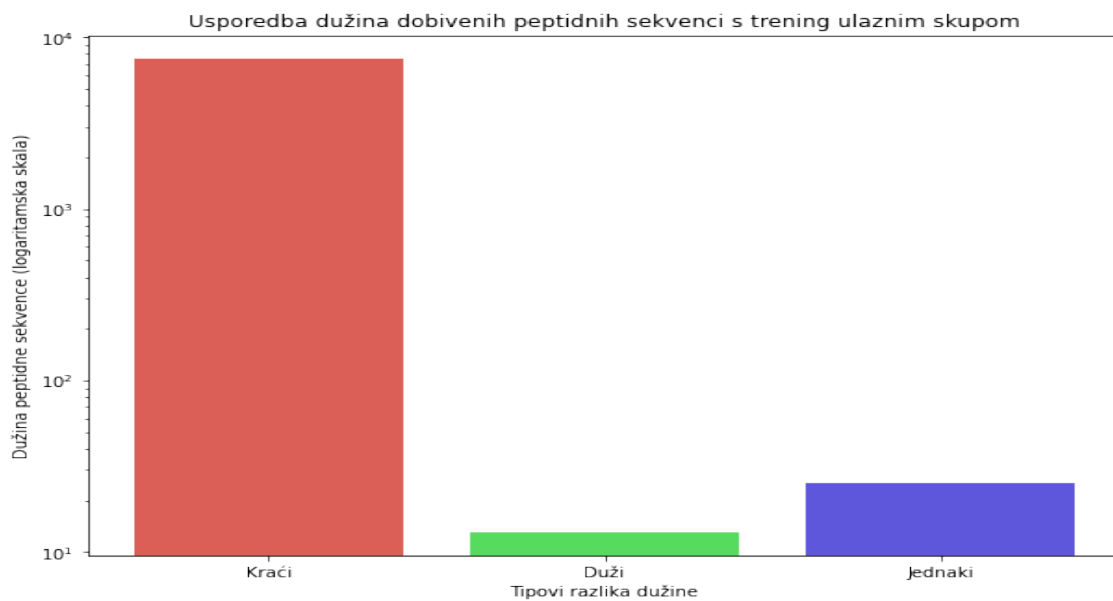
Poglavlje 4. Rezultati



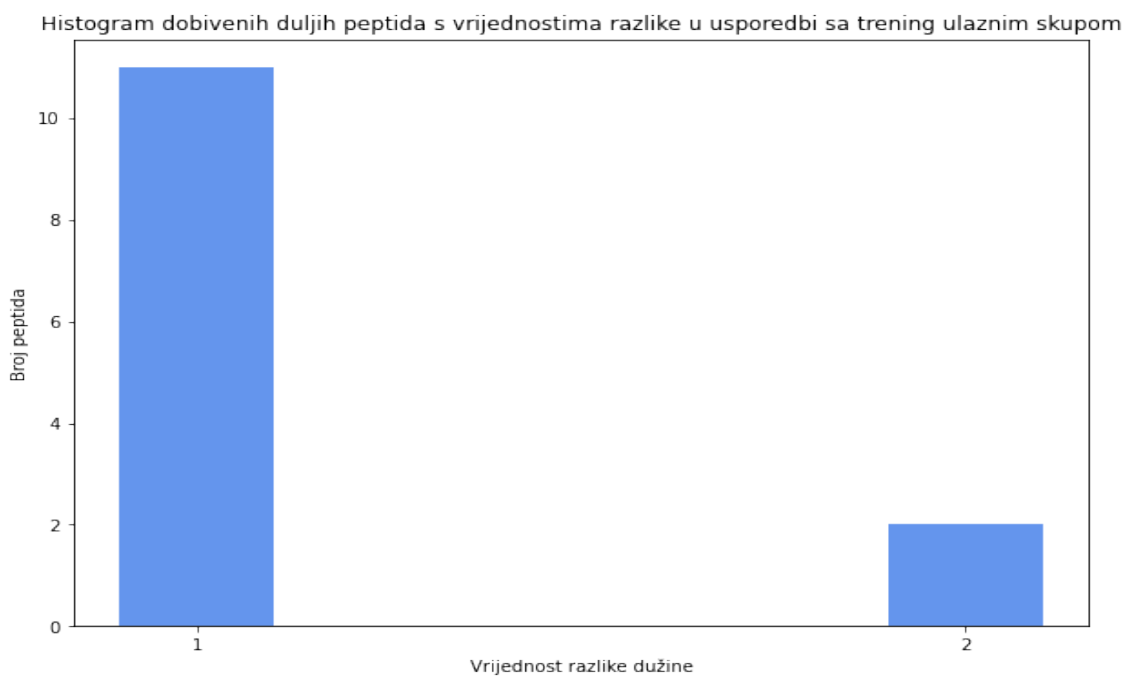
Slika 4.7 Usporedba distribucija dužina novih peptida sa trening peptidima

Na slikama 4.8, 4.9 i 4.10 vidljivi su odnosi dužina novih peptida u usporedbi sa trening skupom peptida. Dobiveno je 7,489 kraćih, 13 dužih te 25 peptida jednake dužine. Srednja vrijednost razlike dužine kod novih kraćih peptida iznosi 10,33 dok je srednja vrijednost razlike dužine kod novih dužih peptida 1,15. Detaljniji prikaz razlika dužina ovisno o tome je li novi peptid duži ili kraći vidljiv je na slikama 4.9 i 4.10. Možemo zaključiti da je velika većina novih peptida kraća, što je vrlo korisno za daljnju praktičnu sintezu takvih peptida.

