



Univerzitet Singidunum
Departman za poslediplomske studije

Doktorski akademski program
Inteligentno softversko inženjerstvo

Doktorska disertacija

**UNAPREĐENJE PRECIZNOSTI PREDVIĐANJA CENA
GORIVA PRIMENOM MODELA DUGOROČNO-
KRATKOROČNE MEMORIJE NEURONSKIH MREŽA
SA VIŠE GLAVA OPTIMIZOVANIH
METAHEURISTIKOM ROJEVA ČESTICA**

Mentor:

prof. dr Nebojša Bačanin-Džakula

Student:

Anđela Jovanović, 2022/480162

Beograd, 2025.

UNIVERZITET SINGIDUNUM
DEPARTMAN ZA POSLEDIPLOMSKE STUDIJE
Danijelova 32, Beograd

Kandidat: Anđela Jovanović, master softverski inženjer
Broj ineksa: 2022/480162
Studijski program: Inteligentno softversko inženjerstvo

Tema:

UNAPREĐENJE PRECIZNOSTI PREDVIĐANJA CENA GORIVA PRIMENOM MODELA
DUGOROČNO-KRATKOROČNE MEMORIJE NEURONSKIH MREŽA SA VIŠE GLAVA
OPTIMIZOVANIH METAHEURISTIKOM ROJEVA ČESTICA

Mentor:

Prof. dr Nebojša Bačanin-Džakula
Redovni profesor Univerziteta Singidunum, Beograd

Članovi Komisije:

Prof. dr Miodrag Živković
Redovni profesor Univerziteta Singidunum, Beograd

Prof. dr Vladimir Simić
Redovni profesor Saobraćajnog fakulteta
Univerzitet u Beogradu

Rezime

Povećana potražnja za energijom uz značajno smanjenje dostupnih zaliha fosilnih goriva na globalnom nivou, kao i brojni socio-ekonomski faktori, usloveli su intezivan razvoj brojnih istraživanja usmerenih na predikciju potrošnje energije i kretanje njenih cena. Predikcije ovih parametara postaju sve značajnije, kako za naučnu zajednicu, tako i za privredu, ekonomiju i društvo. U cilju dobijanja što preciznijih rezultata, ovakva istraživanja zahtevaju rad sa velikom količinom podataka. Prikupljanje, kao i obrada istih može biti dosta složen i dugotrajan proces, koji je vrlo često podložan greškama. S toga se intezivnim razvojem savremenih tehnologija, posebno Veštačke inteligencije, značajno automatizuje ovaj zadatak i smanjuje faktor ljudske greške. Zahvaljujući velikoj popularnosti ovih modela, predikcija cena goriva uspešno pronalazi svoju primenu u brojnim poslovnim domenima. Imajući ovo u vidu, ova disertacija se primarno bavi rešavanjem problema predikcije vremenskih serija, odnosno cena goriva, kroz primenu i optimizaciju postojećih modela Veštačke inteligencije, poput neuronskih mreža. Iako relativno nova oblast, ona pokazuje značajan potencijal za unapređenje energetske analitike, sa brojnim praktičnim i ekonomskim primenama. Analizom postojeće literature uočava se prostor za dodatno unapređenje u ovoj oblasti, pa ova disertacija predlaže i istražuje potencijal drugih modela koji mogu još više poboljšati predikciju vremenskih serija, poput dugoročno-kratkoročnih memorijskih modela sa više glava (*eng. multi-headed LSTM*). Imajući u vidu da ove arhitekture imaju relativno visoke računarske zahteve, istraživanje ove disertacije se dodatno fokusira i na optimizovane pristupe modelovanja istih. Ovo se ogleda u tome što se arhitekture projektuju tako da imaju manji broj neurona po sloju, kao i u skraćivanju vremena potrebnom za obuku modela, odnosno manjem broju epoha. Međutim, treba imati u vidu da ne postoji dovoljno dobro univerzalno rešenje koje je primenljivo na svim tipovima podataka i koje je superiorno u odnosu na ostale. Ovu činjenicu potvrđuje i čuvena teorema „nema besplatnog ručka“ (*eng. No Free Lunch Theorem, NFLT*), koja naglašava potrebu da se ove arhitekture prilagode specifičnim skupovima podataka. U cilju rešavanja ovog problema, u ovom radu se predlaže i korišćenje metaheurističkih tehnika optimizacije. Konkretno govoreći, predstavljena je modifikovana varijanta algoritma optimizacije rojem čestica (*eng. Particle Swarm Optimizer, PSO*), koja je korišćena u cilju što tačnijeg podešavanja arhitekture mreže, kao i samih parametara koji su korišćeni tokom obučavanja predloženog modela. Na kraju, ovako modifikovan model je primenjen nad realnim energetskim podacima koji su prikupljeni iz više javno dostupnih izvora. Tako dobijeni rezultati su tehnikama

objašnjive inteligencije (*eng. Explainable Artificial Intelligence*) poređeni sa savremenim optimizatorima u okviru komparativne analize. Detaljnom statističkom validacijom je potvrđena tačnost i značajnost dobijenih rezultata.

Ključne reči: LSTM, NFLT, Metaheuristike, PSO, Neuronske mreže, Vremenske serije

Abstract

The growing demand for energy, coupled with the significant decline in globally available fossil fuel reserves, as well as numerous socio-economic factors, has driven the intensive development of research focused on forecasting energy consumption and price trends. Forecasting these parameters is becoming increasingly important not only for the scientific community but also for industry, the economy, and society as a whole. To obtain accurate results, such research requires working with large volumes of data. However, data collection and processing can often be complex and time-consuming processes, highly prone to errors. With the rapid development of modern technologies, particularly Artificial Intelligence, these tasks are increasingly automated, thereby reducing the impact of human error. Owing to the growing popularity of such models, fuel price forecasting has found successful application across numerous business domains. Considering this, the present dissertation primarily addresses the problem of time series forecasting, specifically fuel prices, through the application and optimization of existing Artificial Intelligence models such as neural networks. Although still a relatively young field, it demonstrates significant potential for advancing energy analytics, with a wide range of practical and economic applications. A review of the existing literature reveals room for further improvement in this area; therefore, this dissertation proposes and explores the potential of other models that may enhance time series prediction, such as multi-headed Long Short-Term Memory (LSTM) networks. Given that these architectures generally require high computational resources, this research further focuses on optimized modeling approaches. This is reflected in the design of architectures with fewer neurons per layer and reduced training time, expressed in a lower number of epochs. Nevertheless, it must be emphasized that no universally superior solution exists that can be applied effectively across all data types. This fact is confirmed by the well-known “No Free Lunch Theorem” (NFLT), which highlights the need to adapt architectures to specific datasets. To address this challenge, the dissertation also proposes the use of metaheuristic optimization techniques. Specifically, a modified variant of the Particle Swarm Optimization (PSO) algorithm is presented, employed to more accurately fine-tune the network architecture as well as the training parameters of the proposed model. Finally, the modified model was applied to real-world energy datasets collected from multiple publicly available sources. The results obtained were compared with state-of-the-art optimizers within a comparative analysis, using Explainable Artificial Intelligence (XAI)

techniques. Detailed statistical validation confirmed both the accuracy and the significance of the obtained results.

Keywords: LSTM, NFLT, Metaheuristics, PSO, Neural networks, Time series prediction.

SADRŽAJ

1	Uvod.....	10
1.1	<i>Metodološke osnove istraživanja</i>	<i>11</i>
1.1.1	Problem istraživanja	11
1.1.2	Predmet istraživanja.....	12
1.1.3	Ciljevi i zadaci istraživanja.....	13
1.1.4	Hipoteze u istraživanju	14
1.1.5	Metode istraživanja.....	16
1.1.6	Naučna i društvena opravdanost istraživanja i očekivani naučni doprinos.....	17
1.2	Struktura rada.....	20
2	Teorijska osnova istraživanja.....	22
2.1	<i>Uvod i kratak istorijat korišćenja energenata</i>	<i>22</i>
2.2	<i>Uticao cena goriva na globalno tržište i ekonomiju.....</i>	<i>25</i>
2.3	<i>Primena Veštačke inteligencije na predikciju cena goriva.....</i>	<i>28</i>
2.3.1	Kratak istorijat veštačke inteligencije	29
2.3.2	Često korišćeni modeli.....	31
2.3.3	Vremenske serije	37
3	Analiza naučnih radova	45
4	Duboko učenje.....	52
4.1	<i>Uvod i kratak istorijat.....</i>	<i>52</i>
4.2	<i>Rekurentne neuronske mreže</i>	<i>55</i>
4.3	<i>LSTM i GRU modeli</i>	<i>62</i>
4.3.1	LSTM model.....	62
4.3.2	GRU model	65
4.3.3	Model rekurentnih mreža sa više glava.....	67
4.4	<i>Problem odabira hiperparametara u dubokom učenju.....</i>	<i>71</i>
5	Algoritmi optimizacije hiperparametara	73
5.1	<i>Heurističke tehnike optimizacije.....</i>	<i>73</i>
5.1.1	Mrežna pretraga	74
5.1.2	Nasumična pretraga	76
5.2	<i>Metaheurističke tehnike.....</i>	<i>78</i>

5.3	<i>Pregled algoritma čestica rojeva</i>	79
5.3.1	Teorijska osnova.....	79
5.3.2	Matematički model	79
5.3.3	Prednosti i nedostaci	80
5.3.4	Primena u predikciji vremenskih serija.....	80
6	Pregled istraživačkog dela	82
6.1.1	Opis istraživanja	82
6.1.2	Korišćene procedure i tehnike	83
6.2	<i>Ekperimentalna postavka, rezultati simulacije i perspektiva objašnjivosti</i>	95
6.2.1	Primenjeni kriterijumi za ocenjivanje	97
6.2.3	Skup podataka korišćen u eksperimentima	98
6.2.4	Konačni rezultati simulacije	99
6.2.5	Statistička validacija	108
6.2.6	Analiza osetljivosti predloženog SAGPSO algoritma i poređenje sa sličnim pristupima	110
6.2.7	Tumačenje uticaja karakteristika najboljeg modela korišćenjem SHAP Analize	112
7	Zaključak.....	114
	Literatura	116
	PRILOZI	123
7.1	<i>Prilog A – Pseudokovi metaheurističkih algoritama</i>	123
7.1.1	PSO Algoritam.....	123
7.1.2	Salp Swarm Algoritam (SSA).....	124
7.1.3	Artificial Rabbits Optimization Algoritam (ARO)	125
7.1.4	Red Fox Optimization Algoritam Algoritam (RFO).....	126
7.1.5	Bat Algoritam (BA)	127
7.1.6	Firefly Algoritam (FA)	128
7.1.7	Genski Algoritam (GA)	129
7.2	<i>Prilog B – Pseudokovi modela neuronskih mreža</i>	130
7.2.1	Algoritam Rekurentne Neuronske mreže	130
7.2.2	Algoritam Dugoročno-kratkoročne memorije.....	131
7.2.3	Algoritam Zatvorene rekurentne jedinice (GRU)	132
7.2.4	Algoritam Dugoročno-kratkoročne memorije sa više glava (MHLSTM).....	133
	Spisak akronima	134
	Spisak oznaka.....	136
	Spisak tabela	138

Spisak slika.....	139
Spisak formula	141

1 Uvod

U savremenom dobu, praćenje i razumevanje posledica oscilacija cena različitih izvora energije na ukupni finansijski okvir raznih sektora, uključujući transport, a samim tim i životni standard, postalo je od ključnog značaja. Tokom poslednjih godina primećuje se da je ovo područje vrlo često podložno uticaju određenih spoljašnjih faktora, kao što su pandemije, međunarodni konflikti i brojni socio-ekonomski događaji. Posmatranje promena u ceni energenata, konkretno goriva, i njihov efekat na privredu i društvo u celini dobija sve više na značaju. Međutim, kao što je u rezimeu napomenuto, ovaj proces može biti dugotrajan i iscrpljujuć, a vrlo često i podložan greškama, ako mu se ne pristupi adekvatno i sa dobrom strategijom ubrzanja i automatizacije. Kada govorimo o analizi velike količine podataka koji se tokom vremena mogu menjati, otvara se pitanje mogućnosti da se ovaj proces u potpunosti prepusti mašinama, odnosno računarima. Takođe je bitno razumeti i kako se on može potencijalno generalizovati, tako da bude primenljiv na različite tipove podataka. Ako pođemo od ove pretpostavke, a uzimajući u obzir ograničene računarske resurse, postavlja se dodatno pitanje da li današnji računarski sistemi mogu postati dovoljno „inteligentni“ da samostalno iz velike količine podataka izvuku relevantne informacije u odgovarajućem kontekstu, na osnovu kojih bi samostalno mogli donositi odluke. Imajući ovo u vidu, potrebno je dakle naglasiti važnost postojećih metoda predikcije potrošnje energije, odnosno goriva, koji se primenjuju nad velikom količinom istorijskih podataka, jer upravo oni predstavljaju važnu komponentu u modelovanju ovako specifičnih informacionih modela. Pored ovoga, kao što je ranije napomenuto, treba uzeti u obzir i brojne spoljašne činioce – socioekonomske uslove i dostupnost resursa na globalno nivou, distribucije i međunarodne politike. Imajući ovo u vidu, domen energetske analitike postaje sve značajniji, privlačeći veliku pažnju naučnika kako na domaćem, tako i na međunarodnom nivou. U savremenoj literaturi se mogu pronaći brojni primeri primene Veštačke inteligencije, kao i Mašinskog učenja u cilju rešavanja prethodno opisanih problema. Postoji jako velika količina modela i algoritama koji se mogu uzeti u obzir i koristiti kao osnovni alati u istraživačkom procesu. Međutim, tu se dolazi do određene „sive-zone“, gde i pored velike količine postojećih istraživanja, zbog prirode samog problema koji se proučava, postoje određeni segmenti koji nisu u potpunosti pokriveni, ili su nedovoljno istraživani. Ovime se otvara prostor za unapređenje i optimizaciju postojećih rešenja, kao i za razvoj nekih novih rešenja i njihovu primenu na stvarnim podacima.

1.1 Metodološke osnove istraživanja

1.1.1 Problem istraživanja

Iako postoje mnogobrojni modeli Veštačke inteligencije koji se, između ostalog, mogu primeniti i za predikciju cena goriva, ovo ostaje i dalje kao izazovan problem zbog uticaja mnogih faktora kao i zbog izražajne fluktuacije samih podataka. Postojeće arhitekture često ne uspevaju da precizno modeluju kompleksne odnose u okviru samih podataka, posebno tokom vrlo dinamičnih i ekstremnih tržišnih promena, kojima trenutno svedočimo. Dodatno, činjenica je da i postojeći modeli relativno brzo zastarevaju, jer se i sama Veštačka inteligencija jako brzo razvija kako bi mogla da isprati trenutne trendove i potrebe kako običnih ljudi, tako i vodećih korporacija. Takođe, kada govorimo o samoj prirodi ovih podataka, neke vremenske serije mogu biti neosporno nepredvidive zbog svog haotičnog ponašanja, te bi trebalo uzeti u obzir i neke teorijske granice u okviru kojih možemo sprovesti dovoljnu količinu eksperimenata u cilju dobijanja što preciznijih rezultata.

Treniranjem modela obično dolazimo i do dodatnog problema optimizacije hiperparametara. U velikoj većini slučajeva, klasične metode podešavanja hiperparametara, na žalost, nisu dovoljno efikasne kod kompleksnih modela koji se koriste za predviđanje vremenskih serija, te je potrebno razviti pristup koji primenjuje metaheurističke algoritme u cilju rešavanja ovog problema.

Imajući u vidu da prilikom formulacije istraživačkog problema krećemo od poznatog ka nepoznatom, postavlja se pitanje da li je moguće generisati nova ili dovoljno dobro optimizovati postojeća rešenja koristeći metaheuristike, kako bi se sa visokom preciznošću i tačnošću mogle predvideti buduće cene goriva? U ovom kontekstu, postavljaju se određena potpitanja na koja će ova teza pokušati da iznese odgovore:

- Kako integrisati određene spoljašnje faktore, kao što su sirove cene nafte, političke i socioekonomske indikatore, podatke o potražnji i mnoge druge u sam model?
- Kako dovoljno dobro izabrati i optimizovati određene hiperparametre koristeći metaheuristike?
- Kako obezbediti da sam model bude dosledan promenama koje nastaju u podacima i prilagodljiv istim?

1.1.2 Predmet istraživanja

Analizom postojeće literature i problema koji su trenutno aktuelni na globalnom nivou, javila se potreba za dodatnim eksperimentisanjem i istraživanjem u oblasti primenjene Veštačke inteligencije, naročito u kontekstu predikcija vremenskih serija. Glavni predmet ovog istraživanja predstavlja razvijanje metodologije za obuku i modelovanje sekvencijalnih arhitektura, koje su dodatno optimizovane koristeći metaheurističke tehnike, kao i njihova primena nad velikom količinom podataka, na čiju prirodu i stabilnost efektivno utiče dosta eksternih faktora. Cilj samog istraživačkog procesa je dobijanje jednog informacionog modela baziranog na modernim arhitekturama, a u cilju postizanja što preciznijih konačnih rezultata nakon njegove primene na realnim podacima. Imajući ovo u vidu, sam istraživački rad može podeliti na dve strukturalne celine, teorijski i praktični deo.

Predmet teorijskog dela istraživačkog rada se svodi na primenu metoda višekriterijumske analize postojećih modela i tehnika koje se koriste u rešavanju navedenog problema, kako bi se odabrao adekvatan početni model koji će kasnije poslužiti kao osnov za definisanje hipoteza i njegovo dalje unapređenje. Pored toga, kao što je ranije napomenuto, uočeni su i opisani određeni nedostaci u postojećim modelima i tehnikama, koji su opisani u dostupnoj literaturi, a koje ovo istraživanje ima za cilj da nadomesti.

Praktični deo istraživanja je usmeren ka opisu konkretnih koraka unapređenja preciznosti predikcije cena goriva kao glavnog problema. Akcenat je stavljen na razvoj i implementaciju unapređenog modela koji kombinuje arhitekturu neuronskih mreža sa više glava i dugoročno-kratkoročne memorije (*eng. Multi-head Long Short-Term Memory, MH-LSTM*) sa metaheurističkom optimizacijom zasnovanom na algoritmu rojeva čestica (*eng. Particle Swarm Optimization, PSO*).

Imajući ovo u vidu, može se pretpostaviti da se istraživanje fokusira na sledeće ključne celine:

- Adaptacija postojećih neuronskih mreža kroz modelovanje MH-LSTM arhitekture, koja može paralelno procesuirati više ulaznih informacija, odnosno podataka kroz različite „glave“ (*eng. heads*), čime se omogućava da model bolje ekstrahuje i razume kompleksne veze i obrasce u samim podacima.

- Primena metaheurističkog optimizatora roja čestica radi efikasnog pronalaženja optimalnih vrednosti hiperparametara, koji mogu značajno poboljšati preciznost konačnih rezultata nakon izvršenih eksperimenata.
- Evaluacija predloženog modela na realnim podacima sa ciljem da se dokaže njegova superiornost u odnosu na već postojeće i primenjene modele u ovoj oblasti.
- Sistematska provera i analiza performansi predloženog modela, kao i njegovo poređenje sa ostalim postojećim pristupima, sa osnovnim ciljem dokazivanja njegove tačnosti, kao i pouzdanosti tako dobijenih rezultata.

1.1.3 Ciljevi i zadaci istraživanja

Ciljevi istraživanja su sledeći:

- Identifikacija i nadomešćivanje određenih nedostataka u postojećim modelima koji se konkretno primenjuju u svrhu rešavanju problema predviđanja cena goriva
- Primena nove arhitekture neuronskih mreža koja može paralelno obrađivati višestruke vremenske obrasce kroz više memorijskih glava (*eng. heads*)
- Formulisanje problema optimizacije hiperparametara, kao i njegovo rešavanje korišćenjem metaheurističkog algoritma optimizacije rojem čestica (*eng. Particle Swarm Optimizer, PSO*)
- Testiranje predloženog modela i njegova eksperimentalna evaluacija na relevantnim skupovima realnih podataka koji su prikupljeni iz više različitih javno dostupnih izvora
- Poređenje performansi predloženog modela sa standardnim pristupima koji su se koristili za rešavanje istog problema
- Interpretacija dobijenih rezultata i generisanje značajnih informacija za stručnjake iz domena tržišta energenata
- Korišćenje tako dobijenih informacija radi donošenja što boljih biznis odluka i identifikaciji neefikasnih delova postojećih sistema

Da bi se ostvarili postavljeni **istraživački ciljevi**, potrebno je izvršiti **zadatke** koji obuhvataju sledeće:

- Sveobuhvatni pregled modela i algoritama predviđanja cena goriva, kao i prikaz njihove korelacije, a koji su raspoloživi u dostupnoj literaturi.
- Definisanje predložene arhitekture (npr. definisanje broja glava, način njihove integracije, broj epoha itd.).
- Modelovanje optimizacije hiperparametara kao problem globalne optimizacije tako da je pogodan za rešavanje primenom predloženog PSO algoritma.
- Implementacija ovakvog predloženog sistema kroz određeni programski jezik.
- Priprema podataka nad kojima će se sprovesti eksperimenti, a koja između ostalog uključuje osnovne korake normalizacije, podelu na trening, validaciju i test skupove .
- Poređenje rezultata i performansi predložene arhitekture sa referentnim modelima koristeći standardne metrike.
- Formulisanje preporuka za buduće primene i dalje unapređenje modela, zasnovanih na rezultatima istraživanja.

1.1.4 Hipoteze u istraživanju

Analizom opisanog problema, predmeta kao i cilja ovog istraživanja, mogu se definisati sledeće **hipoteze**:

Opšta hipoteza:

Primenom do sada nedovoljno korišćenih i istraženih modela rekurentnih neuronskih mreža za multivarijantno predviđanje cena goriva moguće je generisati bolja rešenja, kako u pogledu brzine konvergencije, tako i u pogledu samog kvaliteta rešenja, čime će se značajno poboljšavati konačni ishodi, a u odnosu na druge metode optimizacije korišćene za istu instancu problema.

Posebna hipoteza:

Na osnovu opšte hipoteze se može izdvojiti posebna hipoteza, koja se odnosi na sam predmet istraživanja, a koja glasi:

H1: Moguće je dovoljno dobro adaptirati i optimizovati postojeće implementacije modela rekurentnih neuronskih mreža za multivarijantno predviđanje vremenskih serija predviđanja cena goriva.