Poglavlje 4. Rezultati

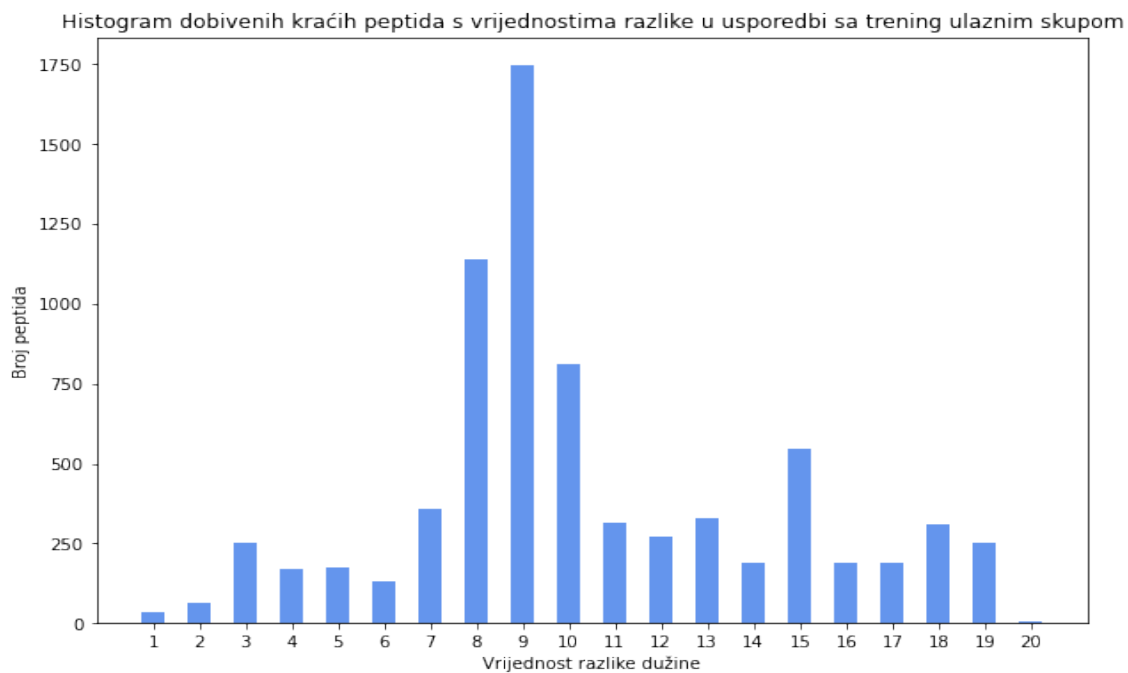


Slika 4.8 Usporedba dužina novih peptida sa trening skupom



Slika 4.9 Razlika dužina duljih novih peptida sa trening skupom

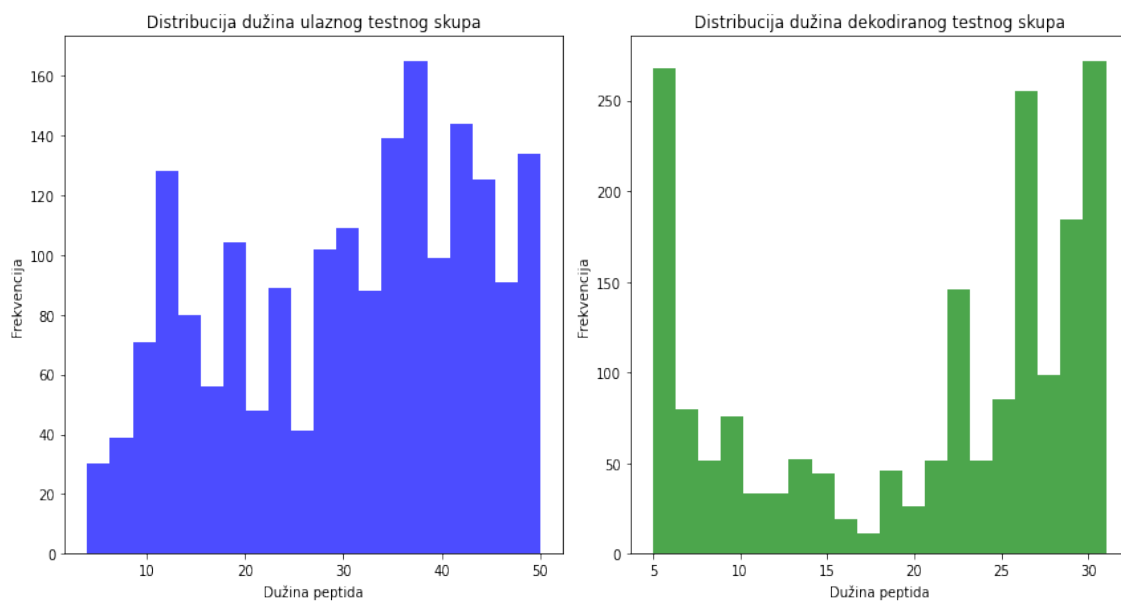
Poglavlje 4. Rezultati



Slika 4.10 Razlika dužina kraćih novih peptida sa trening skupom

Na slici 4.11, prikazana su dva grafa koja pokazuju razliku distribucija dužina peptida namijenjenih za test te novo generiranih