Daljim preciziranjem navedene posebne hipoteze, formulišu se pojedinačne koje se odnose na elementarne činioce predmeta istraživanja:

H1a: Rekurentne neuronske mreže (*eng. Recurrent Neural Networks, RNN*) mogu se dodatno unaprediti kroz specijalizovane arhitekture i optimizacione tehnike, kako bi bolje obavljale zadatak predikcije cena goriva. Proučavanjem različitih pristupa optimizacije RNN modela, moguće je poboljšati njihovu sposobnost generalizacije i samim time tačnost generisanog izlaza.

H1b. Moguće je dovoljno dobro unaprediti postojeće implementacije metaheurističkog algoritma rojeva čestica (*eng. Particle Swarm Optimization, PSO*), koje su ranije već bile primenjene u ovom domenu, radi još bolje optimizacije predloženog modela rekurentnih neuronskih mreža, a u cilju predviđanja cena goriva.

H1c: Upotrebom arhitekture dugoročno-kratkoročnih memorijskih mreža sa višestrukim glavama (*eng. multi-head LSTM*) moguće je značajno poboljšati preciznost predviđanja cena goriva, s obzirom na to da ova metoda omogućava paralelnu obradu različitih aspekata ulaznih podataka, čime se značajno poboljšava kvalitativna i strukturna usklađenost generisanog izlaza sa očekivanim rezultatima.

1.1.5 Metode istraživanja

Da bi se razvile hipoteze koje su u skladu sa predmetom i ciljem istraživanja, biće primenjene metode istraživanja koje su uobičajene u praksi i naučno potvrđene: analitičko–sintetička metoda, deduktivno–induktivna metoda, komparativno–kvantitativna analiza, empirijska metoda, statistička analiza, modeliranje i eksperimentalni deo za dokazivanje hipoteza.

Analitičkom metodom će se obaviti detaljna analiza sadržaja dostupne relevantne naučne i stručne literature u navedenoj oblasti istraživanja, kao i svih relevantnih činjenica publikovanih putem elektronskih ili drugih medija, kako bi se utvrdili određeni nedostaci kao i prostor za unapređenje.

Sintetičkom metodom će se integrisati raščlanjeni analizirani elementi pojave u jedinstvenu celinu, radi definisanja određenih pravila i zakonitosti u ponašanju pojave koja je predmet istraživanja. Drugim rečima, svi prikupljeni podaci biće grupisani metodom sinteze, kako bi se došlo do pouzdanih zaključaka.

Korišćenje **deduktivno–induktivne** metode je u funkciji usmeravanja istraživanja od opšteg ka pojedinačnom, odnosno od pojedinačnog ka opštem u cilju dolaska do adekvatnih zaključaka.

U okviru **komparativno–kvantitativne** analize, a u korelaciji sa ciljevima i problemom istraživanj, kao najadekvatniji pristup izabrana je metoda analize. Konkretno, za analizu prikupljenih rezultata nakon primene metoda, tehnika i metaheuristika za rešavanje problema multivarijantnog predviđanja vremenskih serija cena goriva, koristiće se metoda **kvantitativne analize**. Metodom **komparativne analize** uporediće se dobijeni rezultati optimizacije i izvešće se zaključak o stvarnom unapređenju rešavanja ovog problema primenom metaheurističkih optimizacionih algoritama.

Primenom **empirijske metode** prikupljeni primarni podaci će biti razmatrani i u svetlu diskutovanih postojećih teorijskih i empirijskih nalaza, odnosno sekundarnih izvora podataka, kako bi se testirale polazne istraživačke hipoteze. Praktičnim eksperimentisanjem (izvršavanjem predloženog algoritma) dolazi se do primarnih podataka koji će se koristiti u daljoj analizi.

Statistička analiza podrazumeva korišćenje statističkih podataka i metoda u obradi dobijenih rezultata, a u skladu sa postavljenim istraživačkim ciljevima.

Modeliranje podrazumeva konstrukciju matematičkih modela koji predstavljaju validnu predstavu problema koji se proučava i rešava. Za rešavanje optimizacionih problema takođe se posmatraju matematički modeli i formulacije, kao i funkcije cilja problema istraživanja. U eksperimentalnom delu implementacije, sa dodatnim akcentom se sagledavaju tehnike metaheuristika u rešavanju prethodno pomenutog optimizacionog problema.

1.1.6 Naučna i društvena opravdanost istraživanja i očekivani naučni doprinos

Na osnovu prethodno navedenog, mogu se izdvojiti sledeći očekivani naučni doprinosi:

- Analizira različitih modela i algoritma mašinskog učenja koji se koriste za multivarijantno predviđanje vremenskih serija, uključujući njihovu klasifikaciju i karakteristike.
- Istraživanje kako se primenom naprednih modela rekurentnih neuronskih mreža, uz korišćenje metaheurističkih metoda optimizacije, može poboljšati tačnost predviđanja prethodno pomenute instance problema, odnosno tačnija predikcija cena goriva.
- Detaljno poređenje performansi različitih modela mašinskog učenja u rešavanju ovog tipa problema, kao i analiza unapređenih modela optimizovanih različitim metaheurističkim pristupima koji su zastupljeni u literaturi .
- Pristupiće se interpretaciji najboljih modela i generisanja korisnog znanja za eksperte iz domena ekonomije i energetske analize, a koje se može upotrebiti za donošenje boljih upravljačkih odluka kao i poboljšanje kvaliteta života, i dostizanje stabilnosti na globalnom tržištu.
- Koristiće se tehnike objašnjive veštačke inteligencije (*eng. Explainable AI, XAI*) za sticanje uvida u faktore koji utiču na cenu nafte.

Takođe, može se utvrditi da ova disertacija doprinosi više stepena naučnog nivoa:

- **Fundametalni (teorijski)** nivo, koji se odnosi na predlog i opis nove arhitekture neuronskih mreža i primena modifikovanog PSO algoritma u cilju predviđanja vremenskih serija i dobijanja što preciznijih rezultata.

- **Naučna klasifikacija**, koja se odnosi na metode višekriterijumske analize.

- **Naučno otkriće** se odnosi na razvoj nove metodologije primene neuronskih mreža i metaheurističke optimizacije sa jasno definisanim postupcima i evaluacijama.

- **Naučno objašnjenje** se odnosi na primenu predloženog modela na stvarnim podacima i izvođenje eksperimentálnih studija koje potvrđuju prednost novog pristupa u poređenju sa standardnim modelima i metodama koji su korišćeni u rešavanju istog problema.

Pored ovoga, možemo takođe pretpostaviti da je naučno istraživanje opravdano iz više razloga, među kojima se izdvajaju najrelevantniji:

- **Unapređenje oblasti predikcije vremenskih serija:** Ovim istraživanjem se ide korak napred ka poboljšanju i većoj inovativnosti na polju predviđanja vremenskih serija, pogotovu kada govorimo o cenama goriva. Ovime se rešavaju mnogi izazovi i problemi koji mogu nastati prilikom eksperimentisanja u ovim oblastima, a koji su nedovoljno dobro proučavani u dostupnoj literaturi.
- **Razvoj novih kombinovanih metoda:** Ovo istraživanje kombinuje dve napredne i vrlo popularne oblasti – neuronske mreže i metaheurističku optimizaciju, čime se značajno popularizuje pristup rešavanju određenih problema implementacijom hibridnih modela koji mogu biti primenjeni na širokom dijapazonu relevantnih podataka, u cilju dobijanja što boljih konačnih rezultata.
- **Bolje razumevanje optimizacije ovakvih modela:** Iako je ovo vrlo popularna oblast, i dalje postoji ograničen broj naučnih radova koji sistematski primenjuju metaheuristike u rešavanju različitih optimizacionih problema, a pogotovu kada je reč o predikciji vremenskih serija, koja je sama po sebi vrlo kompleksna oblast. Na osnovu ovoga se može zaključiti da je i sam potencijal primene ovakvih modela u rešavanju ovog problema nedovoljno istraživán. Stoga, ovaj rad ima za cilj da pomogne u boljem razumevanju korišćenih tehnika i dodatno ohrabri njihovu primenu konkretno u rešavanju problema predikcija cena goriva.

Sa druge strane, društvena opravdanosti istraživanja se može ogledati u sledećem:

- **Preciznije predikcije cena goriva:** Pomaže se potrošačima, kompanijama i donosiocima biznis odluka da bolje planiraju potrošnju, logistiku i strateške odluke zasnovane na pouzdanim podacima. Ovime se omogućava bolja usklađenost sa svetskim tržištem, kao i veća adaptivnost vrlo čestim promenama koje nastaju u okviru nekih spoljašnjih činioca koji značajno mogu uticati na cene dostupnih resursa goriva.
- **Podrška stabilnosti tržišta:** Automatizacija procesa predikcije cena goriva dovodi do efikasnijeg smanjenja neizvesnosti na globalnom tržištu energenata. Donošenjem pravovremenih odluka koje su bazirane na preciznim rezultatima, omogućava se mnogo bolje i efikasnije upravljanje resursima, kao i uspostavljanje stabilnije kontrole nad njima.
- **Primena u nekim drugim ekonomskim modelima:** Jedna od osnovnih ideja ovog istraživanje je da, između ostalog, pokaže i kako predloženi model koji je razvijen ne mora biti nužno ograničen samo na problem predikcije cena goriva. Kao što je u jednoj od prethodnih sekcija ovog rada napomenuto, jedan od ciljeva jeste i dovoljno dobro generalizovati predloženu arhitekturu, kako bi ona bila primenljiva u predikcijama i nekih drugih socioekonomskih vremenskih serija, kao što su predviđanje cena prirodnog gasa, električne energije, zatim analiza inflacionih troškova i mnogih drugih.

1.2 Struktura rada

Disertacija će se sastojati iz uvoda, šest poglavlja i zaključka.

U uvodnom delu doktorske disertacije ukratko će biti izložen problem vezan za predviđanje cena goriva, značaj tačnih prognoza, odnosno dobijenih rezultata za industriju i ekonomiju, te opisani izazovi u vezi sa dugoročnim i kratkoročnim predviđanjima vremenskih serija, koje ovaj rad ima za cilj da nadomesti. Takođe će biti razmatrani načini upotrebe modela Veštačke inteligencije, dodatno optimizovanih metaheurističkim algoritmom rojeva čestica, u rešavanju prethodno napomenutog problema. Na kraju biće predstavljeni osnovni ciljevi i hipoteze istraživanja, kao i metodološki pristup koji se koristi i njegova naučno-društvena opravdanost.

Prvo poglavlje biće posvećeno osnovnim aspektima predviđanja cena goriva, sa fokusom na ekonomske, političke i tehnološke faktore koji utiču na same cene. Biće istražene i dosadašnje tehnike koje se koriste za predviđanje cena goriva, kao što su neki standardni ekonometrijski modeli, statističke metode, ali moderni pristupi koji se baziraju na modelima mašinskog učenja.

Drugo poglavlje će dati detaljan pregled i analizu dostupne literature i načunih istraživanja rađenih u okviru oblasti predikcije cena goriva.

Treće poglavlje će predstaviti osnovne modele dubokog učenja, sa posebnim akcentom na rekurentne neuronske mreže (*eng. Recurrent Neural Networks, RNN*) i njihove podtipove, kao što su LSTM (*eng. Long Short-Term Memory*) i GRU (*eng. Gated Recurrent Unit*). Biće detaljno razmatrane matematičke osnove ovih modela, njihova primena u predviđanju vremenskih serija i prilagođavanje za specifične izazove predviđanja cena goriva. Takođe, biće istražen značaj podešavanja hiperparametara u optimizaciji ovih modela.

U četvrtom poglavlju biće prikazana taksonomija optimizacionih algoritama. U ovom poglavlju biće razmatrana primena optimizacije rojem čestica (*eng. Particle Swarm Optimization, PSO*) u kontekstu unapređenja performansi rekurentnih neuronskih mreža za predviđanje cena goriva. Detaljno će biti predstavljeni postupci integracije PSO sa RNN modelima, kao i kako ova kombinacija omogućava efikasnije učenje i predviđanje. Takođe, biće diskutovane prednosti i izazovi ovog pristupa u odnosu na standardne metode optimizacije.

U petom poglavlju, detaljnije će biti opisane i analizirane metaheuristike inteligencije rojeva u cilju predviđanja cena goriva. Biće prikazani konkretni eksperimenti sa setovima podataka o cenama goriva, metodologija obrade podataka, kao i implementacija RNN modela sa PSO za optimizaciju hiperparametara. Biće analizirani rezultati implementacije i poređeni sa tradicionalnim metodama.

U okviru šestog, eksperimentalnog, poglavlja biće prikazana evaluacija razvijenih modela korišćenjem različitih metrika performansi, kao što su tačnost predviđanja, greške u predviđanju, i stabilnost modela. Biće razmotren uticaj optimizacije rojem čestica na poboljšanje tačnosti modela u poređenju sa osnovnim rekurentnim mrežama bez optimizacije. Takođe, biće predstavljeni grafikoni i vizualizacije koji ilustruju dobijene rezultate. Dodatno će biti prikazana i interpretacija dobijenih ishoda, korišćenjem metoda poput SHAP/LIME, o kojoj će nešto kasnije biti više reči. Pozivajući se na čuvenu teoremu „nema besplatnog ručka“, u nastavku poglavlja biće predstavljeni i konkretni koraci za dalji razvoj modela i dodatno unapređenje njihovih performansi kroz dodatne optimizacije.

Na kraju disertacije biće iznet zaključak uz dodatnu diskusiju o ostalim mogućim pravcima razvoja i poboljšanja u ovoj oblasti.

2 Teorijska osnova istraživanja

2.1 Uvod i kratak istorijat korišćenja energenata

Pozivajući se na osnovu teoriju iz fizike, energiju možemo definisati kao nešto što se ne može niti stvoriti, niti uništiti, već se može transformisati iz jednog oblika u drugi. Danas je poznato da energija postoji u mnogim oblicima, od toplotne, električne pa sve do kinetičke i nuklearne, a takođe postoji jako puno načina kako se ona može transformisati u različite oblike i samim time koristiti za razne potrebe.

Poznato je da je čovečanstvo od nekog svog začetka bilo zavisno od određenih izvora energije. Ako posmatramo određenu vremensku liniju i vratimo se u praistorijski period, možemo primetiti da su prvi ljudi razumevanjem prirode i dostupnih resursa oko sebe započeli ciklus korišćenja istih u cilju osnovnog preživljavanja. Tako je došlo do otkrića vatre, koja je kasnije transformisana u prvi poznati izvor energije – toplote. Kasnije, u nekim ranim zapisima su ostali tragovi i o antičkoj Grčkoj, gde se prvi put koristila solarna energija, a nekih 200 godina pre toga u staroj Kini je prvi put otkriven ugalj, koji se i danas u velikoj meri koristi kao jedan od izvora energije. U nekim kasnijim godinama, na mestima različitih civilizacija i na različitim kontinentima, čovekovo upoznavanje sa prirodnim bogastvima sredine u kojoj se nalazio je dovelo i do upotrebe vetra kao još jednog izvora energije, postavljanjem prvih vetrenjača, a naravno u velikoj meri se koristilo i drvo. Pored ovih osnovnih prirodnih resursa, koristila se naravno i voda, a zanimljiv je i podatak da se na Novom Zelandu tokom 18. veka nalazila i prva hidroelektrana.

Idući dalje duž ove vremenske linije, dolazimo do sredine 18. veka u Americi, gde se po prvi put otkriva izvor nafte i gde kasnije nastaje prva bušotina. Ovo je naravno postalo neverovatno otkriće koje je otvorilo vrata mnogim mogućnostima i doprinelo tome da danas Amerika, pored Rusije, Ujedinjenih Arapskih Emirata i ostalih zemalja koje su bogate naftom, postane vodeća ekonomska sila i diktira mnoge socioekonomske faktore koje danas vidamo na tržištu. Nešto kasnije, tridesetih godina prošlog veka dolazi se i do otkrića nuklearne energije, što je otvorilo „Pandorinu kutiju“ i dovelo do masovne proizvodnje atomskih bombi, što se na žalost i danas održalo.

Današnji globalni energetski sistem se u velikoj meri zasniva na korišćenju fosilnih goriva. Prema nekim podacima, ugalj, nafta i gas predstavljaju skoro 80% globalne potražnje (Energy

Economics, Subhes C. Bhattacharyya). Razumevanje i kontrola ovih resursa predstavlja jako bitan faktor koji utiče na njihovu tržišnu cenu i dostupnost na globalnom planu. Imajući u vidu da je proces ekstrakcije, proizvodnje i transformacije energije u oblik koji se može koristiti jako dug i kompleksan, vrlo je bitno raditi sa ispravnim podacima i donositi pravovremene odluke.

Dakle, ako sagledamo istoriju, možemo zaključiti da je čovekova potreba za energijom jedna od konstanti koja u velikoj meri utiče na dinamiku proizvodnje, potrošnje i potencijalnog otkrivanja nekih novih izvora koji su možda više održivi i bolji za životnu sredinu.

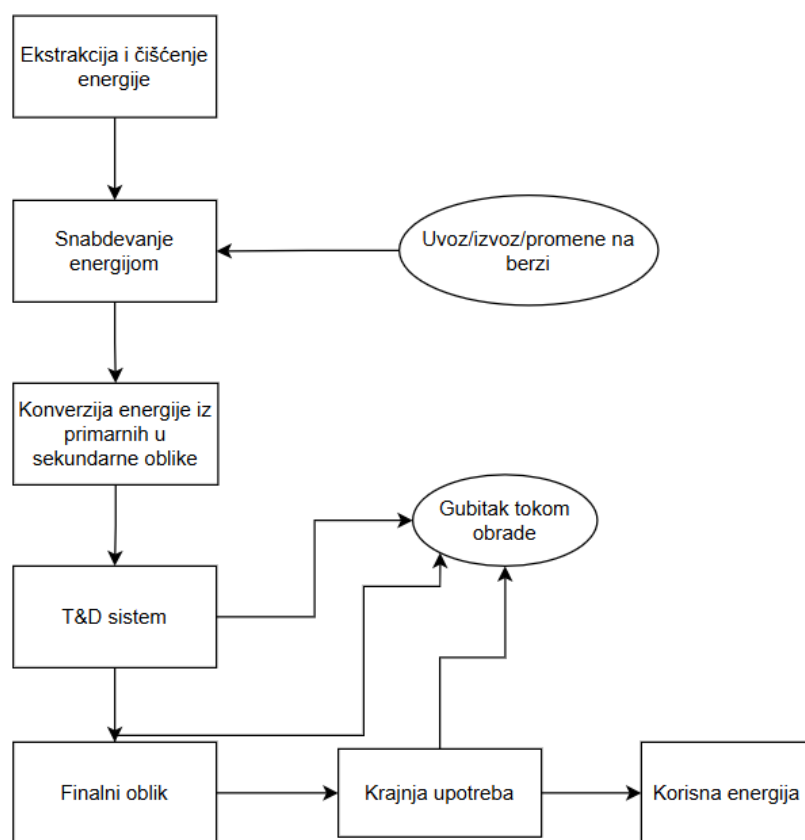
Uzimajući u obzir gore navedeno, kao i činjenicu da se energija može dobiti iz više različitih izvora, možemo je kategorisati na nekoliko načina:

1. **Primarni i sekundarni oblici energije** – ova klasifikacija podrazumeva način na koji se energija dobija. Primarni oblici podrazumevaju da se iz nekih osnovnih prirodnih resursa može izvući čista energija, kao što su recimo ugalj, Sunce i mnogi drugi. Sa druge strane, sekundarni oblici odnose na tip energije koji se može dobiti iz primarnog oblika uključujući neki proces transformacije ili konverzije (Energy Economics, Subhes C. Bhattacharyya).
2. **Obnovljivi i neobnovljivi tipovi energije** – ovaj tip podele se odnosi na to da li se energija dobija iz nekih izvora koji se u prirodi nalaze u nekim konačnim i ograničenim količinama, kao što su recimo nafta, ugalj, a koje nazivamo neobnovljivim izvorima. A obnovljivi tip podrazumeva dobijanje energije preko nekih izvora koje je nemoguće istrošiti, kao što su voda, vetar, solarna moć i mnogi drugi (Energy Economics, Subhes C. Bhattacharyya).
3. **Komercijalni i nekomercijalni tipovi energije** – da li je moguće trgovati ovim resursima na globalnom tržištu, gde komercijalni tipovi podrazumevaju naftu, ugalj, gas ili struju čija količina i moć na globalnom nivou na neki način diktiraju i njihovu cenu na tržištu, dok nekomercijalni tipovi podrazumevaju one kojima se ne može uspostaviti određena tržišna cena (Energy Economics, Subhes C. Bhattacharyya).

Ako sagledamo širu sliku, možemo zaključiti da svođenje energije na oblik koji je koristan za čoveka i njegove potrebe, podrazumeva širok spektr akcija i uključuje mnoge sektore, kao što su proizvodni sektor, sektor koji se bavi obradom i transformacijom energije kao i sektor koji se odnosi na krajnje potrošače (Energy Economics, Subhes C. Bhattacharyya) . Ovde naravno treba

dodati i finansijski sektor koji kontroliše i održava tržišnu cenu i prati promene koje nastaju kao rezultat određenih spoljašnjih faktora, a koje su u velikoj meri vidljive i na samoj berzi.

Takođe, treba uzeti u obzir i činjenicu da prilikom transformisanja energije iz nekog izvornog oblika u onaj koji je korisan za čoveka i njegove potrebe podrazumeva i određenje gubitke, koji mogu dodatno uticati na cenu i biti jedan od odlučujućih faktora za donošenje nekih odluka. Na slici ispod je dat okvirni prikaz ovog procesa i nekih akcija koje u okviru njega postoje.



Slika 1 Proces obrade energije

Razvojem modernih tehnologija, dovelo je i do modelovanja velikih informacionih sistema koji pomažu u donošenju dobrih odluka i razumevanju tržišta i količine energetske resursa sa kojima se danas raspolaze. Ova disertacija, u tom smislu, kao svoj cilj ima predstavljanje još jednog načina prikupljanja, obrade relevantnih podataka, kao i predviđanja novih šablona i promena, koristeći metode Veštačke inteligencije, o kojima će biti nešto više reči u narednoj sekciji. Osnovni cilj je definisati metodologiju kojom se mogu na neki način nadomestiti nedostaci

postojećih informacionih sistema, uzimajući u obzir i određene izazove koji se mogu javiti, kao što su pouzdanost, stabilnost, efikasnost i tačnost postojećih tehnika predviđanja.