peptida iz tog skupa. Iz grafova vidimo da se općeniti oblik distribucija dužina zadržao s time da su se prosječne vrijednosti dužina smanjile. Srednja vrijednost dužine testnih peptida iznosila je 30,48 dok je srednja vrijednost dužine novih peptida iznosila 20,24.

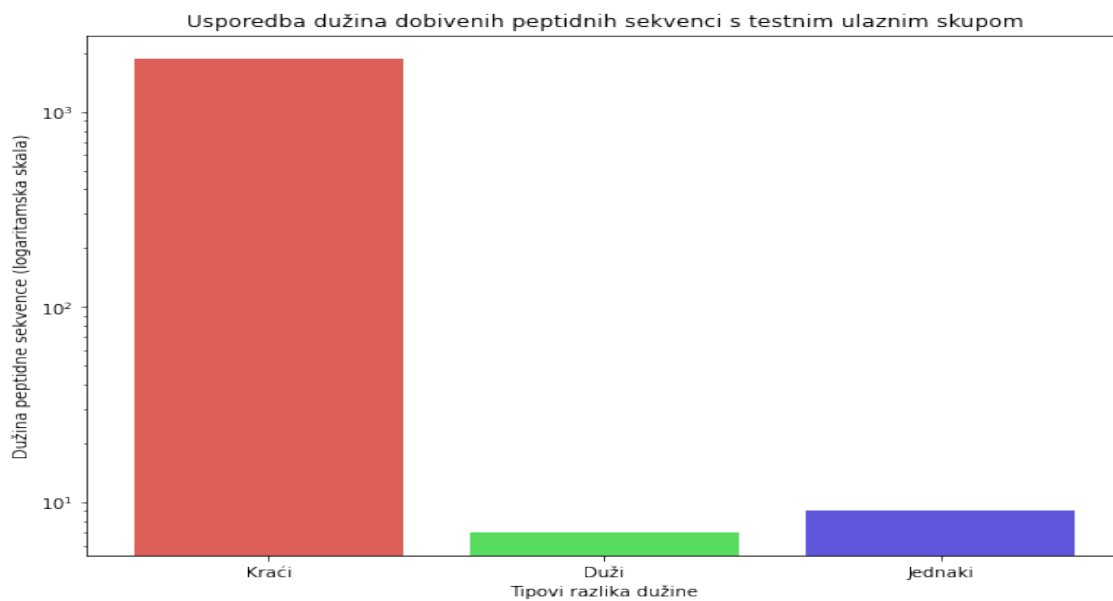
Poglavlje 4. Rezultati



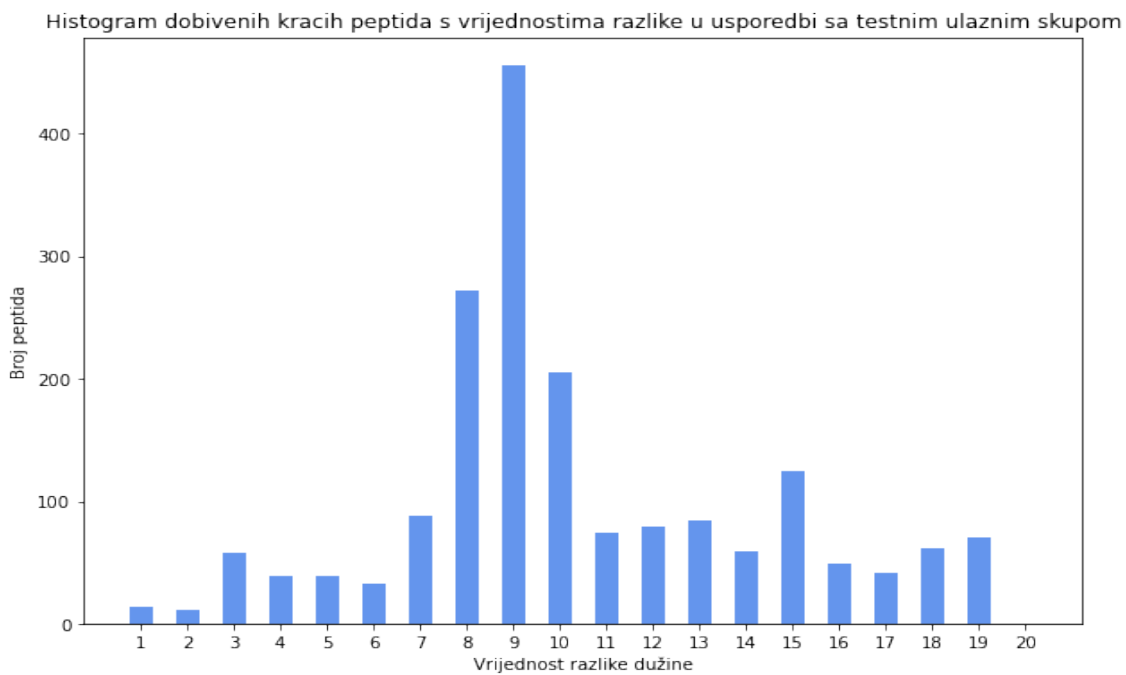
Slika 4.11 Usporedba distribucija dužina novih peptida sa testnim peptidima