2.2 Uticaj cena goriva na globalno tržište i ekonomiju

Razvoj modernog doba je sa sobom povukao brojna unapređenja u infrastrukturi i transportu, ali takođe i veliku potražnju za različitim energentima. Sam termin potražnja se može definisati na mnogo načina, a u nekom generalnom kontekstu možemo reći da je energetska potražnja „...bilo koji oblik energije koji se može iskoristi za ispunjavanje osnovnih potreba i zadataka, kao što su kuvanje, grejanje, transport i mnogi drugi...” (Energy Economics, Subhes C. Bhattacharyya). Ako sagledamo istoriju u nekom širem kontekstu, možemo zaključiti da je globalna ekonomija vrlo tesno povezana sa energetske sektorom i njegovom potražnjom, kao i to da mnogi spoljašni faktori mogu uticati na globalne tržišne cene energenata, čime danas možemo da svedočimo.

Iako je prethodno dat pregled različitih izvora energije i njihovog uticaja na globalnom tržištu, ovaj rad se primarno fokusira na gorivo, kao glavni energetski resurs, sa posebnim akcentom na to kako predikcija njegove cene može uticati na globalnu ekonomiju i tržište. Samo istraživanje je sprovedeno sa osnovnim ciljem da se na što precizniji način dobiju tačne predikcije cena gorive, čime se značajno olakšava donošenje pravovremenih odluka. U nastavku će biti dat kraći prikaz toga kako cene goriva mogu oscilirati i zašto je razumevanje ponašanja i strukture ovakvih podataka jedna od bitnih komponenata prilikom modelovanja pametnih i robusnih informacionih sistema.

Cene goriva, a naročito nafte imaju prilično dugu i kompleksnu istoriju, koja je tokom vremena bila pod stalnim uticajem geopolitičkih, socioekonomskih i tehnoloških promena širom sveta. Nakon njenog otkrića, tokom 20. veka, tržište nafte postaje ključni element svetske trgovine i energetike (Yergin, 2009). Samu industriju nafte su u više navrata tokom godina potresle brojne krize, među kojima je prva nastala kao posledica embarga OPEC-a, što je dovelo do skoro četvorostrukog povećanja cena nafte na globalnom nivou. Ovo je u značajnoj meri pogodilo ekonomiju i razvoj mnogih zemalja, od onih najsiromašnijih pa sve do najbogatijih, i bio prvi pokazatelj koliko je vodeća svetska ekonomija dosta ranjiva na tektonske poremećaje koji mogu nastati u cenama nafte i generalnom snabdevanju energenata (Hamilton, 1983). Brojni američki

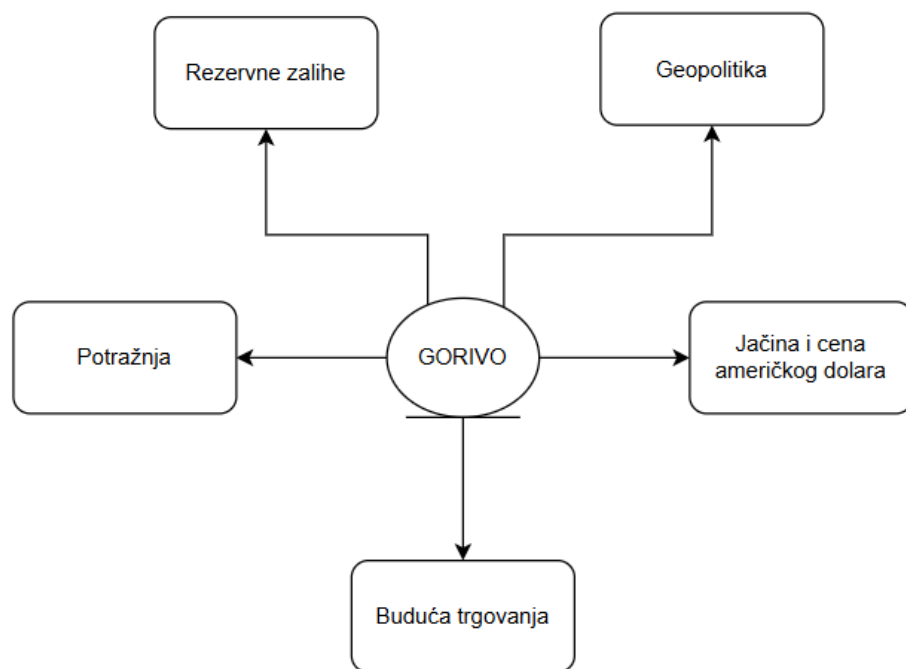
ratovi koji su se dešavali tokom sedamdesetih i devedesetih godina prošlog veka su takođe značajno destabilizovali tržište nafte i ostalih energenata. Ovaj trend se može, na žalost, i danas uočiti, jer je vođen brojnim međudržavnim konfliktima, poput onog između Rusije i Ukrajine, kao i nestabilnostima na Blistom istoku.

Pored militarnih konflikta, u obzir moramo uzeti i svetsku finansijsku krizu 2008. godine, koja je postala centralna tačka oko koje su u značajnoj meri oscilirale cene nafte. Pre same krize, njena cena je dostigla do tada rekordnih 147 dolara po barelu, dok je nakon nje pala čak ispod 40 dolara, što je takođe bila samo jedna od posledica recesije (IEA, 2009). U relativno skorijoj istoriji, pandemija COVID-19 je ubedljivo ostavila svoj najveći trag na cene i potražnju za naftom i ostalim gorivima, zbog izolovanosti brojnih zemalja i nemogućnosti transporta i putovanja. Ovo se takođe u značajnoj meri odrazilo i na logistiku transporta robe i ostalih dobara širom sveta, što je za posledicu uzrokovalo nestašicu određenih proizvoda.

Na osnovu prethodno navedenog, možemo zaključiti da praćenje cena goriva, konkretno nafte, nosi sa sobom veliki značaj, jer oscilacije u njenoj potražnji i ceni su samo jedni od faktora koji dosta utiču na trgovinske bilanse, budžete država i globalni ekonomski rast (Blanchard & Gali, 2007). Pored ovoga, česte promene u cenama mogu uticati i na troškove transporta i razvojne strategije mnogih država. Ovime su posebno pogođene države koje su vodeći neto izvoznici nafte i goriva, poput Norveške, Saudijske Arabije, koje direktno zavise od ovih cena kako bi održale svoju budžetsku stabilnost. Sa druge strane, države uvoznice poput Japana, Nemačke, Amerike postaju ranjive na oscilacije tržišta.

Postoji nekoliko ključnih razloga zbog kojih predviđanje cena goriva može postati izazovan zadatak:

1. **Geopolitika** – brojne političke tenzije, sankcije i sukobi mogu imati dugoročan i nepredvidiv uticaj na cene nafte (Baumer & Kilian, 2016)
2. **Promenljiva ponuda i potražnja** – s obzirom na procenat zagađenja životne sredine, u poslednjim godinama se sve veći akcenat stavlja na korišćenju obnovljivih izvora energije i smanjenju emisija štetnih gasova, pre svega ugljen-dioksida (IEA, 2023)
3. **Makroekonomski šokovi** – ekonomske krize koje mogu nastati na globalnom nivou, monetarne politike centralnih banaka i česte fluktuacije deviznih kurseva mogu takođe ostaviti svoj trag na česte promene u cenama goriva (Kilian & Murphy, 2014)



Ilustracija 2 Faktori koji mogu uticati na cenu goriva

Uzimajući u obzir prethodno navedeno, možemo zaključiti da je predviđanje cena goriva važan zadatak koji može dosta olakšati poslovanje kompanija i održati finansijsku stabilnost zemalja. Stoga njegov značaj možemo podeliti u dve grupe: državni i kompanijski značaji.

Ako govorimo o tome kakav značaj predikcija cena goriva može ostvariti na države, pogotovu one koje direktno zavise od nekih već pretpostavljenih cena nafte, možemo zaključiti sledeće:

- Ostvaruje se preciznije budžetiranje – zemlje koje direktno zavise od cena nafte, mogu trpeti jako velike gubitke ako se njihove pretpostavke ispostave pogrešnim ili ako zemlje poput Rusije ili Saudijske Arabije koje zvanično i određuju budžetsku cenu nafte počnu da diktiraju neke druge trendove
- Bolje planiranje monetarne politike – inflatorni pritisci koji su povezani sa cenama goriva mogu značajno uticati na kamatne stope i valutne kurseve
- Definisanje bolje energetske strategije – zemlje koje teže boljoj i sigurnijoj energetskej budućnosti koriste razne prediktivne metodologije kako bi vršile ulaganja u obnovljive izvore energije i time popularizovale ovaj proces (ovo se

posebno odnosi na zemlje EU, koje jednim delom žele i da preseku ili značajno umanje svoju energetska zavisnost od drugih država poput Rusije ili Emirata)

Ako govorimo o kompanijama, značaj predikcija cena goriva takođe može pozitivno uticati na određene segmente poslovanja i poslovne procese, uključujući i dodatne prednosti:

- Bolje strategije hedžovanja – korišćenjem određenih opcija vodećih hedž fondova (*eng. hedge funds*), kompanije se mogu zaštititi od rizika volatilnosti cena goriva
- Preciznija optimizacija troškova – preciznija očekivanja direktno dovode i do bolje optimizacije troškova i boljeg planiranja proizvodnje i transporta
- Bolje investicione odluke – planiranje novih projekata vodećih kompanija u velikoj meri zavisi i od dugoročnih predikcija cena goriva i njihove fluktuacije

Na osnovu određenih činjenica, kao i istorijskog aspekta možemo izvesti generalni zaključak da neuspeh ili nemogućnost preciznog predviđanja cena goriva, kao i nerazumevanje uticaja koje one mogu imati na ekonomiju i biznis, može u velikoj meri ostaviti brojne posledice i gubitke kako na mikroekonomskom tako i na makroekonomskom nivou.

2.3 Primena Veštačke inteligencije na predikciju cena goriva

Tokom dugog niza godina, pre pojave Veštačke inteligencije, dugo su korišćeni statistički modeli predikcije cena goriva. Tradicionalni statistički modeli, kao što su recimo ARIMA (*eng. Autoregressive Integrated Moving Average*) ili VAR (*eng. Vector Autoregression*), o kojima će u kasnijem poglavlju biti nešto više reči, su jako dugo korišćeni za predikciju cena goriva i nafte. Oni se oslanjaju na korišćenje vremenskih serija, ali polaze od pretpostavke da postoji stabilnost u podacima, te se tako oni mogu efikasno koristiti u nekim uslovima relativno predvidivih i stabilnih tržišnih kretanja. Ovo u isto vreme predstavlja i njihov glavni nedostatak, jer zbog ove početne pretpostavke ne mogu predvideti neke nagle šokove i promene na globalnom tržištu, te je njihova primena u ovakvim situacijama vrlo često nepouzdana i dosta ograničena.

Sa druge strane, razvojem Veštačke inteligencije i modela dubokog učenja, kao što su posebne arhitekture neuronskih mreža sa dugoročno-kratkoročnom memorijom, odnosno LSTM modeli (*eng. Long Short-Term Memory, LSTM*) koji će biti nešto detaljnije objašnjeni u ovoj disertaciji, ne polaze od pretpostavke da postoji određena stabilnost među promenljivama i

ulaznim podacima, te su tako pokazali značajni istraživački potencijal jer mogu predvideti kompleksnije zavisnosti među podacima, kao i neke određene nelinearne obrasce koji se mogu javiti. Jedna od glavnih prednosti ovih modela je i mogućnost pamćenja dužih perioda promena koje nastaju u podacima, zbog čega su i dobili tako specifičan naziv. Zahvaljujući ovim karakteristikama mogu dosta bolje reagovati na često promenljive trendove, a takođe pokazuju obećavajući potencijal u predikciji nekih potencijelnih šokova i promena na tržištu.

Međutim kako i ovi modeli imaju svoje nedostatke, daljim istraživanjima se došlo do neke „zlatne sredine“ u okviru koje su se pojavili takozvani hibridni modeli, koji kombinuju određene statističke pristupe, koji su recimo prisutni kod ARIMA metodologija i Veštačku inteligenciju, kako bi sa što većom tačnošću mogli da prediktuju određene promene na tržištu. Ovime se nadomešćuju neki glavni nedostaci klasičnih statističkih modela, kao što je nepouzdanost rezultata njihove primene u situacijama koje su praćene brojnim oscilacijama u samim podacima. Pored ovoga se takođe prevazilaze i neki nedostaci modela Veštačke inteligencije, kao što su složenost, potreba za velikom količinom podataka i računarskim resursima. Kako ovi modeli nisu dovoljno testirani u praksi, ostavlja se veliki prostor za njihovo unapređenje i dodatno eksperimentisanje.

Da bi se što bolje sagledao potencijal koji primena Veštačke inteligencije ostvaruje na ovom polju potrebno je razumeti njenu istoriju i modele koji se smatraju začetnicima današnjih hibridnih arhitektura koje dobijaju na sve većoj popularnosti na naučno-istraživačkoj sceni.

2.3.1 Kratak istorijat veštačke inteligencije

Iako deluje kao relativno nova oblast, Veštačka inteligencija zapravo svoje korene vuče još u dalekoj prošlosti. Naime, istorijski gledano još u antičkim vremenima se kroz razne priče i mitove proteže ideja o nekim nestvarnim, veštačkim bićima sa natprirodnom inteligencijom, što se dalje protezalo kroz zapise koje su ostavile mnoge civilizacije. U nešto kasnijem, modernom dobu ova ideja nije potpuno nestala već se pretvorila u glavno pitanje: Da li je moguće napraviti mašinu koja može da misli kao čovek? Nezvanično, prvi konkretni radovi na polju Veštačke inteligencije započinju tridesetih godina prošlog veka, radom poznatog matematičara Alana Tjuringa. Njegova fascinacija idejom da mašine mogu postati dovoljno inteligentne da razmišljaju kao ljudi bila je daleko ispred svog vremena. Ovome svedoči i jedan od Tjuringovih kolega iz Blečli Parka, Donald Miš, koji se više puta prisećao Tjuringa koji je govorio o tome da je sasvim moguće da računari uče iz svog iskustva, kao i da mogu rešavati neke nove probleme sa kojima

nisu prethodno bilo suočeni, ukoliko im se definišu dovoljno dobro instrukcije vodilje – tehnika koja je danas poznata kao heuristično rešavanje problema, o čemu će više reći biti u kasnijim poglavljima. Tokom četrdesetih godina prošlog veka, Tjuring ima svoje prvo predavanje u Londonu, gde govori: „...*Ono što želimo je mašina koja može učiti iz svog iskustva, a mogućnost mašine da menja svoje instrukcije predstavlja osnovni mehanizam za ovo...*“ (History of AI, B.J. Copeland 2025).

Tjuringov rad i doprinos tokom Drugog svetskog rata je ostavio jako veliki uticaj na današnje moderno sagledavanje pojmova računarstva i Veštačke inteligencije. Uzimajući u obzir njegove ideje i pretpostavke, Veštačku inteligenciju možemo definisati i u nekom formalnom obliku kao oblast koja nastoji da stvori mašine sposobne za obavljanje zadataka koje bi inače zahtevale biološku, ljudsku inteligenciju, kao što su učenje, rešavanje problema, racionalno rasuđivanje i donošenje odluka.

Zvaničnim začetkom Veštačke inteligencije (*eng. Artificial Intelligence – AI*), se smatra Darmut konferencija (*eng. Dartmouth conference*) koja se odigrala 1956. godine. U okviru nje su Džon Mekarti (*eng. John McCarthy*), Marvin Minski (*eng. Marvin Minsky*), Nataniel Ročester (*eng. Nathaniel Rochester*) i Klod Šenon (*eng. Claude Shannon*) izneli zvanični predlog za prvo sistematsko istraživanje mogućnosti stvaranja dovoljno inteligentih mašina, koje mogu razviti sposobnost učenja i racionalnog razmišljanja, odnosno rasuđivanja. Kasnije, tokom šezdesetih godina, istraživanje na polju Veštačke inteligencije dobija na sve većem značaju, i naučnici uspevaju da postignu dosta dobre uspehe u razvoju algoritama za rešavanje problema i logičkih igara, poput šaha. Međutim glavni problem koji je u određenoj meri dosta kočio i usporio istraživanja bili su značajno ograničeni računarski i hardverski resursi koji su postojali u to vreme. Pored ograničenih resursa, brojni eksterni i ekonomski faktori su doveli do smanjenja interesovanja za daljim proučavanjem Veštačke inteligencije. Ovo je dovelo do toga da ona, kao oblast, u periodu od nekih 20tak godina, tačnije između kraja sedamdesetih i početka devedesetih potpuno zamre, što je još poznato i kao „zima AI-ja“ (Crevier, 1993).

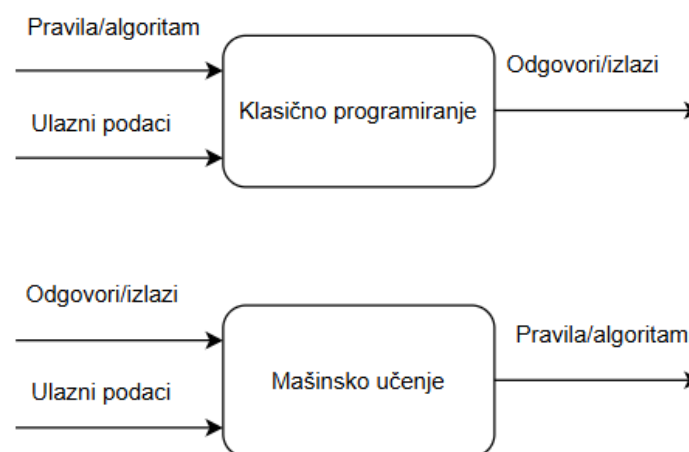
Novi talas razvoja Veštačke inteligencije započinje kasnih 90tih, kada se takođe značajno unapređuju i računarski resursi. Ponovno interesovanje za istraživanje u ovoj oblasti dovodi i do definisanja prvih podoblasti kao što su Mašinsko učenje, a kasnije i Duboko učenje. Ključne prekretnice u svetu Veštačke inteligencije definitivno obuhvataju otkriće i razvoj konvolutivnih neuronskih mreža (LeCun et al., 1998), koje se danas u velikoj meri koriste u kompjuterskom vidu

(eng. *Computer Vision, CV*). Pored ovoga ključno je postizanje skoro ljudskih performansi u zadacima kao što su prepoznavanje govora i slike (LeCun et al., 2015).

Zahvaljujući brojnim postignućima na ovim poljima, možemo zaključiti da se Veštačka inteligencija kao jedna široka oblast bazira na vrlo stabilnim osnovama, kao i to da ona eksponencijalnim poboljšanjem hardverskih resursa takođe dobija značajan rast i omogućava definisanje dosta naprednijih algoritama koji se široko primenjuju u mnogim oblastima.

2.3.2 Često korišćeni modeli

Ako postavimo paralelu sa tradicionalnim programiranjem, možemo uočiti značajne napretke koje Veštačka inteligencija ostvaruje u rešavanju nekih kompleksnijih problema. Naime, u običnom programiranju, možemo videti da ono funkcioniše po vrlo jednostavnom principu: ako postoji neki ulaz x , izvrši neku operaciju y . Dakle programiranje možemo u nekom formalnom obliku definisati kao set nekih instrukcija ili koraka, koje računaru govore kako da reše određeni problem, odnosno kako da na osnovu nekog ulaza x izvrše određenu operaciju y . Ovo zahteva da programeri ručno definišu logiku i ponašanje samog sistema, što znači da moraju potencijalno predvideti sva moguća scenarija i situacije. Ovaj pristup je efektivno koristan kada govorimo o nekom vrlo jednostavnom problemu, koji ima jasno definisani cilj, zahteve i koji nema slobodu stvaranja nekih neočekivanih situacija ili scenarija.



Ilustracija 3 Prikaz razlika između programiranja i mašinskog učenja

Međutim, kada govorimo o daleko kompleksnijim problemima, za koje prosto nije moguće da čovek rezonuje svaki mogući scenario, i koji se u glavnom baziraju na vrlo nestabilnim ulaznim skupovima x koji ne garantuju konzistentnost i uvek očekivane rezultate, potrebno je definisati drugačije principe za rešavanje ovog problema. U ovakvim situacijama se praktično koriste modeli Veštačke inteligencije, koji se oslanjaju na učenje iz primera. Dakle, umesto da se ručno kodiraju sva moguća pravila, koje za neke izuzetno kompleksne i nepredvidive probleme nekad čak i nije moguće apsolutno definisati, ovi modeli se treniraju na velikim skupovima podataka, prepoznajući obrasce u njima i određene odnose koje nije moguće uvek eksplicitno programirati (Goodfellow et al., 2016).

Tabela 1 Paralela između tradicionalnog programiranja i Veštačke inteligencije

Tradicionalno programiranje	Veštačka inteligencija
Eksplicitna pravila definisana programom	Pravila i obrasci se “uče” iz podataka
Neprilagođenost u uslovima promena	Sposobnost adaptacije na nove situacije i šablone
Primenljivo na precizno definisanim problemima	Primenljivo na kompleksnim i nestruktuiranim problemima
Smanjena skalabilnost u često promenljivim okruženjima	Skalabilnost i sposobnost za “samosavršavanje”

Imajući ovo u vidu, razvojem ove oblasti došlo je do popularizacije brojnih modela koji ispunjavaju karakteristike prikazane u tabeli 1. Dakle, većina modela Veštačke inteligencije podrazumeva specifične arhitekture koje omogućavaju rešavanje određenih kompleksnih problema, a koje mogu “učiti” iz ulaznih podataka i biti adaptivne na promene u njima. Međutim, generalno govoreći ne postoji “najbolji model” koji može rešiti apsolutno svaki problem, već izbor najoptimalnijeg rešenja treba da zavisi prvenstveno od prirode podataka, dostupnih resursa kao i kompleksnosti samog problema koji pokušavamo da rešimo. S obzirom da je glavni focus ove disertacije primena neuronskih mreža u cilju predviđanja cena goriva, u nastavku je dat kraći pregled često korišćenih arhitektura ovih mreža, kao i analiza njihovih glavnih prednosti i nedostataka.