Na slikama 4.12 i 4.13 vidljivi su odnosi dužina novih peptida u usporedbi sa testnim skupom peptida. U novo dobivenim peptidima dobiveno je 1,866 kraćih, 7 dužih te 9 jednakih. Srednja vrijednost razlike dužine kod novih kraćih peptida iznosi 10,33 dok je srednja vrijednost razlike dužine kod novih dužih peptida 1,0. Detaljniji prikaz razlika dužina novih peptida koji su kraći vidljiv je na slici 4.13. Razliku dužine duljih novih peptida nije potrebno prikazivati grafom kada iznosi 1. I u ovom slučaju vrijedi zaključak iz usporedbe sa trening skupom, a to je da su peptidi u velikoj većini kraći što pridonosi njihovoj daljnjoj praktičnoj sintezi.

Poglavlje 4. Rezultati



Slika 4.12 Usporedba dužina novih peptida sa testnim skupom



Slika 4.13 Razlika dužina kraćih novih peptida sa testnim skupom

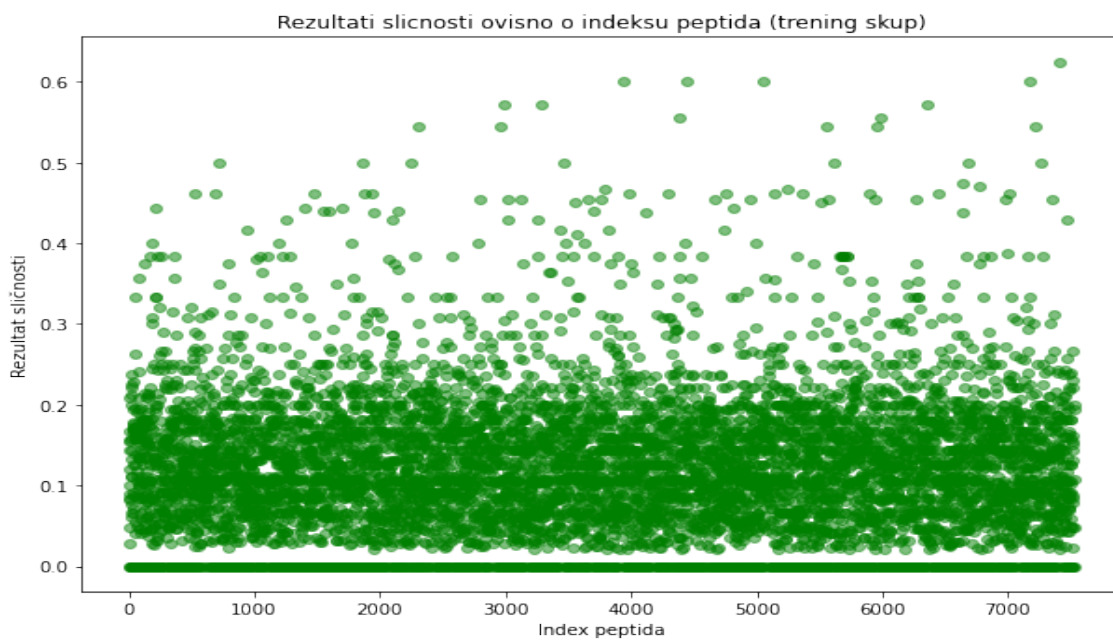
Poglavlje 4. Rezultati

Usporedba sličnosti peptida temelji se na usporedbi njihovih sekvenci, to jest niza aminokiselina. Svaka sekvenca peptida predstavljena je kao niz aminokiselina, a usporedba se provodi na način da se svaka aminokiselina u jednoj sekvenci uspoređuje s odgovarajućom aminokiselinom u drugoj sekvenci. Ideja iza usporedbe je pronalaženje zajedničkih dijelova (podudaranja) te različitih dijelova (nepodudaranja) između dva peptida kako bi procijenili njihovu sličnost.

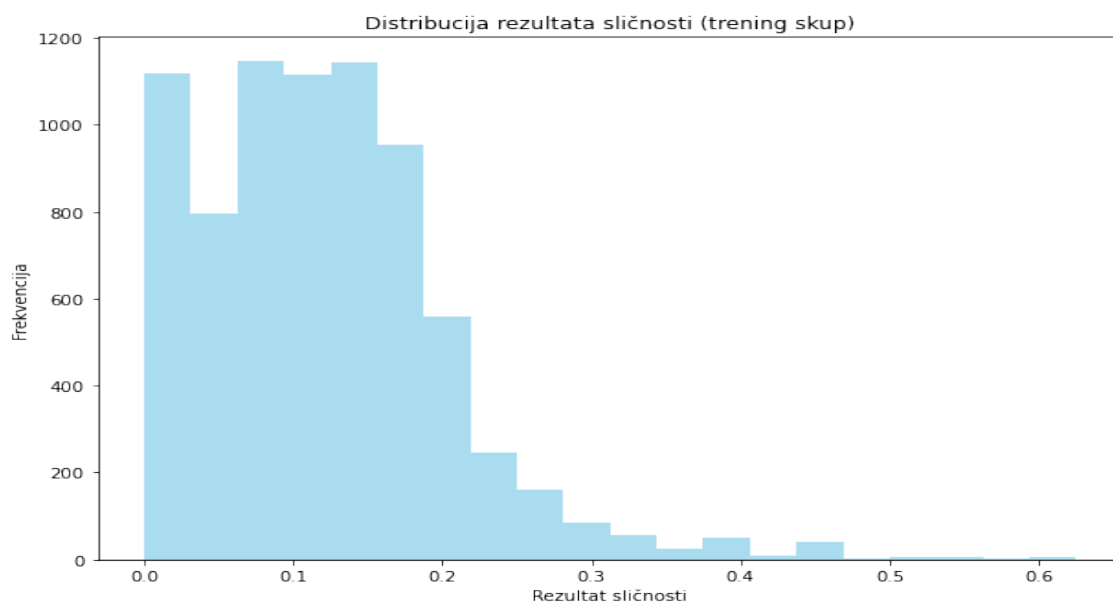
Pronalaženje sličnosti između dva peptida omogućeno je usporedbom koju funkcija radi globalnim poravnanjem peptida. Rezultat ovog poravnanja sadrži podudarajuće dijelove sekvenci, bodove za ta podudaranja, te početnu i krajnju poziciju podudaranja. Ukupna sličnost između peptida izračunava se pomoću omjera podudaranja i dužine dulje sekvence. Konačni rezultat usporedbe prikazan je numeričkim indeksom sličnosti, gdje što je vrijednost bliža 1 ukazuje na veću sličnost između peptida, dok što je vrijednost bliža 0 ukazuje na manju sličnost.

Iz slika u nastavku vidljivi su rezultati i distribucije sličnosti pojedinačno trening skupa peptida sa novo generiranim peptidima te testnog skupa peptida sa novo generiranim peptidima. Srednja vrijednost sličnosti trening skupa peptida s njihovim odgovarajućim novo generiranim peptidima iznosi 0,13 dok sličnost testnog skupa s odgovarajućim novo generiranim peptidima iznosi 0,03. Može se očitati kako vrijednost u oba slučaja varira između 0 i 0,4, iz čega se može zaključiti da sličnost na bazi podudaranja sekvenci nije velika. No to i dalje ne mora značiti da novo generirani peptidi nisu zadržali sličnosti prema njihovim fizikalno-kemijskim značajkama, to jest u ovom slučaju prema antimikrobnom svojstvu.