1. Neuronske mreže (*eng. Neural Networks – NN*)

Ovo su definitivno jedni od najpopularnijih modela koji su svoju široku primenu pronašli u rešavanju najrazličitijeg dijapazona problema. Kako i sam naziv kaže, ovi modeli su inspirisani biološkim neuronskim mrežama i načinu kako one prirodno funkcionišu, odnosno načinima kako biološki mozak obrađuje informacije. Kao i kod prirodnih neuronskih mreža, tako i kod veštačkih je osnovna jedinica funkcionisanja su neuroni koji su organizovani u nekoliko slojeva. Ulazni podaci se transformišu i putuju kroz slojeve mreže, pri čemu se mogu dalje modelovati i razumeti neki kompleksniji obrasci u njima.

2. Duboke neuronske mreže (*eng. Deep Neural Networks – DNN*) i duboko učenje

Duboke neuronske mreže predstavljaju zapravo podvrstu klasičnih neuronskih mreža, gde imamo više sakrivenih slojeva (*eng. Hidden layers*). Prisustvo više sakrivenih slojeva čini njihovu arhitekturu složenijom, te su i zato dobile tako specifičan naziv. Ovi slojevi omogućavaju mrežama da otkriju i bolje razumeju kompleksne obrasce i šablone koji se mogu javiti u podacima, obrađujući i transformišući ulazne podatke kroz više faza. Na ovaj način, duboke neuronske mreže mogu čak da razumeju i identifikuju neke hijarhijske šablone i odnose u podacima. Ovi šabloni se u glavnom kreću od nekih jednostavnijih na nižim nivoima hijerarhije, do nekih kompleksnijih i više apstraktnih koncepata na višim nivoima.

3. Konvolutivne neuronske mreže (*eng. Convolutional Neural Networks – CNN*)

Konvolutivne neuronske mreže takođe predstavljaju specijalnu vrstu, odnosno podtip klasičnih neuronskih mreža koje se danas u velikoj meri koriste za obradu podataka koji se u računarima prikazuju u vidu matrica, kao što su recimo slike, ili nekog drugog multimedijalnog sadržaja. Ključna karakteristika ovih modela je upotreba takozvanih konvolutivnih filtera (*eng. convolutional filters*), koji omogućavaju ekstrakciju bitnih karakteristika iz podataka na automatizovan način, a po kojima su i dobile svoj naziv. Primenom ovih filtera, model može postepeno da prepozna sve kompleksnije strukture, od osnovnih ivica, pa sve do složenijih primera kao što su lica, slova, brojevi ili neki predmeti.

4. Rekurentne neuronske mreže (*eng. Recurrent Neural Networks*)

Ovi modeli se najviše koriste za obradu sekvencijalnih podataka, tako što se vrlo često kombinuju sa takozvanim modelima dugoročno-kratkoročne memorije (*eng. Long term-short term memory*), čime se rešava problem dugoročnih zavisnosti ili eksplozije gradijenata. Sekvencijalni podaci podrazumevaju primere poput teksta, govora, vremenskih serija ili nekih kontinualnih signala. Glavna prednost ovih modela leži u činjenici da oni ove podatke obrađuju sukcesivno, i samim time sva nova stanja zavise od izlaza iz prethodnih slojeva, i na taj način model polako „pamti“ određene obrasce kroz vreme.

5. Generativne adversarijalne mreže (*eng. Generative Adversarial Networks – GAN*)

Relativno novi modeli koji zapravo predstavljaju kombinaciju dva modela (generatora i diskriminatora), koji se međusobno takmiče, čime omogućavaju generisanje novih, realističnih podataka. Osnovni zadatak generatora je da na osnovu nekog ulaznog šuma stvara nove podatke, dok diskriminator u isto vreme pokušava da što preciznije razluči šta su pravi podaci a šta generisani. Ovakvi modeli su našli široku upotrebu u procesu generisanja slika, poboljšanju kvaliteta drugih multimedijalnih sadržaja, kao i u generisanju novih sintetičkih podataka koji se koriste za treniranje nekih drugih modela.

6. Transformer modeli (*eng. Transformers*)

Popularizacijom raznih alata na bazi Veštačke inteligencije, kao što su brojni chatbotovi, Gemini, ChatGPT i mnogi drugi, došlo je do razvoja i takozvanih transformera, koji koriste takozvani mehanizam pažnje (*eng. Attention mechanism*) za modelovanje i razumevanje odnosa između sekvencijalnih podataka, bez obzira na međusobnu udaljenost tih podataka u jednoj sekvenci. Ovi modeli u značajnoj meri nadomešćuju nedostatke koje imaju klasične rekurentne i generativne neuronske mreže, time što mogu bolje da modeluju neke kompleksnije obrasce.

Tabela 2 Prikaz prednosti i nedostaka ovih modela

Model	Prednosti	Nedostaci	Ključne primene
ANN	Jednostavni za impementaciju	Lako updaju u lokalne minimume	Klasifikacija i osnovni oblici regresije
DNN	Visoka preciznost kod kompleksnijih zadataka	Zahtevaju veliku količinu podataka i hardverske resurse	Prepoznavanje govora, slike
CNN	Efikasna ekstrakcija bitnih karakteristika iz slika	Nisu primenljive nad sekvencijalnim podacima	Kompjuterski vid, dijagnostika
RNN/LSTM	Dobri za obradu velike količine sekvencijalnih podataka	Problemi sa dugoročnim zavisnostima	Obrada prirodnih jezika (eng. Natural Language Processing – NLP)
GAN	Generisanje novih, realnih podataka	Teško se treniraju i teže ka lošijoj konvergenciji	Generisanje slika, augmentacija
Transformeri	Odlična primena kod sekvencijalnih podataka kod kojih treba pamtit i dugoročne odnose	Zahtevaju velike količine podataka, dosta vremena i računarskih resursa za treniranje	NLP, chatbotovi, tekstualna analiza

Svi prethodno opisani tradicionalni modeli su tako dizajnirani da se mogu primeniti nad jednim tipom podataka – tekstom, slikom ili zvukom. Iz prethodne tabele možemo izvući zaključak da recimo konvolutivne neuronske mreže mogu jako dobro raditi sa vizuelnim podacima poput slika, ali su zato vrlo često pokazale lošije rezultate i preciznost generalizacije kada se koriste nad određenim sekvencijalnim podacima. Kako bi se nadomestili određeni nedostaci ovih tradicionalnih modela, došlo je do razvoja nove generacije modernih arhitektura koji imaju sposobnost obrade više različitih tipova podataka, na primer tekstova i slika. Ovo je poznato pod

drugim nazivom kao multimodalnost. Mnogi alati koje danas vrlo često koristimo u svakodnevnom radu a koji se baziraju na recimo GPT-4o modelu, se koriste upravo za obradu i teksta i slika i zvuka u okviru jedinstvenog arhitekturnog rešenja. Ovime se značajno poboljšava i proširuje njihova primena i efikasnost.

Pored multimodalnih modela, razvila se potreba i za individualnim pristupima Veštačke inteligencije specifičnim potrebama ili uslugama koje se pružaju krajnjim korisnicima, što je za posledicu imalo razvoj takozvane Personalizovane veštačke inteligencije. Korišćenjem Personalizovane veštačke inteligencije se značajno poboljšava komunikacija sa korisnikom koji je koristi.

Kako je opšte poznato da svi ovi napredni modeli zahtevaju dosta hardverskih resursa i kompjuterske snage, javila se tendencija ka mogućnosti razvoja manjih i efikasnijih modela Veštačke inteligencije, koji se mogu testirati i na uređajima sa ograničenim hardverskim resursima, ali koji i dalje mogu ostvariti dovoljno dobre performanse i veliku preciznost. Međutim ono što trenutno deluje da je konačan cilj kada govorimo o kontinuiranom razvoju modela Veštačke inteligencije, jeste zapravo postizanje onoga što je i Tjuring tokom svog vremena govorio i predvideo, pravljenje mašine koja može biti dovoljno inteligentna da obavi bilo koji kognitivni zadatak koji bi inače zahtevao neki ljudski resurs. Ovo je dovelo do istraživanja i pojave relativno nove podoblasti Veštačke inteligencije, poznate kao Generalna veštačka inteligencija (*eng. Artificial General Intelligence – AGI*).

Dakle, ako ispratimo trendove koji su trenutno aktuelni, kao i neke generalne pravce u okviru kojih napreduje sam razvoj ovih modela, možemo zaključiti da će se u budućnosti u velikoj meri težiti nekoj većoj efikasnosti prilikom razvoja i korišćenja ovih modela, kao i većoj i boljoj mogućnosti generalizacije na različitim vrstama podataka. Ovde takođe možemo razmatrati i pitanje toga kako se ovi trendovi odražavaju i na samu budućnost predikcije vremenskih serija, konkretno u ovom slučaju cena goriva. U cilju boljeg razumevanja ovog pitanja, u nastavku će biti dat opis vremenskih serija i metoda koji se koriste za njihovu predikciju.

2.3.3 Vremenske serije

Pojam vremenskih serija možemo u nekim formalnim okvirima definisati kao niz nekih vrednosti određene promenljive koja se posmatra kroz određeni vremenski okvir i koje su zabeležene u pravilnim intervalima. Analiza ovakvih vremenskih serija omogućava prepoznavanje različitih obrazaca i trendova među njima, kao i predviđanje njihovih budućih vrednosti ili nekih potencijalnih anomalija. Konkretno, u istraživanju sprovedenom kao deo ove disertacije, kao primer vremenskih serija su korišćene cene goriva, koje su date u obliku istorijskih podataka koji su vremenom beleženi u stabilnim intervalima. Kao što je ranije napomenuto, analiza ovih podataka omogućava bolje razumevanje promenljivosti samog tržišta i toga kako se određene oscilacije u ovim podacima mogu odraziti na generisane ishode, a samim tim i na neke strateške odluke koje se donose.

Kako bi se što bolje razumele prednosti primene modela Veštačke inteligencije na ovim podacima u cilju njihove predikcije, potrebno je sagledati i neke stohastičke pristupe koji su ranije korišćeni u okviru istog cilja. S toga će u nastavku biti dat prikaz nekih od poznatijih i često korišćenih stohastičkih modela.

2.3.3.1 Pregled stohastičkih metodologija za predikciju vremenskih serija

Istorijski gledano, možemo reći da su neki prvi oblici istraživanja i analize vremenskih serija začeti još sa prvom pojavom nekih klasičnih, stohastičkih metodologija tridesetih godina prošlog veka. Međutim u ovom periodu su ove ideje još uvek postojale na nekom osnovnom, konceptualnom nivou, a prvo konkretno modelovanje vremenskih serija počinje šezdesetih i traje tokom sedamdesetih godina prošlog veka, gde pronalazi široku primenu pre svega u ekonomiji a potom i u inženjerstvu. Prva metodologija koja je uspostavljena tokom ovog procesa je takozvana Box-Jenkins ARIMA metodologija (Box & Jenkins, 1970) i ona postaje standard za modelovanje stacionarnih vremenskih serija. U periodima između osamdesetih i devedesetih godina razvijene su nešto specifičnije metodologije koje su manje-više predstavljale nastavak originalne ARIMA metodologije, sa ciljem da se nadomeste njeni nedostaci, kao što su SARIMA, GARCH i VAR (koja se u velikoj meri koristila za analizu višedimenzionalnih vremenskih serija). Glavni fokus ovih metodologija je bio na definisanju i modelovanju postojećih linearnih zavisnosti u podacima.

2.3.3.1.1 Auto-Regresivni (AR) modeli

Ovi modeli predstavljaju prve začetke analize vremenskih serija. Kod Auto-Regresivnih modela, vrednost promenljive koja se trenutno posmatra se modeluje kao linerana kombinacija njenih prošlih vrednosti (Box & Jenkins, 1970). Drugim rečima, kod AR modela, buduće ponašanje neke varijable se prediktuje na osnovu prethodnog ponašanja. Teorijski gledano ovi modeli su jako slični klasičnoj linerarnoj regresiji u smislu da su rezultati Y direktno zavisni od ulaznih promenljivih, odnosno prediktora X . Ključna razlika između AR modela i klasične linearne regresije se ogleda u tome da je Y pored toga što je zavisno od prediktora X , takođe zavisno i od svojih prethodnih vrednosti. Ovi modeli su prvi put primenjeni u analizi cena energenata kako bi se što preciznije definisale kratkoročne dinamike globalnog tržišta (Box & Jenkins, 1970).

Matematički prikaz ovih modela je definisan sledećom formulom:

$$y_t = \delta + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \dots + \varphi_p y_{t-p} + A_t$$

Jednačina 1 AR model

Gde su:

- y_t – vrednost vremenske serije u trenutku t
- δ – konstanta
- $\varphi_1 y_{t-1}$ – koeficijenti autoregresije
- y_{t-1} – prethodne vrednosti vremenske serije
- A_t – stohastički termin (beli šum)

Primena auto-regresivnih modela je u značajnoj meri poboljšala identifikaciju nekih ključnih obrazaca i trendova u fluktuaciji cena goriva. Rezultati dobijeni nakon primene ovih modela su imali veliki doprinos u boljem planiranju proizvodnje i trgovinske strategije.

2.3.3.1.2 Pokretni Prosek (MA) modeli

Modeli pod nazivom pokretni prosek (*eng. Moving Average – MA*) se nastavljaju na prethodnu grupu modela, s tim da oni modeluju trenutnu vrednost kao funkciju prethodnih slučajnih grešaka u sistemu (Box & Jenkins, 1970). Obično se primenjuju u ekonomiji kako bi se prediktovali određeni trendovi tržišta i bolje razumeli obrasci u istorijskim podacima.

Matematička formulacija ovih modela se ogleda u sledećoj formuli:

$$X_t = c + \epsilon_t + \theta_1 \cdot \epsilon_{t-1} + \theta_2 \cdot \epsilon_{t-2} + \dots + \theta_q \cdot \epsilon_{t-q}$$

Jednačina 2 MA model

Gde su:

- X_t – trenutna vrednost vremenske serije
- c – konstanta
- ϵ_t – slučajna greška (beli šum)
- θ_i – parametri MA modela
- q – red modela

Za razliku od klasični AR modela, MA modeli su svoju široku primenu pronašli u situacijama kada je bilo potrebno predvideti određene neočekivane šokove na tržištu, što je bila vrlo česta pojava koja je predstavljala posledicu brojnih geopolitičkih konflikata, a koja je značajno uticala na često neočekivano skakanje cena goriva na globalnom tržištu.

2.3.3.1.3 Kombinovani ARMA modeli

Jedan od glavnih nedostataka MA modela koji se može uočiti je taj što se oni baziraju na pretpostavci da su podaci nad kojima se primenjuje stacionarni, odnosno da se njihove karakteristike ne menjaju tokom vremena. U realnom okruženju, pronaći ovakav skup istorijskih podataka koji nisu promenljivi tokom dužeg vremenskog perioda je gotovo nemoguć zadatak, te kako bi se nadomestio ovaj nedostatak, definisana je verzija modela koja kombinuje prethodno opisane AR i MA pristupe. Ova nova arhitektura je poznata kao kombinovani ARMA model (*eng. Autoregressive Moving Average*), koji je idealan za podatke koji su stacionarni, a takođe se

ostvaruje i mogućnost predikcije moguće inercije i šokova u vremenskim serijama (Box & Jenkins, 1970).

Formalno, ARMA model se može prikazati sledećom matematičkom formulom:

$$Y_t = c + \varphi_1 Y_{t-1} + \dots + \varphi_p Y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q}$$

Jednačina 3 ARMA model

Gde su:

- Y_t – trenutna vrednost vremenske serije u trenutku t
- c – konstanta
- $\varphi_1, \varphi_2, \dots, \varphi_p$ – koeficijenti autoregresivne komponente
- p – red AR komponente, broj prethodnih vremenskih tačaka koje utiču na Y_t
- ε_t – greška, odnosno neočekivani šok u vremenskoj seriji
- $\theta_1, \theta_2, \dots, \theta_q$ – koeficijenti pokretnog proseka
- q – red MA komponente, broj prethodnih grešaka koje utiču na Y_t

2.3.3.1.4 Nelinearni modeli (ARIMA I SARIMA)

Kako su mnogi analitičari uočili da većina vremenskih serija realno ne mogu uvek biti stacionarne, pogotovu cene goriva ili nafte koje su vrlo često podložne neočekivanim šokovima i promenama, prvobitno je uveden koncept takozvanih ARIMA modela (*eng. Autoregressive Integrated Moving Average*), kao generalizacija klasičnog ARMA modela. Ovo su modeli koji kombinuju nekoliko ključnih komponenti, kao što su autoregresija (*eng. Autoregression – AR*), koja, kao što je prethodno objašnjeno, definiše kako prošle vrednosti određene promenljive mogu uticati na njene buduće vrednosti, zatim razlikovanje podataka, koje nadomešćuje početnu pretpostavku u stacionarnosti podataka koja je vezana za mnoge standardne linearne modele i na kraju pokretni prosek (*eng. Moving Average – MA*), koji u obzir uzima i određene greške u sistemu. Konkretno na primeru cena goriva, ARIMA model ima mogućnost da predviti buduće cene goriva, na osnovu prethodnih cena, kao serije istorijskih podataka, ali u okviru svojih

predikcija uključuje i mnoge faktore kao što su promenljivi trendovi, neočekivani šokovi, uticaj greške na tačnost predikcija i mnoge druge.

Matematička formula opisanog modela je prikazana na sledeći način:

$$Y_t = c + \varphi_1 Y_{t-1} + \varphi_2 Y_{t-2} + \dots + \varphi_p Y_{t-p} + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

Jednačina 4 ARIMA I SARIMA model

Gde su:

- Y_t – trenutna vrednost vremenske serije u trenutku t
- c – konstanta
- $\varphi_1, \varphi_2, \dots, \varphi_p$ – koeficijenti autoregresivne komponente
- p – red AR komponente, broj prethodnih vremenskih tačaka koje utiču na Y_t
- ε_t – greška, odnosno neočekivani šok u vremenskoj seriji
- $\theta_1, \theta_2, \dots, \theta_q$ - koeficijenti pokretnog proseka
- q – red MA komponente, broj prethodnih grešaka koje utiču na Y_t
- d - broj razlikovanja (engl. differencing) koji se primenjuje na vremensku seriju da bi postala stacionarna

Međutim, ovaj model takođe pokazuje određene nedostatke. Naime, kada govorimo u predikciji cena goriva, zbog prirode samih podataka, treba uzeti u obzir i njihovu sezonalnost. Na primer, možemo pretpostaviti da je potrošnja goriva mnogo veća u letnjim periodima, nego u zimskim. Ovo su određeni faktori koje klasičan ARIMA model ne može da precizno definiše, te su istraživači došli do zaključka da je potrebno modelovati proširenje standardnog modela i tako nastaje SARIMA modela (*eng. Seasonal ARIMA*). Ovaj model zapravo na arhitekturi standardnog ARIMA modela nadograđuje još dodatnu dimenziju sezonalnosti.

Pored predikcija cena goriva, ovaj model je jako koristan i kada je potrebno prediktovati određene cene, recimo kada su sniženja u nekim prodavnicama koja se dešavaju u određenim intervalima, kao što su letnje, zimsko, prolećno ili jesenje sniženje (Hyndman & Athanasopoulos, 2018).

Ovaj model se matematički može formalizovati na sledeći način:

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t + \Phi_1 Y_{t-s} + \Phi_2 Y_{t-2s} + \dots + \Phi_P Y_{t-Ps} + \Theta_1 \varepsilon_{t-s} + \Theta_2 \varepsilon_{t-2s} + \dots + \Theta_Q \varepsilon_{t-Qs}$$

Jednačina 5 SARIMA model

Gde su:

- Y_t – trenutna vrednost vremenske serije u trenutku t
- c – konstanta
- $\phi_1, \phi_2, \dots, \phi_p$ – koeficijenti autoregresivne komponente
- p – red AR komponente, broj prethodnih vremenskih tačaka koje utiču na Y_t
- ε_t – greška, odnosno neočekivani šok u vremenskoj seriji
- $\theta_1, \theta_2, \dots, \theta_q$ - koeficijenti pokretnog proseka
- q – red MA komponente, broj prethodnih grešaka koje utiču na Y_t
- s - sezonalni period
- $\Phi_1, \Phi_2, \dots, \Phi_P$ - koeficijenti AR sezonalne komponente koji određuju uticaj prošlih sezonalnih vrednosti na trenutnu vrednost
- P - red AR sezonalne komponente
- $\Theta_1, \Theta_2, \dots, \Theta_Q$ - koeficijenti MA sezonalne komponente koji određuju uticaj prethodnih sezonalnih grešaka na trenutnu vrednost

- **Q** - red MA sezonalne komponente

Generalno govoreći, možemo zaključiti da su se ovi prvobitni modeli u većini slučajeva pokazali prilično korisnim kada su korišćeni nad vremenskim serijama koje su bile stacionarne, odnosno serije koje u okviru sebe nisu uključivale dodatne dimenzije sezonskih obrazaca ili novih trendova. Međutim, sa bržim razvojem globalnog tržišta kao i brojnim turbulentim periodima koji su izazivali neočekivane promene, bilo je potrebno proširiti klasične modele kako bi se u obzir uzeli i ovi faktori, jer je postalo očigledno da u uslovima povećane nestabilnosti tržišta, osnovni linearni modeli nisu pokazali dovoljno dobre rezultate i mogućnosti adekvatnog modelovanja volatilnosti (Bollerslev, 1986). Ovi modeli su postavili osnovu za dalje usavršavanje i za definisanje novih, preciznijih načina predikcija vremenskih serija, koje se u današnjem modernom dobu baziraju na principima Veštačke inteligencije i mašinskog učenja.

Tabela ispod pokazuje generalni prikaz prethodno navedenih modela zajedno sa uopštenom analizom njihovih prednosti i nedostataka.