Poglavlje 4. Rezultati

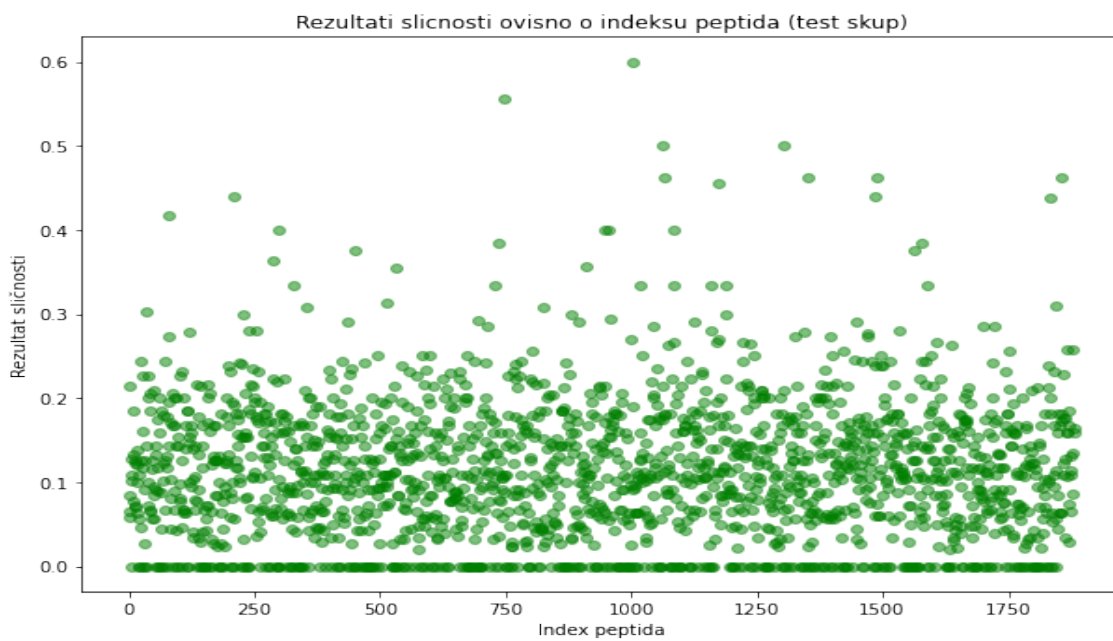


Slika 4.14 Rezultati sličnosti ovisno o indeksu - novi peptidi s trening peptidima

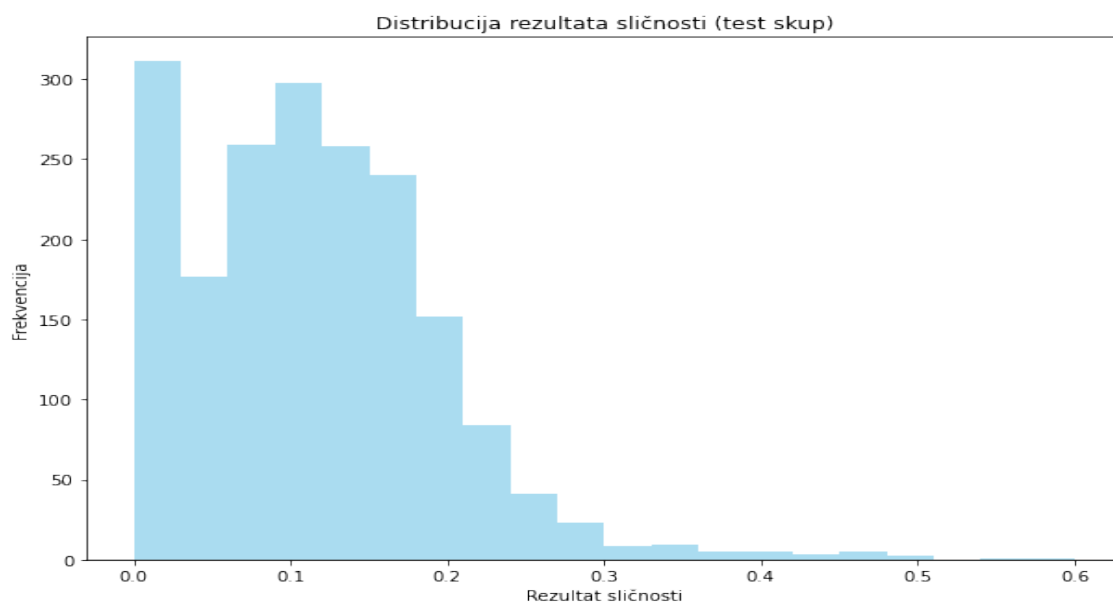


Slika 4.15 Distribucija sličnosti - novi peptidi s trening peptidima

Poglavlje 4. Rezultati



Slika 4.16 Rezultati slicnosti ovisno o indeksu - novi peptidi s testnim peptidima



Slika 4.17 Distribucija slicnosti - novi peptidi s testnim peptidima

Poglavlje 4. Rezultati

U zadnjem dijelu analize provedeno je vrednovanje kvalitete novo generiranih peptida predviđanjem aktivnosti antimikrobnog svojstva. Koristeći se nepristranim modelom koji može predviđati takva fizikalno-kemijska svojstva, cilj nam je ocijeniti koliko su dobro novo generirani peptidi zadržali antimikrobnu karakteristiku uspoređujući ih sa izvornim oznakama aktivnosti pojedinačno trening i testnog skupa.

Kako bi stvorili reprezentacije peptida kompatibilne s nepristranim modelom za predviđanje antimikrobnog svojstva, korištena je `SequentialPropertiesEncoder` klasa iz python knjižnice `seqprops`. Nadalje, numeričke reprezentacije peptida su skalirane i oblikovane kako bi odgovarale ulazu nepristranog modela. Nakon pripreme podataka, učitano nepristranom modelu postavljene su težine predefinirane za ovaj slučaj, tako da možemo predviđati s obzirom na antimikrobno svojstvo. U slučaju da predviđena vjerojatnost prelazi prag od 0,5, takav novi peptid biti će označen kao pozitivan, inače kao negativan.

Za nove peptide generirane iz skupa peptida za trening kojih je ukupno 7,527; dobiveno je točno predviđanje oznake aktivnosti za 5,104, dok je broj krivih predviđanja bio 2,423. To rezultira točnošću predviđanja antimikrobne aktivnosti od 68%, iz čega se može zaključiti da je VAE model prilikom generiranja novih peptida u velikoj većini uspio zadržati antimikrobno svojstvo ulaznih peptida. Detaljniji prikaz rezultata, vidljiv je pomoću matrice zabune (eng. *Confusion Matrix*) nepristranog modela pri predviđanju novih peptida generiranih iz trening skupa u tablici 4.2.