Tabela 3 Prikaz prednosti i nedostaka klasičnih modela

Model	Primena	Prednosti	Nedostaci
AR	Kratkoročna predikcija koja se bazira na stacionarnim vremenskim serijama	Jednostavna i brza implementacija	Loše performanse kod nestacionarnih vremenskih serija
MA	Korekcija šokova i grešaka u predikciji	Takođe jednostavan i brz za modelovanje	Teže hvata dugoročne zavisnosti
ARMA	Predikcija stacionarnih serija uz uključivanje šokova	Kombinacija prednosti AR i MA modela uz dodatak dimenzije šoka	Iskazuje relativno loše performanse kod nelinearnih i nestacionarnih serija
ARIMA	Dugoročna predikcija koja uključuje i promenljive trendove	Modelovanje kombinacije trendova i stacionarnosti kroz diferenciranje	Ne uključuje sezonske efekte i trendove

SARIMA	Dugoročna predikcija koja uključuje i sezonske trendove	Nadomešćuje prednosti klasičnog ARIMA modela za modelovanje sezonskih obrazaca	Nešto kompleksniji za implementaciju i podešavanje parametara
--------	---	--	---

2.3.3.1.5 Analiza klasičnih modela i modela Veštačke inteligencije

Tokom 2000tih godina razvojem neuronskih mreža, ovaj proces postaje značajno lakši i precizniji, a određeni nedostaci koje su klasične arhitekture ovih mreža pokazale u okviru dobijenih rezultata se nadomešćuju razvojem modernijih i specifičnijih arhitektura, kao što su Informer, Autoformer i FEDformer. Takođe, popularizacijom multimodalnog učenja, vremenske serije se uspešno mogu kombinovati i sa ostalim tipovima ulaznih podataka, kao što su slike ili tekstovi.

Tabela 4 Analiza stohastičkih i modernih modela za obradu vremenskih serija

Kriterijum	Stohastički modeli (ARIMA, SARIMA, ARMA)	Savremeni AI modeli (LSTM, Transformer, Informer)
Obrada nelinearnosti	Dosta ograničeni, zahtevaju predefinisane struktura podataka	Modeluju složene nelinearne odnose
Dugoročno pamćenje	Nemaju ovu mogućnost	LSTM modeli mogu jako dobro pamtit dugoročne zavisnosti
Multimodalni ulazi	Teška integracija	Moguće lako integrisati različite tipove podataka
Robusnost na šokove	Prilično osetljivi na nepredvidive promene	Imaju veću prilagodljivost
Potreba za stacionarnošću	Neophodno (pogotovo kod ARIMA modela)	Može postojati potreba, ali nije obavezno implementirati

Objašnjivost modela	Veća (parametri koji se koriste imaju ekonomsko značenje)	Manja te je potrebno proširiti metodama za objašnjavanje
Skalabilnost	Ograničena na male i srednje serije	Odlična skalabilnost na velikoj količini vremenskih serija

3 Analiza naučnih radova

Kao što je u odeljku „Struktura rada“ opisano, u ovom poglavlju biće dat sažet pregled relevantnih naučnih radova, između ostalog i onih koji su korišćeni kao glavna referenca za pisanje ovog rada. Analiza sveobuhvatne, dostupne literature je odrađena sistematičnim pristupom i kao takva biće tabelarno predstavljena u nastavku.

Pregledom dostupne literature se generalno može zaključiti da se u većini istraživačkih radova predominantno koriste hibridni modeli, gde se klasični modeli poput ARIMA ili CEEDAN modela kombinuju sa neuronskim mrežama kako bi se dobile što preciznije predikcije. Takođe, radi bolje optimizacije iz predloženog se može videti da su metaheuristički algoritmi ključni kada govorimo o optimizaciji modela i kroz dostupnu literaturu se mogu jasno uvideti i primeri određenih algoritama koji su korišćeni, poput PSO, SSA, RSA i mnogih drugih. Međutim, ovde ponovo dolazimo do teoreme „nema besplatnog ručka“, odnosno činjenice da čak i u ovako velikom broju istraživačkih radova ne postoji jedna univerzalno najbolja metoda, odnosno model. Ono što je generalno glavni izazov kada govorimo o problemu predviđanja vremenskih serija jeste zapravo poteškoća da se napravi adekvatan balans između tačnosti i efikasnosti samih modela koji se koriste u rešavanju ovog problema.

Kao glavni nedostatak većine istraživačkih radova jeste fragmentisani skup podataka nad kojima se rade evaluacije modela, s obzirom na to da je jako teško naići na javno dostupne podatke koji su dovoljno dobro standardizovani za ove potrebe. Samim time se veći akcenat mora staviti na pripremu samih podataka. Isto tako postoji dosta empirijskih studija koje nisu rađene nad realnim, industrijskim podacima. Takođe može se primetiti da postoji dosta rupa u

opisu toga kako objašnjiva inteligencija može dodatno pomoći u donošenju pravovremenih
biznis odluka.

Cilj ove disertacije između ostalog, kao što je već napomenuto, jeste i da popuni određene praznine koje postoje u dostupnim istraživačkim radovima, kao i da razvojem optimizovanijeg modela pokuša da poboljša preciznost predikcija. Takođe, koristi tehnike objašnjive inteligencije kako bi se što bolje interpretirali dobijeni rezultati, i kako bi se na osnovu toga mogli doneti precizniji zaključci i poslovne odluke.

Tabela 5 Tabelarni skraćeni pregled literature i naučnih radova iz ove oblasti

#		Opis istraživanja	Rezultati istraživanja
1	Dağdeviren, M. et al. (2007)	Uveden je Analytic Network Process (ANP) za selekciju osoblja, naglašavajući međuzavisnosti faktora i konflikte među kriterijumima.	Omogućava precizniji izbor kadrova, smanjenje grešaka i bolju podršku u kompleksnim organizacijama. Fuzzy proširenje poboljšava kvantifikaciju teško mernih kriterijuma.
2	Lin, H. et al. (2020)	Hibrid CEEMDAN + ML-GRU za predviđanje cena sirove nafte. CEEMDAN razbija signal na komponente, GRU uči nelinearne obrasce.	Poboljšana tačnost u odnosu na ARIMA i LSTM. Slabost: visoki računarski troškovi i složenost implementacije.
3	Li, J. et al. (2021)	Enhanced Sliding Recurrent Wave Neural Network (E-SRWNN) za predikciju cena energije; koristi klizne prozore i talasne funkcije.	Visoka prediktivna moć, nadmašuje standardne RNN i LSTM. Ograničenje: niska objašnjivost i testiranje na jednom datasetu.
4	Hualing et al. (2021)	Hibrid CNN-RNN model sa skip konekcijama za dugoročne zavisnosti u cenama nafte.	Poboljšana tačnost predikcije kod naglih promena, ali ograničena generalizacija na druge serije.
5	L. Jovanovic et al. (2020)	SSA (Salp Swarm Algorithm) optimizuje LSTM hiperparametre za predikciju sirove nafte.	Nadmašuje konkurenciju u tačnosti; slabost: visoka računarska zahtevnost i osetljivost na kvalitet ulaznih podataka.
6	Jovanović, A. et al. (2024)	Multi-Headed LSTM optimizovan modifikovanim PSO za predikciju cena goriva.	Postignut MSE=0.044 i R ² =0.91; visok potencijal za industrijsku primenu; zahtevni računarski resursi.
7	Baćanin, N. et al. (2020)	Unapređene varijante PSO za optimizaciju težina i hiperparametara neuronskih mreža.	Brža konvergencija i stabilniji ANN modeli; slabija skalabilnost na velike dataset-e.

8	Pamucar, D. et al. (2019)	Višekriterijumska EDAS metoda u transportu i energetici.	Efikasnost u rangiranju i odabiru optimalnih rešenja; ne primenjena na vremenske serije.
9	RSA, WOA, ARO, Chimp optimization (2019–2023)	Metaheurističke metode za optimizaciju LSTM i dubokih mreža za energiju, finansije, zlato i EEG signale.	Poboljšanje tačnosti predikcije; slabost: visoka specifičnost po domenu.
10	Tradicionalni optimizatori (ABC, FA, BA, SCA, GA)	Klasični metaheuristički algoritmi za podešavanje hiperparametara neuronskih mreža.	Nijedan algoritam univerzalno superioran, potvrđuje No Free Lunch teorem.
11	Zhang, X. et al. (2017)	Hibrid ARIMA-LSTM kombinuje linearnu sezonalnu komponentu (ARIMA) sa nelinearnim kapacitetom LSTM mreže.	Bolja tačnost od pojedinačnih ARIMA ili LSTM; potreban ručni tuning ARIMA komponente.
12	Feng, J. et al. (2019)	WOA optimizuje LSTM hiperparametre u predikciji cena akcija.	Poboljšana tačnost; slabost: osetljivost na inicijalne parametre.
13	Heidari, A. et al. (2019)	HHO optimizuje težine neuronskih mreža u vremenskim serijama.	Brza konvergencija i bolje rešenje od PSO i GA; rizik od lokalnih optimuma.
14	Mei, Y. et al. (2020)	ARO optimizuje predikciju obnovljivih izvora energije.	Smanjena greška u odnosu na PSO i GA; ograničena validacija bez industrijske primene.
15	Wang, H. et al. (2021)	Chimp Optimization (ChOA) optimizuje LSTM modele u finansijskim podacima.	Nadmašuje tradicionalne optimizatore u tačnosti; heuristička priroda.
16	Zhao, Y. et al. (2022)	RSA optimizuje duboke mreže za predikciju cena zlata; balansira eksploataciju i istraživanje.	Poboljšana tačnost i brža konvergencija; nova metoda sa ograničenim testiranjem.
17	Mirjalili, S. et al. (2014, 2016)	GWO i SSA za optimizaciju ANN i LSTM modela,	Dobra ravnoteža eksploatacije i istraživanja; moguća stagnacija u kasnim iteracijama.

		inspirisani prirodnim ponašanjima.	
18	Sun, Y. et al. (2023)	Hibridni CNN + BiLSTM za predikciju potrošnje goriva i derivata.	Povećana tačnost u odnosu na RNN ili ARIMA; problem interpretabilnosti.
19	Alqahtani, A. et al. (2021)	XAI sa SHAP metodom za LSTM modele u energetske podacima.	Veća transparentnost i bolja interpretacija; ponekad neusklađeno sa domenskim znanjem.
20	Baćanin, N. et al. (2023)	Self-adaptive hybrid PSO za treniranje laganih LSTM sa adaptivnim parametrima.	Značajno smanjenje vremena treniranja bez gubitka tačnosti; potreban dodatni test na realnim dataset-ima.
21	Hinton, G. et al. (2012)	Duboke neuronske mreže za akustičko modelovanje u prepoznavanju govora.	Poboljšana preciznost prepoznavanja; zahteva veliki dataset i resurse.
22	Kingma, D. & Ba, J. (2015)	Adam optimizer za stohastičku optimizaciju.	Brža i stabilnija konvergencija u treniranju dubokih mreža.
23	Graves, A. (2012)	RNN za sekvencijalno učenje u nadgledanom režimu.	Visoka tačnost u predikciji sekvenci; zahteva optimizovane hiperparametre.
24	Devlin, J. et al. (2019)	BERT: Pre-treniranje bidirekcionalnih transformera za NLP.	Bolja performansa u NLP zadacima; visok zahtev za memoriju i resurse.
25	Mirjalili, S. (2015)	Ant Lion Optimizer za optimizaciju funkcija.	Dobra ravnoteža eksploracije i eksploatacije; može stagnirati kod velikih problema.
26	Mirjalili, S. (2016)	Dragonfly Algorithm za rešavanje single- i multi-objective problema.	Efikasno pronalaženje rešenja; slabije performanse kod visoko dimenzionalnih prostora.
27	Mirjalili, S. (2014)	Grey Wolf Optimizer za ANN i LSTM optimizaciju.	Brza konvergencija i dobra tačnost; rizik od lokalnih maksimuma.
28	Baćanin, N. et al. (2022)	Metaheuristike za optimizaciju hiperparametara LSTM u predikciji cena nafte.	Poboljšana tačnost i stabilnost; ograničena generalizacija na druge domene.

29	Pedregosa, F. et al. (2011)	Scikit-learn framework za ML u Pythonu.	Pojednostavljuje implementaciju i evaluaciju ML modela; ne optimizuje hiperparametre automatski.
30	McKinney, W. (2017)	Python za analizu podataka, biblioteka Pandas.	Brza manipulacija podacima i vizualizacija; nije specijalizovano za duboko učenje.
31	Chen, T. & Guestrin, C. (2016)	XGBoost za skalabilno boosting drveće.	Visoka preciznost klasifikacije/regresije; potreban fin tuning hiperparametara.
32	Breiman, L. (2001)	Random Forests algoritam za klasifikaciju i regresiju.	Robustan, smanjuje overfitting; slab u interpretabilnosti.
33	Freund, Y. & Schapire, R. (1997)	AdaBoost algoritam za poboljšanje klasifikatora.	Poboljšana preciznost; osetljiv na šum u podacima.
34	Cortes, C. & Vapnik, V. (1995)	SVM za klasifikaciju i regresiju.	Efikasan u visoko dimenzionalnim prostorima; izbor kernela je kritičan.
35	Quinlan, J. (1993)	C4.5 algoritam za učenje stabala odlučivanja.	Dobri rezultati na manjim datasetima; sklono overfittingu na velikim setovima.
36	Kohavi, R. (1995)	Analiza cross-validation i bootstrap za procenu modela.	Preporučuje metode za stabilnu procenu tačnosti; može biti računski zahtevno.
37	Schwarz, G. (1978)	BIC kriterijum za procenu dimenzionalnosti modela.	Balansira fit i kompleksnost modela; ograničen kod malih uzoraka.
38	Akaike, H. (1974)	AIC kriterijum za statističku selekciju modela.	Efikasan u balansiranju fit i broj parametara; ne daje apsolutnu tačnost modela.
39	Lichman, M. (2013)	UCI ML repository za dataset-e.	Omogućava lako testiranje i poređenje ML algoritama; ograničen u domenskim specifičnostima.
40	Gal, Y. & Ghahramani, Z. (2016)	Dropout kao Bayesian aproksimacija za model uncertainty.	Daje merenje nesigurnosti modela; zahteva višestruke forward pass-ove.

41	He, K. et al. (2016)	ResNet arhitektura za prepoznavanje slika.	Omogućava treniranje dubokih mreža bez problema sa vanishing gradient; visoki resursi.
42	Devlin, J. et al. (2019)	BERT za NLP (ponovljeno za referencu).	Poboljšava razumevanje konteksta; memorijski zahtevno.
43	Tibshirani, R. (1996)	Lasso regresija za selekciju i shrinkage.	Efektivno smanjuje dimenzionalnost; može eliminisati važne varijable pri visokoj korelaciji.
44	Friedman, J. et al. (2001)	The Elements of Statistical Learning za ML teoriju.	Obuhvata širok spektar algoritama; ne sadrži praktične primere kodiranja.
45	Rousseeuw, P. & Leroy, A. (1987)	Robust Regression i detekcija outliera.	Bolje upravljanje outlier-ima; može biti računarski zahtevno za velike setove.
46	Cohn, D. et al. (1994)	Active learning za poboljšanje generalizacije ML modela.	Smanjuje broj potrebnih labela; zahteva iterativni proces.
47	Russell, S. & Norvig, P. (2010)	Pregled AI: teorija i primena.	Sveobuhvatan vodič; ne fokusira se na konkretne optimizacione metode.
48	Nilsson, N. (1998)	Synthesized AI pristupi.	Kombinuje paradigme AI; zastarelo za moderni deep learning.
49	Baćanin, N. et al. (2023)	Cloud computing load prediction pomoću LSTM + PSO optimizacije.	Precizna predikcija opterećenja; potrebna dalja industrijska validacija.
50	Predić, B. et al. (2024)	Cloud-load forecasting uz decomposition-aided attention RNN + PSO.	Poboljšana tačnost predikcije; visok potencijal za edge computing aplikacije.

4 Duboko učenje

U ovom poglavlju biće dat osnovni prikaz taksonomije dubokog učenja, njegov kratak istorijat kao i značaj koji ostvaruje na naučnom polju, pogotovu kada govorimo o predikciji vremenskih serija. Takođe, biće prikazani i modeli koji su korišćeni u samom istraživačkom procesu, kao i značaj njihove optimizacije radi dobijanja što preciznijih rezultata.

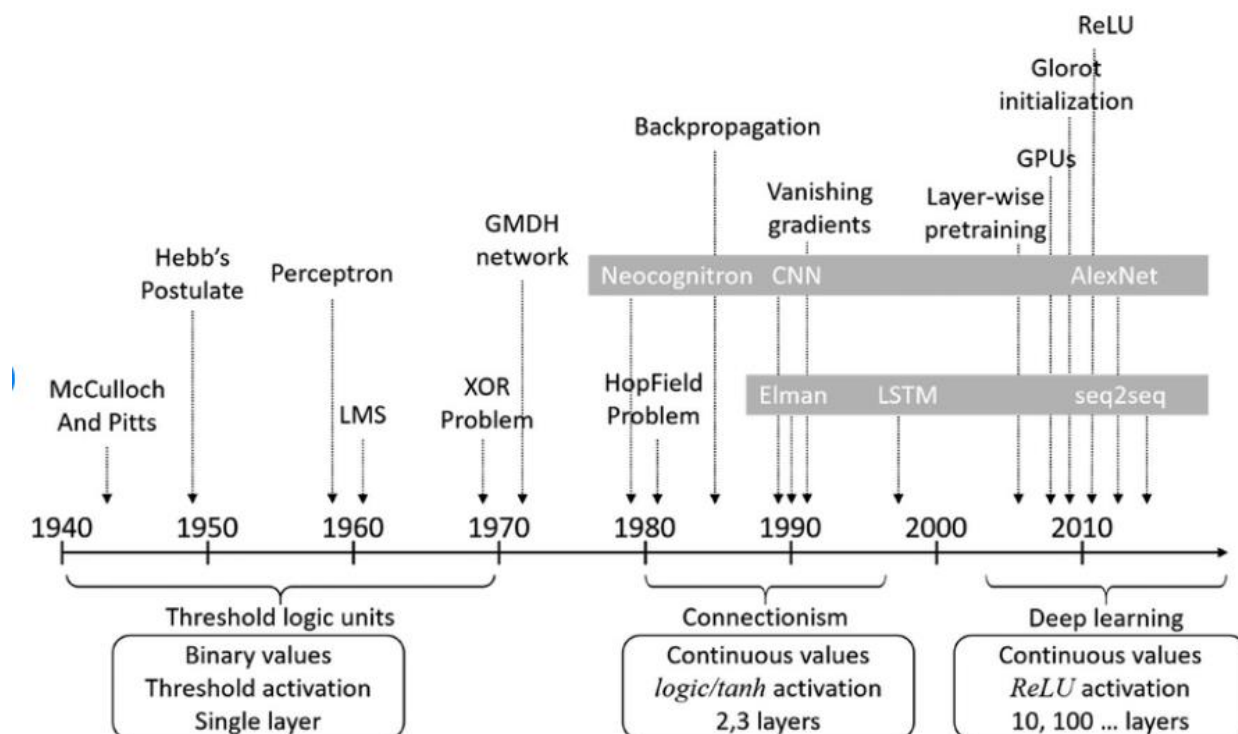
4.1 Uvod i kratak istorijat

Do sada su u više navrata bili korišćeni termini poput Mašinskog učenja, modela, Veštačke inteligencije. Neki osnovni principi i osnovni značaj Veštačke inteligencije je objašnjen u prethodnim poglavljima, međutim nekim logičnim sledom ovih termina, otvara se novo pitanje, a to je šta zapravo predstavlja mašinsko učenje i u kakvoj je ono korelaciji sa Veštačkom inteligencijom.

Mašinsko učenje možemo definisati kao podoblast Veštačke inteligencije, koja koristi algoritme trenirane na nekim ulaznim skupom podataka. Osnovni cilj ove oblasti je dobijanje modela koji mašinama omogućavaju da rešavaju neke logične probleme ili zadatke, za koje bi inače bio potreban ljudski resurs. Ilustracija ovog pristupa data je na slici 3. Generalno govoreći, mašinsko učenje možemo podeliti na tri osnovne podgrupe, na osnovu načina kako se algoritmi treniraju nad podacima:

- **Nadgledano učenje (eng. *Supervised Learning*)** – ovo je tip mašinskog učenja gde modeli uče iz nekih označenih, labeliranih podataka i vrlo često se koristi za potrebe klasifikacije ili regresije.
- **Nenadgledano učenje (eng. *Unsupervised Learning*)** – modeli uče i otkrivaju strukturu u podacima koji nisu označeni, za šta imamo različite metode kao što su klasterovanje (eng. *Clustering*), redukcija dimenzionalnosti (eng. *Dimensionality Reduction*) i mnogi drugi.
- **Učenje sa podsticajem (eng. *Reinforcement Learning*)** – modeli uče nove šablone kroz princip nagrade ili kazne.

Sa druge strane, pored ovih osnovnih podela, u sklopu mašinskog učenja postoji i jedna specifična podoblast koja se zove duboko učenje (*eng. Deep Learning*). Ovaj tip mašinskog učenja predstavlja kombinaciju nadgledanog i nenadgledanog pristupa, koji se u glavnom bazira na arhitekturama neuronskih mreža sa više slojeva. Ova podoblast se često smatra posebnom taksonomijom zbog svoje specifične primene, kapaciteta i sposobnosti za učenje iz nekih prethodno neobrađenih setova podataka (Goodfellow et al., 2016).



Ilustracija 4 Istorijat razvoja modela mašinskog učenja

Ako posmatramo istorijat ove oblasti koji je prikazan na Slici 4, možemo uočiti da iako sam koncept neuronskih mreža datira još iz četrdesetih godina prošlog veka (McCulloch & Pitts, 1943), neka prva zvanična primena ovih arhitektura se javlja tokom 1980tih godina. Paralelno sa razvojem i unapređenjem arhitekture neuronskih mreža, javlja se potreba i razvojem dodatnog algoritma, takozvanog unazadnog širenja greške (*eng. Backpropagation*) (Rumelhart et al., 1986). Kako i sam naziv nagoveštava, ovaj algoritam se koristi tokom procesa treniranja neuronskih mreža, kada se greška koja je izračunata na izlaznim slojevima mreže vraća nazad kroz ostale slojeve kako bi se izračunali težinski koeficijenti, i ovo je jedan od najčešćih načina kako se može poboljšati tačnost i performanse samih mreža. Nešto kasnije, tokom 2000tih

godina dolazi do velike ekspanzije u istraživanju i poboljšanju kvaliteta ovih arhitektura, na šta su uticali mnogi faktori a pre svega značajno bolji računarski resursi kao i mnogo veće količine podataka kojima je bilo znatno lakše pristupiti.