		Predikcija klasifikatora	
		Pozitivni peptid	Negativni peptid
Izvorna oznaka antimikrobne aktivnosti	Pozitivni peptid	3226	1341
	Negativni peptid	1082	1878

Tablica 4.2 Matrica zabune predviđanja aktivnosti novih peptida - trening skup

Za nove peptide generirane iz skupa peptida za testiranje kojih je ukupno 1,882; dobiveno je točno predviđanje oznake aktivnosti za 1,256, dok je broj krivih predviđanja bio 626. To rezultira točnošću predviđanja antimikrobne aktivnosti od 67%, iz čega se može potvrditi gore navedeni zaključak. Detaljniji prikaz rezultata, vidljiv je pomoću matrice konfuzije nepristranog modela pri predviđanju novih peptida iz

Poglavlje 4. Rezultati

testnog skupa u tablici 4.3.

		Predikcija klasifikatora	
		Pozitivni peptid	Negativni peptid
Izvorna oznaka antimikrobne aktivnosti	Pozitivni peptid	800	334
	Negativni peptid	292	456

Tablica 4.3 Matrica zabune predviđanja aktivnosti novih peptida - testni skup

Poglavlje 5

Zaključak

Zbog rastuće potrebe za novim i učinkovitim terapijskim peptidima, istraživačka zajednica ulaže velike napore u otkrivanje novih pristupa za njihov uspješan dizajn. Eksperimentalne metode otkrivanja peptida u laboratorijima su skupocjene, spore i ograničene. Stoga se za dizajn novih peptida, osobito onih antimikrobnih, sve više koriste metode strojnog učenja te je to područje postalo iznimno aktivno. Generiranje i optimizacija peptida putem modela VAE predstavlja jedan od primjera kako računalna znanost može unaprijediti proces istraživanja novih lijekova na osnovu znanja prikupljenog u mnogobrojnim uspješnim i neuspješnim eksperimentima, na taj način ubrzavajući i olakšavajući postupak otkrivanja na način koji je tradicionalnim postupcima bio nedokučiv.

U ovom istraživanju fokus je bio na primjeni varijacijskih autoenkodera u svrhu stvaranja novih peptidnih sekvenci, stavljajući naglasak na razumijevanje veza između sekvenci koje čine jedan peptid. Uz to cilj je bio proučiti mehanizme rada samih VAE modela te kako podešavanje parametara slojeva modela i mjenjanje svojstva funkcija gubitaka utječe na generativna obilježja.

Kroz analizu rezultata otkriveno je da model ima sklonost učiti smanjivanjem Kulback-Leiblerova gubitka rezultirajući latentnim prostom koji više zadržava karakteristike cjelovitosti nego kontinuiranosti. Generiranjem novih sekvenci iz takvog latentnog prostora dobivamo nove peptide koji u 68% slučajeva zadržavaju točne oznake aktivnosti antimikrobnog svojstva, što je potvrđeno predviđanjem nepristranog

Poglavlje 5. Zaključak

modela za predviđanje antimikrobnog svojstva. Pri tome, zadržan je kraći format dužine peptidne sekvence u usporedbi sa trening i testnim skupovima peptida. S druge strane, manjak kontinuiranosti latentnog prostora te preklapanje peptida različitih oznaka aktivnosti u samom prostoru rezultira novim peptidima koji, iako zadržavaju svojstva ulaznih, usporedbom samih sekvenci vrlo su različiti. Iako je sličnost distribucija dužina novih peptida sa ulaznim zadržana, iz analize je vidljivo da je sličnost na bazi poretka aminokiselina novih peptida u usporedbi sa ulaznim iznosila samo do 40%.

Ovo istraživanje potvrđuje potencijal koji imaju VAE modeli za generiranje novih antimikrobnih peptida, međutim naglašava važnost kontinuiranog istraživanja i prilagođavanja modela kako bi se postigli traženi rezultati. Primjenom drugačije arhitekture VAE modela, korištenjem različitih slojeva i ukupnog broja slojeva, promjenom veličina latentnih dimenzija i daljnjom prilagodbom funkcija gubitaka mogu se za rezultat dobiti VAE modeli novih karakteristika, te s tim novi potencijali za primjenu u različite generativne svrhe.