Tokom 2012. godine dolazi do prelomnog momenta za oblast dubokog učenja, kada je AlexNet, duboka konvolutivna neuronska mreža, drastično povećala svoje performanse i nadmašila konkurenciju na takmičenju ImageNet. Možemo reći da je ovaj momenat na neki način označio početak „zlatnog doba“ u daljem razvoju oblasti mašinskog učenja (Krizhevsky et al., 2012).

U tabeli ispod mogu se sagledati osnovne prednosti modela dubokog učenja u odnosu na klasične modele, što na neki način samo pokazuje da obično naučnici prirodno navigiraju ka izboru ovih modela kada je u pitanju rad sa velikim, dinamičnim i nelinearnim tipova podataka, kao što su recimo vremenske serije.

Tabela 6 Analiza modela dubokog učenja i klasičnih modela mašinskog učenja

Dimenzija	Klasično mašinsko učenje	Duboko učenje
Potreba za inženjeringom osobina (eng. Feature engineering)	Visoka	Niska
Performanske na velikim skupovima	Ograničene	Visoke – modeli imaju mogućnost skaliranja u skladu sa podacima
Nelinearnost	Ograničena	Visoka
Generalizacija	Zависи od specifičnog modela, kod nekih modela je visoka, ali kod nekih može biti izuzetno niska	Većina modela ima mogućnost dobre generalizacije
Interoperabilnost	Visoka	Niska

Kada govorimo o ulozi koji ovi algoritmi ostvaruju u predikciji vremenskih serija, možemo zaključiti da mogu ostvariti značajno bolje rezultate u odnosu na tradicionalne modele. S obzirom na to da se klasični modeli mašinskog učenja u glavnom baziraju na pretpostavkama o stacionarnosti podataka i linearnosti u njima, duboko učenje ove nedostatke nadomešćuje time što se može koristiti za modelovanje nelinearnih i višedimenzionalnih odnosa u podacima, koristeći sekvencijalne arhitekture, kao što su prethodno pomenuti LSTM ili GRU modeli (Hewamalage et al., 2021), o kojima će biti nešto više reči u narednim poglavljima. Pored ovoga, modeli dubokog učenja mogu takođe u okviru svojih predikcija uzeti u obzir i integrisati određene eksterne faktore, o kojima je bilo nešto više reči ranije, kao što su geopolitički događaji ili brojni ekonomski indikatori, bez potrebe za eksplicitnim programiranjem dodatnih pravila.

4.2 Rekurentne neuronske mreže

U jednom od prethodnih poglavlja je bio dat kraći pregled često korišćenih modela mašinskog učenja koji se svakodnevno koriste u načno-istraživačkim procesima ili za rešavanje nekih specifičnih problema. Kako se ovo istraživanje, pa i sama disertacija baziraju na korišćenju neuronskih mreža, odnosno njihovih specifičnih arhitektura kao što su rekurentne neuronske mreže sa više glava, u narednim sekcijama biće dat nešto detaljniji prikaz ovih modela, kao i potencijal koji ostvaruju u okviru istraživanja koji se baziraju na predikciji vremenskih serija, konkretno u ovom slučaju na predikciji cene goriva.

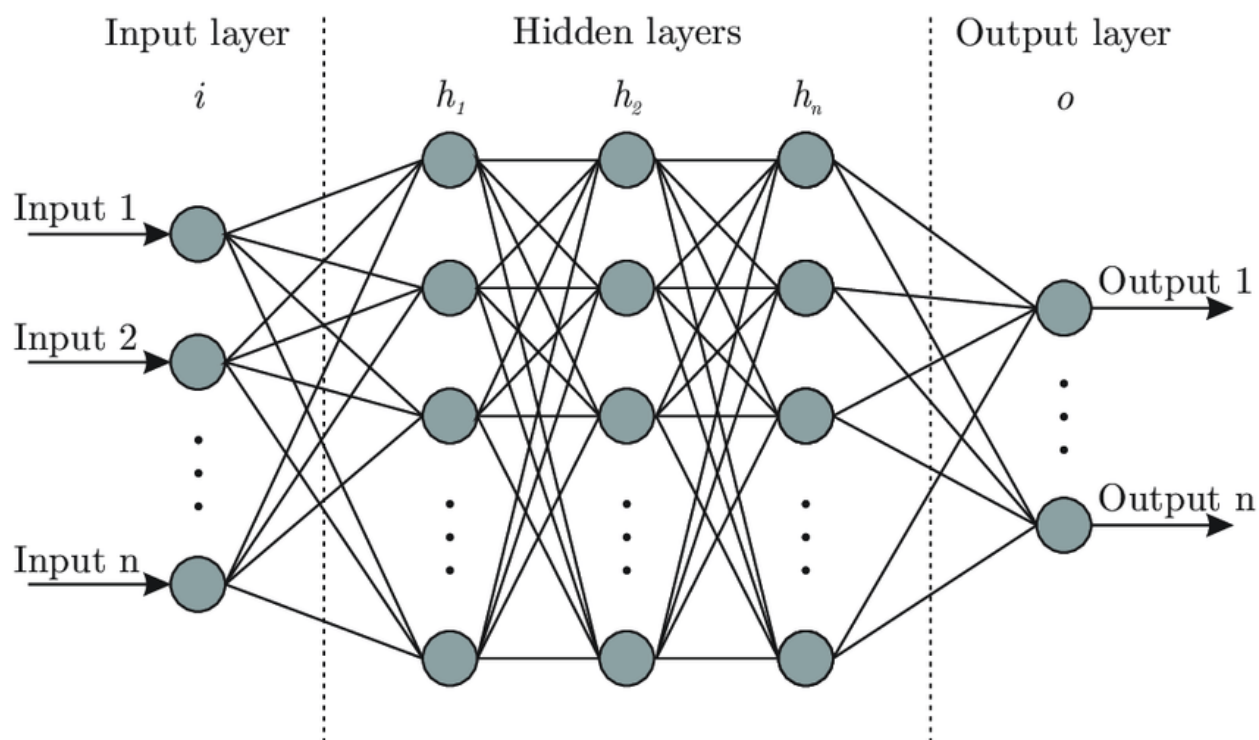
Dakle, kao što je ranije napomenuto, koncept neuronskih mreža se prvi put pominje četrdesetih godina prošlog veka, kada su MekKuloh (*eng. McCulloch*) i Pits (*eng. Pitts*), predstavili prvi formalni oblik ovog modela. Kako i sam naziv nagoveštava, veštačke neuronske mreže se u svojoj suštini baziraju na biološkim osnovama prirodnih neuronskih mreža, i samim time oponašaju njihovo funkcionisanje. Ideja do koje su MekKuloh i Pits došli kada su pokušavali da formalizuju ovaj model jeste zapravo matematička reprezentacija biološkog neurona, koji je sposoban da obavlja razne logičke zadatke na osnovu jednostavnog praga aktivacije (McCulloch & Pitts, 1943).

Dalji razvoj i istraživanje ovih modela se nastavlja tokom pedesetih godina prošlog veka, kada je Frenk Rozenblat (*eng. Frank Rosenblatt*) predstavio prvi perceptron, odnosno algoritam za učenje sa težinama, što je kasnije omogućilo osnovne operacije klasifikacije (Rosenblatt, 1958). Međutim, sam perceptron je imao nekih svojih ograničenja, posebno u situacijama kada je trebalo rešiti probleme koji nisu linearnog razdvajanja (Minsky & Papert, 1969).

Dalje, tokom sedamdesetih godina prošlog veka, interesovanje za istraživanje ovih modela, kao i cele oblasti Veštačke inteligencije je bilo u znatnom padu, da bi 1986. ponovo zaživelo. Tada su Rumelhart (*eng. Rumelhart*), Hinton (*eng. Hinton*) i Vilijsams (*eng. Williams*) predstavili mehanizam unazadnog širenja greške (*eng. backpropagation*), omogućivši time učenje višeslojnih perceptrona i rešavanje dosta kompleksnijih problema (Rumelhart et al., 1986).

Kada govorimo u arhitekturi klasičnih neuronskih mreža, možemo zaključiti da se one sastoje od nekoliko bitnih komponenti, kao što su neuroni i slojevi. Svaka tradicionalna neuronska mreža se u glavnom sastoji od tri glavne komponente:

1. Ulazni sloj (*eng. Input layer*) – Prima ulazne podatke
2. Skriveni slojevi (*eng. Hidden layers*) – Obrađuju podatke kroz nelinearnih transformacija
3. Izlazni sloj (*eng. Output layer*) – Generiše neki rezultat



Ilustracija 5 Arhitektura neuronske mreže

Svaki od slojeva se sastoji od više neurona, gde je svaki neuron jednog sloja povezan sa neuronima prethodnog sloja putem težina (*eng. weights*) i bijasa (*eng. bias*). Težine predstavljaju glavne faktore koji određuju koliko će signal iz jednog neurona prethodnog sloja uticati na aktivaciju neurona u narednom sloju. Sa druge strane, bijas predstavlja dodatnu konstantu koja omogućava modelu da pomeri aktivacionu funkciju neurona nezavisno od ulaznih vrenosti. Svaki neuron ima zadatak da na svom izlazu računa svoju aktivaciju primenom takozvane aktivacione funkcije na određeni ponderisani skup ulaznih podataka, odnosno signala koji su došli iz prethodnog sloja.

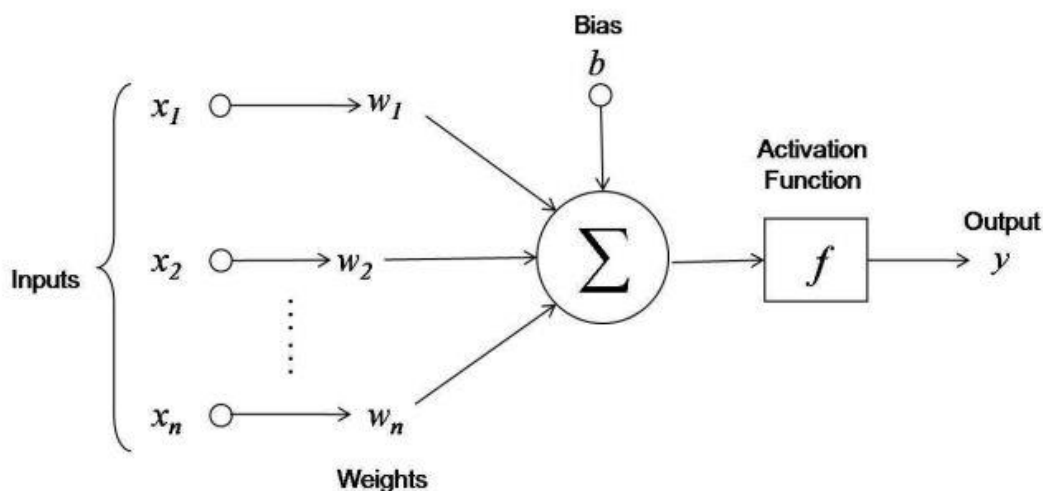
Rad jednog neurona možemo matematički formalizovati sledećom formulom:

$$y = f\left(\sum_{i=1}^n w_i x_i + bt\right)$$

Jednačina 6 Matematički prikaz neurona

Gde su:

- x_j – ulazne vrednosti za neuron
- w_j – težina koja odgovara ulazu x_i
- $\sum_{i=1}^n w_i x_i$ – zbir ponderisanih ulaza
- b – bijas
- $f(\cdot)$ – aktivaciona funkcija
- y - izlaz neurona nakon primene funkcije aktivacije



Ilustracija 6 Arhitektura neurona

Kada govorimo o aktivacionoj funkciji, možemo reći da ona predstavlja takođe jednu od ključnih komponenti u arhitekturama neuronskih mreža, jer omogućava da se uvede izvesna nelinearnost u radu samog modela i time pospeši preciznost generalizacije. Ovime se neuronskim mrežama omogućava da modeluju određene složene obrasce i zavisnosti među podacima (Goodfellow et al., 2016). Glavni zadatak aktivacione funkcije je da dohvati ulaznu sumu, odnosno ponderisani zbir ulaznih podataka – signala, najčešće iz prethodnog sloja i transformiše je u adekvatni izlaz (*eng. output*) koji se šalje dalje kroz mrežu, odnosno na naredni sloj.

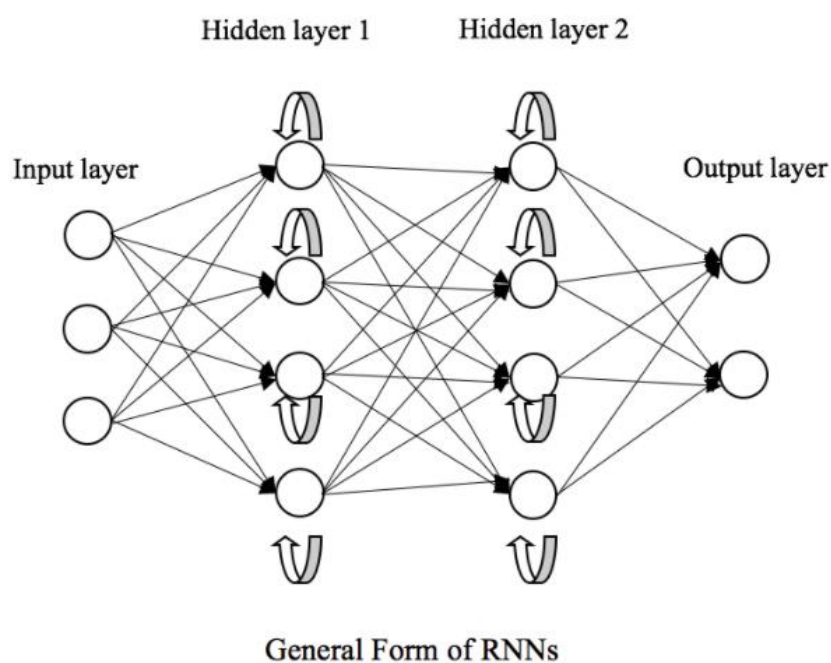
Postoji nekoliko različitih tipova aktivacionih funkcija, a neke od njih su predstavljene u nastavku:

Tabela 7 Prikaz često korišćenih aktivacionih funkcija

Naziv	Formula	Opseg vrednosti	Prednosti	Nedostaci
Sigmoid (Logistička funkcija)		0,1	Korisna kod binarnih klasifikacija	Javlja se problem zanemarljivih gradijenata kod veoma velikih ili malih ulaznih vrednosti
Tanh (Hiperbolički tangens)		-1,1	Centrirana oko 0, što ubrazava konvergenciju	Takođe se javlja problem zanemarljivih gradijenata (eng. Vanishing gradient problem)
ReLU (Rectified Linear Unit)		$[0, \infty)$	Jednostavna za implementaciju	Javlja se tzv. Problem mrtvih neurona
Softmax			Niz izlaznih vrednosti se pretvaraju u distribuciju verovatnoća koje zbirno uvek daju 1	Eksponecijalno „zasićenje“

Tokom implementacije neuronskih mreža, treba imati uvek u vidu da je izbor aktivacione funkcije od izuzetnog značaja za sam model. Ona pre svega može uticati na određene parametre kao što je brzina konvergencije, sposobnost modeliranja određenih nelinearnih obrazaca kao i na celokupnu stabilnost i preciznost samog procesa učenja i kasnije generalizacije. Recimo, ReLU aktivaciona funkcija, koja se ujedno i najčešće upotrebljavana je omogućila obuku dubokih mreža koje ranije nije bilo moguće u potpunosti trenirati zbog problema sa propagacijom gradijenata kroz mnoge skrivene slojeve (Krizhevsky et al., 2012).

Nakon što su aktivacione funkcije omogućile neuronskim mrežama da modeluju složene nelinearne obrasce, sledeći glavni izazov koji se pojavio bio je obrada sekvencijalnih podataka, odnosno podataka gde redosled i njihova vremenska zavisnost igraju ključnu ulogu. Ovakvi tipovi podataka podrazumevaju recimo tekst, vremenske serije ili neke govorne signale. Kako bi se nadomestili nedostaci koji su postojali u klasičnim modelima neuronskih mreža, došlo je do pojave novog podtipa ili podarhitekture ovih modela pod nazivom rekurentne neuronske mreže (*eng. Recurrent neural networks, RNN*). Ove mreže praktično imaju sposobnost da „pamte“ informacije o prethodnim ulazima putem takozvanih skrivenih stanja (*eng. hidden states*) (Elman, 1990).



Ilustracija 7 Arhitektura rekurentnih neuronskih mreža

U poređenju sa arhitekturom klasičnih neuronskih mreža, modeli rekurentnih neuronskih mreža imaju takozvane povratne veze koje omogućavaju prenošenje informacija kroz vremenske korake.

Ovaj tip modela možemo matematički formalizovati kroz narednu formulu:

$$ht = \sigma(Wxhxt + Whhht - 1 + bh)$$

Jednačina 7 Model RNN mreže

Gde su:

- xt – ulaz u mrežu u trenutnom vremenskom koraku t
- ht – skriveno stanje neurona u vremenskom koraku t
- $ht - 1$ – skriveno stanje iz prethodnog vremenskog koraka
- Wxh – matrica težina koja povezuje trenutni ulaz xt sa skrivenim stanjem ht
- Whh – matrica težina koja povezuje prethodno stanje $ht - 1$ sa trenutnim stanjem ht
- bh – bijas skrivenog sloja
- $\sigma(\cdot)$ – aktivaciona funkcija

Na osnovu ovoga možemo zaključiti da informacija iz prošlih slojeva može da utiče na trenutni izlaz, što omogućava ovakvom tipu modela da jako lako i efikasno modeluje određene vremenske zavisnosti. Ovo je posebno korisno u situacijama kada je potrebno predvideti određene buduće vrednosti na osnovu dostupnih istorijskih obrazaca uz pratnju sekvencijalnosti samih podataka, kao što je slučaj kod predviđanja vremenskih serija, poput predikcija cena goriva, nafte, cena na berzi i mnogih drugih.

Međutim, iako su ovi tipovi neuronskih mreža jednostavni za implementaciju i trening, kao što obično biva i oni se suočavaju sa određenim izazovima i nedostacima, kao što su:

- Problem nestajućih gradijenata (*eng. vanishing gradient problem*) – gradijenti postaju izuzetno mali kroz duže sekvence podataka i time se dodatno otežava učenje dugoročnih zavisnosti (Bengio et al., 1994)

- Problem eksplodirajućih gradijenata (*eng. exploding gradients problem*) – suprotno od problema nestajućih gradijenata, u ovom slučaju oni mogu postati izuzetno veliki i na taj način otežavaju trening ovih modela

Kako bi se ovi nedostaci na neki način umanjili, i kako bi se ovi modeli napravili takvim da budu više prilagodljivi i primenljivi u rešavanju nekih zadataka i problema koji zahtevaju dugoročnu memoriju, razvijeni su specijalizovane varijante ovih modela, među kojima su najpoznatiji i najčešće korišćeni:

1. Dugoročno – kratkoročne memorije (*eng. Long Short-Term Memory – LSTM*) koje uvode zakozvane „memorijske ćelije“ koje daju veću kontrolu nad time šta mreža treba da „upamti“, a šta da zaboravi „odbaci“ (Hochreiter & Schmidhuber, 1997)
2. Jedinice sa zatvorenim povratnim vezama (*eng. Gated Recurrent Unit – GRU*) koje predstavljaju dosta jednostavniju varijantu LSTM modela, koji koriste manje parametara i pogodniji su za neke jednostavnije i brže obuke (Cho et al., 2014)
3. Rekurentne mreže sa više glava (*eng. Multi-Head RNN*)

Ono što je zajedničko za obe arhitekture, to je da koriste specijalizovane aktivacione funkcije u okviru svojih „kapija“ (*eng. gates*), čime se značajno poboljšava preciznost generalizacije i bolje treniranje.

4.3 LSTM i GRU modeli

U ovoj sekciji biće dat biti detaljniji pregled ovih modela, sa posebnim akcentom na model dugoročno-kratkoročne memorije (*eng. LSTM model*), koji je ujedno i predmet istraživačkog dela ove disertacije.

4.3.1 LSTM model

Kao što je u prethodnoj sekciji rada nagovešteno, glavni nedostaci neuronskih mreža poput nestajućih ili eksplodirajućih gradijenata u glavnom nastaju tokom procesa obučavanja modela na dužim sekvencijalnim podacima. Ovo može značajno uticati na preciznost konačnih predikcija i sposobnosti modela da generalizuje i ustanovi naučene obrasce (Bengio, Simard, & Frasconi, 1994). Kako bi se ovi nedostaci na neki način prevazišli, tokom devedesetih godina

prošlog veka došlo je do razvoja modela dugoročno-kratkoročnih memorija (*eng. Long Short-Term Memory – LSTM*) (Hochreiter & Schmidhuber, 1997). Učestala primena ovih modela u brojnim istraživanjima dovela je do toga da se oni smatraju za jednim od značajnijih naučnih pomaka, pogotovu kada govorimo o treniranju modela nad sekvencijalnim podacima. Ove arhitekture omogućavaju modelovanje dugoročnih zavisnosti među podacima, pa su tako pronašle široku primenu u obradi mnogih sekvencijalnih podataka kao što su tekst (Sundermeyer, Schlüter, & Ney, 2012), govorni signali (Graves, Mohamed, & Hinton, 2013) ili vremenske serije, poput cena goriva, nafte i mnogih drugih.

Kada govorimo o samoj arhitekturi ovih modela, možemo videti da ona predstavlja proširenje arhitekture klasičnih rekurentnih neuronskih mreža uvođenjem takozvanih memorijskih ćelija (*eng. memory cell*). Ove ćelije na svojim ulazima imaju specifičnu strukturu koju nazivamo “kapijama” (*eng. gates*). Kapije su zapravo posebni mehanizmi koji zapravo služe za filtriranje informacija, odnosno pomažu samoj mreži da odredi koje informacije treba zadržati, zaboraviti ili dodati (Gers, Schmidhuber, & Cummins, 2000). Na osnovu navedenog možemo zaključiti da osnovna ideja funkcionisanja memorijskih ćelija i ovih modela zapravo potiče ponovo iz biologije. Tačnije, ovi modeli praktično simuliraju biološko pamćenje, odnosno biološki mehanizam obrade i čuvanja određenih informacija u okviru nekog konteksta. Na osnovu toga šta određena ćelija treba da uradi sa dobijenom informacijom, možemo razlikovati i četiri osnovna tipa kapija, kao što su:

Tabela 8 Prikaz tipova kapija i njihove primene

Tip kapije	Simbol	Funkcija	Matematička formula
Ulazna kapija	i_t	Kontroliše koliko novih informacija iz trenutnog ulaza x_t treba dodati u stanje ćelije	$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$
Kapija zaboravljanja	f_t	Odlučuje koje informacije iz prethodnog stanja ćelije C_{t-1} treba zaboraviti	$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$
Izlazna kapija	o_t	Određuje koji deo stanja ćelije C_t ide na izlaz h_t	$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$
Ćelija kandidat	$\tilde{C}_t$	Kreira potencijalni novi sadržaj stanja ćelije koji može biti dodan preko input gate	$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C)$

Gde su

- h_{t-1} - prethodni izlaz
- x_t - trenutni ulaz
- C_{t-1} - prethodno stanje ćelije
- σ - sigmoid funkcija aktivacije
- ***tanh*** - tangens hiperbolična funkcija

Prethodno stanje ovih ćelija C_{t-1} se može delimično „resetovati“, kako bi se napravio dovoljan prostor za ažurno stanje same ćelije C_t , koje se zatim popunjava nekim novim informacijama. Matematički gledano, ovo možemo formalizovati korišćenjem takozvanog elementnog proizvoda (*eng. Element-wise product*), označen simbolom $\odot$, koji se koristi zajedno sa kapijom zaboravljanja f_t . Prikaz ovog proizvoda dat je sledećom formulom:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

Jednačina 8 Prethodno stanje ćelije

Inicijalni izlaz sigmoidalne fukcije o_t se dalje transformiše u zavisnosti od trenutnog stanja ćelije, i ovo je prikazano narednom formulom:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} - 1 + b_o)$$

Jednačina 9 Izlaz sigmoidalne funkcije

Konačni izlaz se zatim prosleđuje kroz *tanh* sloj, čime se formira ažurirano skriveno stanje modela, a što je predstavljeno formulom:

$$h_t = o_t \odot \tanh(C_t)$$

Jednačina 10 Konačni izlaz prosleđen tanh funkciji

Gde su:

- W_o, U_o – težinske matrice
- b_o - pristrasnost

Glavne prednosti ovakvog tipa modela se ogledaju pre sve u modelovanju dugoročnih zavisnosti među podacima, zahvaljujući mogućnosti da zadrže određene informacije kroz duže vremenske periode (Hochreiter & Schmidhuber, 1997). Ovo omogućava dosta stabilnije testiranje jer smanjuje verovatnoću za pojavom nestajućih ili eksplodirajućih gradijenata, koji su kako je ranije napomenuto, česti problem kod klasičnih rekurentnih neuronskih mreža (Pascanu, Mikolov, & Bengio, 2013). Pored ovoga, značajno se povećava fleksibilnost i prilagodljivost algoritma različitim sekvencama, čime se mogu lakše predvideti određene promene u obrascima samih podataka, šokovi ili neke druge potencijalno neočekivane promene, pogotovu u vremenskim serijama (Greff, Srivastava, Koutník, Steunebrink, & Schmidhuber, 2017).

Takođe, treba napomenuti da su arhitekture ovih mreža vrlo često bile predmet optimizacije pomoću brojnih metaheurističkih algoritama, o kojima će biti nešto više reči u narednom poglavlju. Među primerima ovakvih optimizacija, izdvaja se recimo korišćenje Reptilskog algoritma pretrage (*eng. Reptile Search Algorithm, RSA*) za podešavanje LSTM mreže u svrhu predviđanja količine energije proizvedene vetrenjačama (M. Pavlov-Kagadejev et al., 2024). Još jedan zanimljiv pristup vredan pomena obuhvata kombinaciju LSTM mreže optimizovane korišćenjem algoritma optimizacije kita (*eng. Whale Optimization Algorithm – WOA*) i metode varijacione dekompozicije modova (*eng. Variational Mode Decomposition – VMD*) za predviđanje cena zlata, što takođe predstavlja primer predikcije vremenskih serija (S. Golubovic et al., 2023).

4.3.2 GRU model

Modeli jedinica sa zatvorenim povratnim vrednostima (*eng. Gated Recurrent Unit – GRU*) predstavljaju noviju generaciju rekurentnih neuronskih arhitektura koji su prvi put predstavljani 2014. godine (Cho et al., 2014). Ideja ovih modela je da zapravo predstavljaju dosta jednostavniju varijantu prethodno opisane LSTM arhitekture, uz zadržavanje svih njihovih prednosti. Takođe dosta se olakšala implementacija i trening ovih modela, uz dosta manje potrebnih parametara i računarskih resursa (Cho et al., 2014). Zbog svoje jednostavne arhitekture i mogućnosti da dosta ubrzaju process treniranja, ovi modeli stižu jako veliku popularnost i primenu u mnogim istraživanjima i eksperimentima koji se odnose na obradu prirodnog jezika (*eng. Natural Language Processing, NLP*), prepoznavanje govora ili predviđanje vremenskih serija (Jozefowicz, Zaremba, & Sutskever, 2015).

Jednostavnost ovih arhitektura se ogleda u tome što za razliku od klasičnih LSTM modela, GRU modeli koriste zapravo samo dve glavne kapije:

1. Reset kapija (*eng. Reset gate*) – odlučuje koliko prethodnih informacija treba ignorisati, odnosno zanemariti, a koja se može matematički predstaviti sledećom formulom:

$$r_t = \sigma(Wrxt + Urht - 1 + br)$$

Jednačina 11 Reset kapija

2. Ažurirajuća kapija (*eng. Update gate*) – odlučuje koliko trenutnih informacija treba ažurirati i zadržati za neki naredni vremenski korak, a koja se može formalizovati narednom formulom:

$$z_t = \sigma(Wzxt + Uzht - 1 + bz)$$

Jednačina 12 Ažurirajuća kapija

Takođe, za razliku od LSTM-a modela, ove arhitekture ne koriste memorijske ćelije, već imaju takozvano skriveno stanje neurona, u okviru kojih se čuvaju sve informacije. Ovo dodatno pojednostavljuje njihovu celokupnu strukturu i samim time omogućava značajno bržu propagaciju određenih informacija kroz ostatak mreže (Chung et al., 2014).

Kandidatsko skriveno stanje neurona možemo predstaviti sledećom formulom:

$$h_{\sim t} = \varphi(Whxt + Uh(rt \odot ht - 1) + bh)$$

Jednačina 13 Kandidatsko skriveno stanje neurona

Dok se ažuriranje ovog stanja može matematički formalizovati na sledeći način:

$$ht = (1 - zt) \odot ht - 1 + zt \odot h_{\sim t}$$

Jednačina 14 Ažuriranje kandidatskog stanja

Gde su:

- x_t — ulazni vektor u vremenskom koraku t
- h_{t-1} — skriveno stanje iz prethodnog koraka (vektor dimenzije n_h)
- z_t — vektor vrata ažuriranja, sa vrednostima u intervalu $[0,1]$

- r_t —vektor vrata resetovanja, sa vrednostima u intervalu $[0,1]$
- $\tilde{h}_t$ —kandidatsko (privremeno) skriveno stanje (vrednosti obično u $[-1,1]$)
- h_t —ažurirano skriveno stanje u koraku t
- W_z, W_r, W_h —težinske matrice za ulaz x_t (dimenzije $n_h \times n_x$)
- U_z, U_r, U_h —težinske matrice za prethodno skriveno stanje h_{t-1} (dimenzije $n_h \times n_h$)
- b_z, b_r, b_h —vektori bias-a (dimenzije n_h)
- $\sigma(\cdot)$ —sigmoidna aktivaciona funkcija: $\sigma(u) = \frac{1}{1+e^{-u}}$
- $\varphi(\cdot)$ —obično tanh ili druga nelinearna funkcija (često tanh)
- $\odot$ —element-po-element (*eng. Element-wise*) proizvod

U narednoj tabeli je prikazano poređenje GRU i LSTM modela na osnovu različitih parametara kao što su broj memorijskih ćelija, broj kapija u okviru svake ćelije, brzina treniranja i mnogi drugi.

Tabela 9 Poređenje LSTM i GRU modela

Karakteristika	LSTM	GRU
Broj kapija	3	2
Memorijska ćelija	Da	Ne
Složenost	Veća	Manja
Brzina treniranja	Sporije	Brže
Perfomanse na složenim sekvencama podataka	Visoke	Dovoljno dobre

4.3.3 Model rekurentnih mreža sa više glava

Rekurentne neuronske mreže sa više glava (*eng. Multi-head Recurrent Neural Networks – RNN*), predstavljaju još jedan vid arhitekture koja na neki način proširuje već postojeće modele klasičnih rekurentnih neuronskih mreža. Kako i sam naziv kaže, ova arhitektura podrazumeva da se na vrlo jednostavan način mogu obraditi duže sekvence ulaznih podataka kroz više “glava” (*eng. heads*), čime se omogućava proces paralelizacije (Luong et al., 2015). Ove “glave” zapravo predstavljaju ništa drugo no niz rekurentnih ćelija, slične onima koje se koriste u klasičnim RNN

arhitekturama, a koje se mogu konvergirati u jedinstvenu tačku ili polaziti od jednog ulaza sa više izlaznih jedinica (A. Jovanović et al., 2025).

Ovi modeli su zapravo inspirisani takozvanim mehanizmom pažnje sa više glava (*eng. Multi-head attention*), koji je prvi put predstavljen i prisutan kod arhitektura transformer modela (Vaswani et al., 2017). Ovom metodom se zapravo omogućava da model uči različite reprezentacije istih sekvenci podataka, čime se znatno lakše mogu uočiti i preciznije modelovati određeni višedimenzionalni obrasci u podacima i njihovi međusobni odnosi koji možda nisu odmah lako uočljivi. Ovako specifični obrasci se najčešće javljaju u vremenskim serijama, jer kao što je ranije napomenuto, ovi podaci nisu stacionarni već vrlo često haotični i nelinearni. Proces istovremene obrade i predviđanje više različitih faktora je još drugačije poznat pod nazivom višezadatno učenje (*eng. Multitask learning*) (A. Jovanović et al., 2025).

4.3.3.1 Arhitektura modela

Kada govorimo o arhitekturi ovih mreža, možemo zaključiti da se one sastoje od nekoliko ključnih komponenata, među kojima su:

1. Višestruke “glave” (*eng. Multiple heads*) – svaka glava je nezavisna RNN komponenta koja prima isti ili različit ulaz
2. Kombinatorni sloj (*eng. Combination layer*) – u okviru ovog sloja se kombinuju različiti izlazi iz više glava na različite načine o kojima će biti nešto više reči kasnije
3. Dekodirajući sloj (*eng. Decoding layer*) – agregirani izlazi se dalje prosleđuju u slojeve koji na svom izlazu daju konačno predikciju

Matematičkom formalizacijom ovog tipa modela dolazimo do njegove opšte formule. Dakle, ako imamo H broj glava u jednoj rekurentnoj mreži, njihov izlaz u nekom trenutku t možemo definisati kao:

$$h_t^{(head)} = \text{RNN}^{(head)}(x_t, h_{t-1}^{(head)})$$

Jednačina 15 Opšta formula RNN modela sa više glava

Ukupan izlaz cele mreže možemo predstaviti kao:

$$h_t = \text{Concat}(h_t^{(1)}, h_t^{(2)}, \dots, h_t^{(H)})$$

Jednačina 16 Izlaz mreže

Gde su:

- x_t – trenutni izlaz
- $h_t^{(head)}$ – izlaz jedne glave
- $h_{t-1}^{(head)}$ – prethodni izlaz te glave
- H – broj glava

Kao što je ranije napomenuto, ove arhitekture u svojoj osnovi imaju mehanizam pažnje (*eng. Attention mechanism*), koji ima za cilj da objedini informacije koje potiču iz više različitih izvora podataka. Takođe, on omogućava samom modelu da u različitim delovima neke ulazne sekvence identifikuje samo one segmente koji su najviše relevantni za neki konkretan zadatak koji pokušava da reši. Ovaj mehanizam se matematički može formalizovati na sledeći način:

Ako pretpostavimo da imamo određenu ulaznu sekvencu u vidu nekog vektora x :

$$x = [x_1, x_2, \dots, x_n]$$

Jednačina 17 Nasumični vektor x

Kao i da imamo q vektorski prikaz upita (*eng. Query*), onda možemo reći da mehanizam pažnje zapravo određuje stepen sličnosti između svake komponente x_i iz vektora x i vektora q , primenom funkcije poređenja $f(x_i, q)$. Na osnovu ovako dobijenih vrednosti se zatim računa distribucija verovatnoće $p(z | x, q)$, koja se dobija tako što se svi dobijeni rezultati normalizuju primenom takozvane *softmax* funkcije, i ovo je predstavljeno sledećim formulama:

$$a = [f(x_i, q)]_{i=1}^n$$

Jednačina 18 Funkcija poređenja

$$p(z | x, q) = \text{softmax}(a)$$

Jednačina 19 Primena softmax funkcije za računanje distribucije verovatnoće

$$p(z = i | x, q) = \frac{\exp(f(x_i, q))}{\sum_{i=1}^n \exp(f(x_i, q))}$$

Jednačina 20 Konačan oblik mehanizma pažnje

Mehanizam pažnje kao svoj izlaz daje ponderisanu sumu vektora x_i , gde su težine izračunate verovatnoćom $p(z = i | x, q)$. Pomoću ovako dobijenih težina, model može lako da filtrira informacije koje su bitnije, tako što se pažnja usmerava samo na one komponente koje su relevantnije za upit. Međutim, ono što je bitno naglasiti jeste da kada se ovi modeli primenjuju na nekim podacima koji nisu stacionarni, kao što su recimo vremenske serije, u okviru postojećeg mehanizma pažnje treba takođe uvrstiti i određenu dozu nelineranosti kako bi se na što bolji način reflektovala sama priroda ovih podataka. Ovako dobijamo adaptivni mehanizam pažnje, koji se od klasičnog razlikuje u implementaciji funkcije poređenja, što je prikazano sledećom formulom:

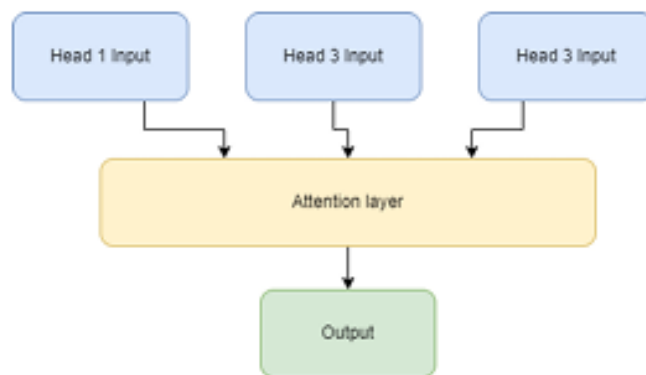
$$f(x_i, q) = w^T \sigma(W^{(1)} x_i + W^{(2)} q)$$

Jednačina 21 Adaptivni mehanizam pažnje

Gde su:

- $W^{(1)}$ – matrica težina
- $W^{(2)}$ – matrica težina
- $\sigma(\cdot)$ – nelinearna funkcija aktivacije
- w – vektor težina

Prikaz jednostavne arhitekture ovog modela dat je na sledećoj ilustraciji:



Ilustracija 8 Arhitektura višeglavnih modela

4.3.3.2 Prednosti i nedostaci

Prilikom korišćenja ovih arhitektura neuronskih mreža u brojnim istraživanjima, možemo uočiti mnogobrojne prednosti, kao što su paralelno učenje različitih obrazaca iz podataka, znatno veća robusnost modela, fleksibilnost, kao i bolja preciznost prilikom generalizacije (Zhang et al., 2022). Ono što možemo definisati kao glavne nedostatke ovih modela jeste pre svega potreba za tačnom hiperparametrizacijom, kao što je na primer što precizniji izbor broja glava, zatim relativno skupa i nešto komplikovanija implementacija.

I pored nekih određenih nedostataka, ovi modeli su postali jako popularni u modernim oblastima mašinskog učenja. Neki od poznatijih modela koji se danas vrlo često koriste, a koji se baziraju na ovim arhitekturama su BERT (*eng. Bidirectional Encoder Representations from Transformers*), koji se uspešno primenjuje prilikom procesa obrade prirodnih jezika (A. H. Mohammed et al., 2021), kao i YOLO (*eng. You Only Look Once*), koji predstavlja jedno od najefikasnijih rešenja u oblasti računarskog vida (P. Jiang et al., 2022).

4.4 Problem odabira hiperparametara u dubokom učenju

Imajući u vidu složenost modela dubokog učenja, a pogotovu rekurentnih neuronskih mreža, možemo zaključiti da je adekvatan izbor hiperparametara jedan od najbitnijih zadataka prilikom implementacije ovih modela.

Hiperparametri ne predstavljaju rezultat težina koje je model naučio tokom treniranja, već vrednosti koje programmer sam unapred definiše prilikom inicijalne implementacije, a pre samog procesa treniranja (Bengio, 2012). Kod različitih podvrsta rekurentnih neuronskih mreža, a pogotovu kod mreža sa više glava, broj hiperparametara može eksponencijalno da poraste zbog same paralelizacije, koja je, kao što je ranije napomenuto, njihova glavna karakteristika. Stoga, u okviru ovih modela možemo izdvojiti nekoliko različitih vrednosti koje programeri definišu u početnim fazama implementacije, kao što su recimo broj i tip glava, dužina sekvence ulaznih podataka, dimenzije skrivenih slojeva svake glave, način agregacije izlaza iz pojedinačnih glava i noge druge. Ukoliko se ovi parametri ne iskonfigurišu dovoljno precizno, može doći do brojnih problema prilikom generalizacije, kao što su prenaučenosć ili nestabilne konvergencije (Goodfellow et al., 2016).

Na osnovu navedenog, možemo uočiti da adekvatni izbor hiperparametara na neki način igra ulogu “kontrolnih” poluga koje generalno upravljaju treningom i određuju balans između preciznosti, brzine treniranja i konačne sposobnosti generalizacije nad novim podacima. Imajući ovo u vidu, optimizacija hiperparametara igra ključnu ulogu prilikom dizajniranja i implementacije znatno kompleksnih modela, pogotovu kada govorimo o njihovoj primeni na vremenskim serijama, koje su same po sebi dinamični tipovi podataka sa jako dugim sekvencama.

Kao i većina procesa koji su do sada spomenuti, podešavanje hiperparametara se može odvijati ručno, ali se takođe može i automatizovati. Ručno podešavanje ovih hiperparametara može biti korisno u nekim eksperimentalnim situacijama ili kada su modeli jednostavni i ne treniraju se nad većim skupom podataka. Međutim, prilikom rešavanja nekih značajnih i realnih problema, ovaj proces nije dovoljno skalabilan i dovoljno teško se praktično mora uključiti i faktor ljudske greške. Kako bi se ovi izazovi prevazišli, došlo je do pojave nove oblasti koja se specifično bavi automatizovanim pristupom optimizacije hiperparametara koristeći razne metode i algoritme (Feurer & Hutter, 2019) o kojima će biti nešto više reči u narednom poglavlju.

5 Algoritmi optimizacije hiperparametara

U okviru ovog poglavlja, biće dat pregled različitih metoda za optimizaciju hiperparametara sa posebnim akcentom na:

1. Osnovne heurističke tehnike
2. Napredne metaheurističke tehnike

Pored ovoga biće predstavljen i sam algoritam optimizacije rojevima čestica (*eng. Particle Swarm Optimiser, PSO*) koji je ujedno i predloženi način optimizacije hiperparametara korišćen u istraživačkom delu ove disertacije.

5.1 Heurističke tehnike optimizacije

Analizom dostupne literature, može se zaključiti da oblast mašinskog učenja gotovo paralelno prati i problematika adekvatnog izbora hiperparametara. U nekim prvim godinama razvoja ove oblasti, podešavanje hiperparametara se najčešće izvodilo manuelno, u glavnom na osnovu iskustva samog istraživača i testiranja različitih vrednosti odabranih parametara. Kako je ranije napomenuto, ovaj pristup se u nekim složenim scenarijima koji su zahtevali rešavanje dosta kompleksnijih problema pokazao neefikasnim, jer je porastom složenosti samog problema, rastao i broj mogućih kombinacija optimalnih parametara (Bergstra & Bengio, 2012).

Uviđanje ovih glavnih nedostataka ovakvog načina izbora hiperparametara, dovelo je do uvođenja prvih sistematskih pristupa, poput heurističkih metoda, koje su korišćene kao optimalno rešenje za prevazilaženje brojnih izazova sa kojima su se manuelni pristupi suočavali. Tokom devedesetih godina prošlog veka, dva algoritma Mrežna pretraga (*eng. Grid search*) i Nasumična pretraga (*eng. Random search*) uvedeni su kao jednostavne i efikasne tehnike za pretragu prostora parametara. U okviru njega se nalazile optimalne kombinacije parametara koje dovode do boljih performansi i preciznosti samog modela (Bergstra & Bengio, 2012). Međutim, kako je vreme prolazilo, došlo je do povećanja kompleksnosti i otkrivanja novih modela mašinskog učenja.

Heurističke metode optimizacije hiperparametara predstavljaju značajno jednostavne, ali vrlo efikaksne načine u rešavanju ovog problema. One su posebno pogodne kada se primenjuju u

situacijama gde matematički formalizovan model nije dostupan, ili kada je prostor pretrage prevelik da bi bio dovoljno iscrpno analiziran (Kirkpatrick et al., 1983). Prvim primenama ovih modela došlo se do generalnog zaključka da one igraju ključnu ulogu u poboljšavanju performansi učenja modela i da dosta mogu uticati na preciznost konačnih predikcija, što je dovelo do velike popularnosti u daljem naučnom istraživanju ove oblasti.

Generalno prve upotrebe heurističkih metoda se javljaju još početkom pedesetih godina prošlog veka, u okviru posebnih istraživanja koji su sprovedeni sa ciljem rešavanja industrijskih i inženjerskih problema (Kirkpatrick et al., 1983). Nešto kasnije, eksponencijalnim rastom popularnosti mašinskog učenja, dolazi i do prvih implementacija heurističkih metoda. One su najčešće korišćene za rešavanje problema optimalnog izbora hiperparametara, koje se do tada, kao što je ranije opisano, rešavalo isključivo manuelnim putem. Kako su vremenom modeli, ali i sami problemi koji su se rešavali postali složeniji, ovakav pristup je postajao sve manje i manje održiv, a verovatnoća ljudske greške tokom ovog procesa sve veća i veća. Kako bi se nadomestili nedostaci ovog pristupa, dolazi do primene dveju relevantnih heurističkih metoda, koje i danas predstavljaju predmet brojnih istraživanja, a to su:

1. Mrežna pretraga (*eng. Grid search*)
2. Nasumična pretraga (*eng. Random search*)

Pored rešavanja mnogobrojnih problema manualnog pristupa odabira optimalnih hiperparametara, ove dve metode su takođe predstavila i naučnu osnovu za razvoj dosta naprednijih pristupa, kao što su na primer Bajesova optimizacija, genetski algormi optimizacije i algoritmi rojeva čestica (Bergstra & Bengio, 2012; Snoek et al., 2012).