Bibliografija

- [1] TowardsDataScience, “Understanding variational auto-encoders (vae),” <https://towardsdatascience.com/understanding-variational-autoencoders-vae-f70510919f73>, 2022, s intenet; pristupljeno 02.09.2022.
- [2] P. Das, T. Sercu, K. Wadhawan, I. Padhi, S. Gehrmann *et al.*, “Accelerated anti-microbial discovery via deep generative models and molecular dynamics simulations,” <https://www.nature.com/articles/s41551-021-00689-x>, 2023, s intenet; pristupljeno 31.11.2023.
- [3] Wikipedia, “Aminokiseline,” <https://bs.wikipedia.org/wiki/Aminokiselina>, 2022, s intenet; pristupljeno 03.09.2022.
- [4] TechnologyNetworks, “Aminokiseline dodatno,” <https://www.technologynetworks.com/applied-sciences/articles/essential-amino-acids-chart-abbreviations-and-structure-324357>, 2022, s intenet; pristupljeno 01.09.2022.
- [5] EnciklopedijaHr, “Peptidi,” <https://www.enciklopedija.hr/natuknica.aspx?ID=47511>, 2022, s intenet; pristupljeno 01.09.2022.
- [6] MachinelearningMastery, “Introduction to dimensionality reduction for machine learning,” <https://machinelearningmastery.com/dimensionality-reduction-for-machine-learning/>, 2022, s intenet; pristupljeno 04.09.2022.
- [7] TowardsDataScienceWeb, “A one-stop shop for principal component analysis,” <https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c>, 2022, s intenet; pristupljeno 04.09.2022.
- [8] JeremyJordan, “Variational autoencoders.” <https://www.jeremyjordan.me/variational-autoencoders/>, 2022, s intenet; pristupljeno 04.09.2022.

Bibliografija

- [9] TowardsDataScience, “Regularization in machine learning,” <https://towardsdatascience.com/regularization-in-machine-learning-76441ddcf99a>, 2022, s internetu; pristupljeno 04.09.2022.
- [10] A. Hawkins-Hooker, F. Depardieu, S. Baur *et al.*, “Generating functional protein variants with variational autoencoders,” <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1008736>, 2023, s internetu; pristupljeno 29.10.2023.
- [11] S. N. Dean and S. A. Walper, “Variational autoencoder for generation of antimicrobial peptides,” <https://pubs.acs.org/doi/10.1021/acsomega.0c00442?ref=pdf>, 2023, s internetu; pristupljeno 29.10.2023.
- [12] MediumRNN, “Recurrent neural network: A practical guide to rnn,” <https://medium.com/@usama.6832/recurrent-neural-network-a-practical-guide-to-rnn-26befdf07047>, 2023, s internetu; pristupljeno 29.10.2023.
- [13] Medium, “Understanding rnn and lstm,” <https://aditi-mittal.medium.com/understanding-rnn-and-lstm-f7cdf6dfc14e>, 2023, s internetu; pristupljeno 29.10.2023.
- [14] G. Shi, X. Kang, F. Dong, Y. Liu, N. Zhu, Y. Hu, H. Xu, X. Lao, and H. Zheng, “Welcome to dramp database,” <http://dramp.cpu-bioinform.org/>, 2023, s internetu; pristupljeno 29.10.2023.
- [15] X. Kang, F. Dong, C. Shi, S. Liu, J. Sun, J. Chen, H. Li, H. Xu, X. Lao, and H. Zheng, “Dramp 2.0, an updated data repository of antimicrobial peptides,” 2019, *scientific Data*, vol. 6.
- [16] T. U. Consortium, “Uniprot: the universal protein knowledgebase in 2023,” *Nucleic Acids Research*, vol. 51, no. D1, pp. D523–D531, 11 2022. , s Internetu, <https://doi.org/10.1093/nar/gkac1052>

Pojmovnik

AMP Antimicrobial peptides. 18, 22

KL Kulback-Leibler divergence. 15, 16, 29–31

LSTM Long Short-Term Memory. 18–20

PCA Principal components analysis. 7–10

RNN Recurrent Neural Network. 18, 19

VAEs Variational Autoencoders. 5, 13, 15, 16

Sažetak

Rastuća prijetnja rezistencije na antibiotike ističe potrebu za inovativnim terapijskim strategijama. Antimikrobni peptidi (AMP-ovi) su se pojavili kao potencijalni kandidati s izvanrednim antimikrobnim svojstvima. Duboki generativni modeli, posebice varijacijski autoenkoderi (VAE), obećavaju učinkovito generiranje ovih peptida. Ovo istraživanje ispituje primjenu VAE modela za generiranje AMP-ova, ističući njihov potencijal u borbi protiv rezistencije na antibiotike. Naglašava sposobnost VAE modela za olakšavanje dizajna AMP-ova, uzimajući u obzir važnost daljnjeg usavršavanja modela kako bi se potpuno iskoristile njihove mogućnosti i optimizirali rezultati.

Ključne riječi — antimikrobni peptidi, autoenkoderi, varijacijski autoenkoderi, generativni modeli, oblikovanje peptida

Abstract

The increasing menace of antibiotic resistance underscores the need for innovative therapeutic strategies. Antimicrobial peptides (AMPs) have emerged as potential candidates, exhibiting remarkable antimicrobial properties. Deep generative models, particularly Variational Autoencoders (VAEs), offer promise in the efficient design of these peptides. This study delves into the application of VAEs for AMP generation, highlighting their potential in combating antibiotic resistance. It underscores VAEs' ability to facilitate AMP design while recognizing the importance of further model refinement to fully unleash their capabilities and optimize outcomes.

Keywords — Antimicrobial Peptides, Autoencoders, Variational Autoencoders, Generative Models, Peptide Design