Iako su obe metode ostvarile svoj veliki potencijal na naučnom polju, dosta dugo je tehnika mrežne pretrage bila popularnija, zbog svoje jednostavne implementacije i lakog razumevanja. Međutim, nešto kasnije, brojnim dodatnim istraživanjima i analizama utvrđeno je da i metoda nasumične pretrage takođe ostvaruje dosta dobre rezultate u praksi (Bergstra & Bengio, 2012).

5.1.1 Mrežna pretraga

Algoritam mrežne pretrage (*eng. Grid search*) predstavlja prvi značajni oblik sistematskog pristupa u optimizaciji hiperparametara. Osnovna ideja se bazira na tome da postoji neki unapred definisan skup vrednosti, koji se zatim sistematski pretražuje za svaki hiperparametar kako bi se pronašla kombinacija koja na najoptimalniji način poboljšava celokupne performanse nekog modela (Bergstra & Bengio, 2012). Na ovaj način se u okviru prostora pretrage prave sve moguće kombinacije vrednosti, formirajući na taj način mrežu, koju sam algoritam pretražuje.

Ovo možemo demonstrirati na jednostavnom primeru. Ako recimo posmatramo dva hiperparametra, stopu učenja (*eng. learning rate*) i broj neurona u nekom sloju – a za svaki od njih definišemo po, recimo, tri moguće vrednosti, ovaj algoritam će praktično napraviti mrežu u okviru koje će evaluirati ukupno devet različitih modela, testirajući sve moguće kombinacije ovih parametara. Ovo dalje možemo matematički formalizovati koristeći narednu formulu:

$$\theta^* = \arg \max_{\theta_1 \in V_1, \theta_2 \in V_2, \dots, \theta_n \in V_n} \text{Score}(\theta_1, \theta_2, \dots, \theta_n)$$

Jednačina 22 Mrežna pretraga

Gde su:

- θ_i — hiperparametar i-tog tipa
- V_i — skup vrednosti za θ_i
- θ^* — optimalna kombinacija hiperparametara
- $\text{Score}(\dots)$ — metrička funkcija koja meri performanse modela (npr. tačnost, F1-score)

Kada govorimo generalno o prednostima ovog pristupa, možemo izdvojiti sledeće:

1. Jednostavna implementacija
2. Programerima se omogućava dosta detaljan uvid u samu osetljivost modela na adekvatan izbor hiperparametara
3. Pokazuje izuzetno dobre performanse prilikom rešavanja nekih problema sa relativno malim brojem parametara

Kao glavne nedostatke, možemo izdvojiti sledeće:

1. Eksplozivna složenost koja raste zajedno i sa brojem hiperparametara
2. Značajno mala prilagodljivost

5.1.2 Nasumična pretraga

Kako i sam naziv kaže, ova metoda nasumične pretrage (*eng. Random search*) podrazumeva da se nasumično biraju kombinacije različitih hiperparametara iz nekih unapred definisanih distribucija. Umesto iscrpne pretrage mrežnog prostora, ova tehnika se bazira na stohastičkom uzorkovanju prostora (Bergstra & Bengio, 2012).

Stohastički pristupi zapravo predstavljaju posebne klase algoritama koje u okviru sebe uključuju i određene elemente slučajnosti. Matematički gledano, za razliku od determinističkih metoda koje za isti ulaz uvek generišu isti rezultat, stohastičke metoda mogu proizvesti različite ishode zbog nasumičnih komponenti koje koriste prilikom svog izvršavanja.

Ako uzmemo isti primer kao u prethodnoj sekciji sa parametrima stope učenja i brojem neurona, ali pod pretpostavkom da želimo da testiramo pet različitih kombinacija – ovaj algoritam će iz dostupnog prostora izabrati nasumično pet tačaka. Na ovaj način se na postojeće vrednosti parametara mogu dodati i vrednosti iz različitih delova prostora koji se pretražuje, te se tako mogu ispitati i određene oblasti koje bi algoritam mrežne pretrage možda propustio.

Ovaj algoritam možemo matematički formalizovati korišćenjem sledeće formule:

$$\theta^* = \arg \max_{\theta \sim P(\Theta)} \text{Score}(\theta)$$

Jednačina 23 Nasumična pretraga

Gde su:

- θ — vektor hiperparametara
- $P(\Theta)$ — distribucija iz koje se nasumično uzorkuju vrednosti hiperparametara
- $\text{Score}(\theta)$ — metrička funkcija (npr. tačnost, F1-score, RMSE...)
- θ^* — kombinacija hiperparametara koja daje najbolji rezultat

Kao glavne prednosti ovog modela, možemo izdvojiti sledeće:

1. Veća efikasnost u poređenju sa mrežnom pretragom kada se primenjuje nad većim dimenzijama (Bergstra & Bengio, 2012)
2. Relativno brzo nalaženje optimalnih parametara u nekim realnim uslovima

U neke ključne nedostatke ovog pristupa, možemo uključiti sledeće:

1. Rezultati mogu pretežno da zavise od faktora sreće, zbog svoje stohastičke prirode
2. Nema sigurnih garancija da će optimalna vrednost uvek biti pronađena
3. Zbog ovoga je teže analizirati osetljivost modela na odabir vrednosti hiperparametara

U narednoj tabeli je dat detaljniji prikaz analize oba pristupa, zajedno sa prednostima i nedostacima u korišćenju istih.

Tabela 10 Analiza mrežne i nasumične pretrage

Osobina	Mrežna pretraga	Nasumična pretraga
Efikasnost u 2D i 3D prostorima	Sistematska	Slučajna
Rizik promašaja	Srednji	Srednje-visok
Lakoća implementacije	Veoma lako za implementaciju	Veoma lako za implementaciju
Podesivost distribucija	Zbog svoje determinističke prirode nema dodanu podesivost	Ima
Osetljivost na parametre	Visoka	Može biti neujednačena zbog slučajnosti

Korišćenje obe metode sa sobom povlači određene prednosti i rizike, ali u svojoj suštini one predstavljaju ključnu osnovu za razumevanje modernijih metoda i pristupa u optimalnom podešavanju hiperparametara, kao što su recimo metaheurističke metode, o kojima će biti nešto više reči u narednom poglavlju. Implementacijom klasičnih, heurističkih metoda, značajno je porasla popularnost automatizovanom pristupu prilikom odabira hiperparametara, čime su se

drastično poboljšale performanse i preciznost mnogih modela, pogotovu u kontekstu predviđanja vremenskih serija.

5.2 Metaheurističke tehnike

Metaheuristike zbirno predstavljaju posebnu podoblast optimizacije hiperparametara, koje se baziraju na određenom skupu algoritama i njihovom korišćenju u rešavanju određenih kompleksnijih problema, koje klasične heurističke metode ne mogu rešiti (Talbi, 2009). Ovi problemi se mogu javiti kada govorimo o obradi određenih skupova podataka koji su nelinearni, diskontinuirani ili multimodalni. Metaheuristike u ovim slučajevima predstavljaju logičan zbor, prvenstveno zbog svog opšteg karaktera, te mogućnosti da se primene u rešavanju širokog dijapazona problema (Blum & Roli, 2003). Sam termin metaheuristika (*eng. Metaheuristics*) je tokom osamdesetih godina prošlog veka popularizovao Fred Glover (*eng. Fred Glover*) (Glover, 1986).

Ono što je glavna karakteristika ovih algoritama jeste to da su oni inspirisani biološkim pojavama, kao što su evolucija ili kolektivno ponašanje većeg skupa životinja, kao što su jata, kolonije, krda i mnogi drugi, kao određene fizičke simulacije (Dorigo & Stützle, 2004; Kennedy & Eberhart, 1995).

Kao ključne karakteristike ovih modela, možemo izvodjiti:

1. Stohastička priroda – koriste slučajnost u generisanju i selekciji optimalnog rešenja
2. Globalna pretraga – izbegavaju lokalne minimum pomoću strategija eksploracije
3. Ne zavise od gradijenata – ovo ih posebno čini primenljivim u rešavanju problema sa skupovima podataka koji su nelinearni
4. Veća fleksibilnost i robusnost – vrlo lako se prilagođavaju različitim tipovima problema i mogu davati prilično stabilna rešenja

5.3 Pregled algoritma čestica rojeva

5.3.1 Teorijska osnova

Algoritam čestica rojeva (*eng. Particle Swarm Optimizer – PSO*) predstavlja jedan od popularnijih i često korišćenih metaheurističkih algoritama. Svoju primenu je pronašao u rešavanju brojnih optimizacionih problema (Kennedy & Eberhart, 1995), a osmišljen je i dizajniran od strane Džejms Kenedija (*eng. James Kennedy*) i Rasel Eberharta (*eng. Russel Eberhart*) tokom devedesetih godina prošlog veka. Ono što je glavna karakteristika ovog algoritma jeste to da je on u potpunosti osmišljen tako da simulira biološko ponašanje zajednice ili populacije u prirodi, kao što su recimo jata ptica ili rojevi insekata, tokom procesa nalaženja hrane ili plena.

Ono što razlikuje ovaj algoritam od ostalih je to da on ne koristi mutaciju ili ukrštanje. Umesto toga, simulirajući biološko kretanje neke populacije, modeluje kretanje posebnih čestica ili agenta kroz određeni prostor rešenja ili pretrage. Svaki agent podešava svoju poziciju u skladu sa sopstvenim iskustvom i iskustvom ostalih čestica, odnosno agenata iz roja (populacije). Na ovaj način se celokupan prostor rešenja efikasno pretražuje, a takođe se balansiraju eksploracija i eksploatacija.

Kada govorimo o eksploraciji, u ovom kontekstu je možemo definisati kao sposobnost čestica da istražuju različite oblasti datog prostora pretrage. Na ovaj način se znatno povećava verovatnoća pronalaženja globalnog optimuma. Suprotno ovome, eksploatacija predstavlja fokusirano pretraživanje u okviru već nađenih optimalnih rešenja. Ovime se dosta dobro optimizuju već postojeće pozicije agenata u okviru prostora rešenja, te se samim time poboljšava i kvalitet dobijenog rešenja (Kennedy & Eberhart, 1995).

5.3.2 Matematički model

Model PSO algoritma možemo matematički formalizovati koristeći formule za predstavljanje ažuriranje brzine i konkretne pozicije čestica u okviru dostupnog prostora rešenja.

Ažuriranje brzine čestice možemo predstaviti sledećom formulom:

$$v_i^{t+1} = w \cdot v_i^t + c_1 \cdot r_1 \cdot (pbest_i - x_i^t) + c_2 \cdot r_2 \cdot (gbest - x_i^t)$$

Jednačina 24 Ažuriranje brzine čestice

Ažuriranje pozicije čestice se može predstaviti na sledeći način:

$$x_i^{t+1} = x_i^t + v_i^{t+1}$$

Jednačina 25 Ažuriranje pozicije čestice

Gde su:

- x_i^t - trenutna pozicija i-te čestice u iteraciji t
- v_i^t - trenutna brzina i-te čestice u iteraciji t
- $pbest_i$ - najbolja pozicija do sada i-te čestice
- $gbest$ - globalno najbolja pozicija svih čestica
- W - težinski faktor inercije (*eng. inertia weight*)
- c_1, c_2 - koeficijenti učenja (*eng. cognitive components*)
- r_1, r_2 - slučajne vrednosti u intervalu [0,1]

5.3.3 Prednosti i nedostaci

Tabela 11 Prednosti i nedostaci PSO algoritma

Prednost	Nedostatak
Jednostavna implementacija	Rizik od lokalnog minimuma
Brza konvergencija	Podešavanje hiperparametara
Pogodan za optimizovanje složenijih funkcija	Perfomanse mogu opadati sa porastom dimenzionalnosti prostora
Fleksibilnost i skalabilnost	
Balans eksploatacije i eksploracije	

5.3.4 Primena u predikciji vremenskih serija

Na osnovu prethodno navedenog može se zaključiti da ovaj algoritam nalazi svoju najveću primenu u optimizaciji modela koji se primarno koriste za optimizaciju vremenskih serija, što je ujedno i glavni predmet istraživanja ovog rada.

Najčešći oblik primene algoritma rojeva čestica u kontekstu predviđanja vremenskih serija je optimizacija hiperparametara modela zasnovanih na rekurentnim neuronskim mrežama, kao što su LSTM, GRU i klasične rekurentne neuronske mreže. Među ključnim hiperparametrima koji se potencijalno mogu dodatno optimizovati spadaju broj slojeva, stopa učenja (*eng. Learning rate*), dropout i slično (Zhang et al., 2018).

Druga glavna stvar oko koje nam ovaj algoritam može značajno pomoći jeste izbor prave arhitekture neuronskih mreža, ukoliko se ovi modeli primarno koriste za predikciju vremenskih serija. Dalje, ovaj algoritam se može dosta lako ukombinovati i sa nekim drugim algoritmima ili modelima mašinskog učenja, gde je najpoznatija kombinacija PSO-LSTM hibridnog modela. Ovo dosta olakšava istraživački proces kada se radi sa velikom količinom podataka poput vremenskih serija. Korišćenje ovakvog algoritma značajno utiče na robusnost arhitekture modela i povećava mogućnost da se uhvate neki nelinearni ili sezonski obrasci koji su dosta učestali, pogotovu kada govorimo o predikciji cena goriva (Huang et al., 2020). Takođe vrlo lako se može automatizovati proces pretrage optimalne konfiguracije modela što je velika prednost kada se radi sa podacima koji imaju realitvno visoku volatilnost.

Na osnovu navedenog, može se zaključiti da je ovaj algoritam koji vrlo efikasan, fleksibilan i lak za implemetnaciju i kombinaciju sa nekim drugim modelima i / ili arhitekturama neuronskih mreža. Kada se upotrebljava u istraživanjima koji se baziraju na predikciji vremenskih serija, konkretno u ovom slučaju, predviđanju cena goriva, značajno se dobija na performansama i na većoj preciznosti modela da prediktuju neke očekivane rezultate.

6 Pregled istraživačkog dela

Nakon teorijske obrade i analize oblasti koje prate istraživački deo ove disertacije, u ovom poglavlju biće predstavljeno samo istraživanje, podaci, kao i opis eksperimenata koji su izvršeni i rezultati koji su dobijeni izvođenjem istih. Kao deo samog zaključka, biće dat kratak osvrt na hipoteze opisane na samom početku ovog rada i kako su zapravo ovi eksperimenti, odnosno dobijeni rezultati pomogli u njihovom dokazivanju.

6.1.1 Opis istraživanja

Kao što je u prethodnim poglavljima napomenuto, pod pritiskom brojnih i turbulentnih socio-ekonomskih faktora, znatno se povećala potražnja za dostupnim energetske resursima. Ovakva dinamika može direktno uticati i na značajne oscilacije u cenama dostupnih energenata, kao i do smanjenja dostupnih zaliha fosilnih goriva. Kako bi se pravovremeno donele ispravne poslovne odluke, kao i automatizovali određeni procesi radi povećanja agilnosti brojnih svetskih kompanija koje praktično kontrolišu svetsko tržište nafte i ostalih goriva, popularizovana su brojna istraživanja u oblasti predviđanja potrošnje energije (Zhang et al., 2018; Huang et al., 2020).

Već uveliko razvijena oblast Veštačke inteligencije, a posebno predviđanje vremenskih serija, čuvaju veliki i još uvek nedovoljno istraženi potencijal za unapređenjem prognoza troškova i potražnje na globalnom energetske tržištu. Analizom postojeće literature uočen je značajni prostor za dodatno poboljšanje postojećih modela kroz primenu naprednijih arhitektura nerunskih mreža, kao i tehnika optimizacije.

Imajući ovo u vidu, ovaj rad predlaže nov pristup u rešavanju problema predviđanja cena goriva, istraživanjem potencijala koje imaju višeglavne LSTM (*eng. Long Short-Term Memory*) mreže, a s obzirom na to da ovaj specifičan problem nije ranije razmatran korišćenjem ovakvog modela. Takođe, imajući u vidu da su računarski zahtevi za treniranje i implementaciju ovakvih modela izuzetno zahtevni, rad se fokusira na primenu nešto efikasnijih pristupa koji uključuju malo drugačiju arhitekturu (manji broj neurona po sloju), kao i treniranje u manjem broju epoha.

Međutim, s obzirom na to da na performanse značajno može uticati i izbor adekvatnih hiperparametara, ovaj rad predlaže implementaciju modifikovane varijante algoritma za

optimizaciju koji se bazira na roju čestica (*eng. Particle Swarm Optimizer, PSO*), kako bi se što bolje pomoglo u dodatnoj optimizaciji same arhitekture modela i parametara treniranja, bez gubitka u preciznosti konačnih rezultata. Razlog za izbor baš ovog optimizacionog algoritma zapravo leži u Wolpertovom (*eng. Wolpert*) teoremu o nepostojanju univerzalnog rešenja (*eng. No-free lunch theorem*), koji navodi da ne postoji univerzalno superioran algoritam koji daje najbolje rezultate za svaki optimizacioni problem (Wolpert & Macready, 1997). Zbog toga je neophodno eksperimentisati sa više optimizacionih algoritama, a kako je ovaj algoritam postigao veoma ohrabrujuće rezultate tokom eksperimenata manjeg obima, izabran je za dalje modifikacije kako bi se postigli još bolji rezultati u rešavanju problema predviđanja cena goriva.

Kada govorimo o konačnim rezultatima ovog istraživanja, kako bi se oni što jasnije sagledali i kako bi se što bolje razumeo doprinos primene ovakvog modela, sprovedena je i komparativna analiza korišćenjem javno dostupnih podataka o energiji i gorivima, a nad kojima je takođe primenjeno nekoliko savremenih optimizacionih algoritama. Dobijeni rezultati su prošli kroz detaljnu statističku validaciju kako bi se potvrdila značajnost dobijenih nalaza.

Dakle, naučni doprinos ovog istraživanja možemo kategorisati na sledeći način:

- Predlog novog pristupa za predviđanje cena goriva, koji se bazira na višeglavnim LSTM modelima, koji do sada nije bio validiran u kontekstu rešavanja ovog problema.
- Implementacija modifikovane varijante dobro poznatog i istraženog optimizacionog algoritma koji se bazira na rojevima čestica, koji je dodatno prilagođen potrebama ovog istraživanja.
- Primena tehnika objašnjive veštačke inteligencije poput XAI (*eng. Explainable AI*) kako bi se što jasnije sagledali i razumeli faktori koji utiču na fluktuaciju cena goriva na globalnom nivou.

6.1.2 Korišćene procedure i tehnike

U ovom poglavlju detaljno su opisane primenjene metode i pristupi korišćeni u istraživanju sa ciljem efikasnijeg predviđanja promena u cenama goriva. Akcenat je stavljen na razvijanje modela zasnovanog na višeglavnoj LSTM arhitekturi optimizovanoj tako da omogući visoku tačnost uz nisku računarsku zahtevnost za treniranjem i validacijom modela. Pored toga,

uvedena je unapređena verzija algoritma rojem čestica (PSO), osmišljena da poboljša izbor optimalnih hiperparametara i time dodatno doprinese performansama predloženog modela. U cilju boljeg razumevanja ključnih faktora koji utiču na cene, primenjene su i tehnike objašnjive inteligencije (eng. *Explainable AI*, XAI), koje omogućavaju transparentnije tumačenje generisanih odluka i rezultata samog modela.

6.1.2.1 Modifikovani algoritam optimizacije rojem čestica

Kao što je ranije napomenuto, jedan od glavnih nedostataka klasičnog algoritma optimizacije rojem čestica je njegova tendencija ka sporoj konvergenciji u kasnijim iteracijama njegovog izvršavanja.

Predložena varijanta algoritma koja je korišćena u ovom istraživanju dodatno unapređuje postojeći algoritam tako što kombinuje elemente osnovnog oblika optimizacije rojem čestica sa principima genetskih algoritama, kao što su uniformno ukrštavanje (eng. *Uniform crossover*) i mutacija (eng. *Mutation*) i dodatnih adaptivnih mehanizama, kako bi se značajno poboljšala efikasnost pretrage prostora rešenja i kvalitet dobijenih rezultata.

Ako sumiramo neke ključne karakteristike modifikovane varijante ovog optimizacionog algoritma, možemo ih predstaviti na sledeći način:

- Inicijalizacija koristeći kvazi-reflektivno učenje (eng. *Quasi reflective learning*, *QRL*) koja omogućava da početna populacija ima bolju raznolikost, što smanjuje rizik od preranog konvergiranja ka lokalnom optimum
- Uvođenje genetskih operatora (ukrštavanje i mutacija) kako bi se kombinovala postojeća rešenja, na takav način da podstiče stvaranje novih, boljih rešenja, čime se značajno povećava istraživačka sposobnost algoritma
- Adaptivno podešavanje parametara, kao što su recimo faktor za ažuriranje brzine i težine iteracije, koje podrazumeva da se oni sami prilagođavaju tokom istraživanja u zavisnosti od nekih drugih faktora, kao što su raznolikost populacije
- Primena mehanizma za zamenu rešenja zasnovanog na L1 normi, koji ima za cilj očuvanje raznovrsnosti populacije i sprečavanje stagnacije istog, tako što se pravovremeno zamenjuju slična ili redundantna rešenja nekim novim kandidatima

Uzimajući sve ovo u obzir, možemo reći da ova predložena modifikovana verzija PSO algoritma omogućava dosta dublju i detaljniju pretragu dostupnog prostora rešenja, istovremeno održavajući stabilnost same konvergencije, što je, kako je u prethodnim poglavljima napomenuto, ključno kada govorimo o rešavanju određenih problema koji se baziraju na predikciji vremenskih serija.

6.1.2.2 Procedura inicijalizacije zasnovana na kvazi-reflektivnom učenju

Generalno govoreći, kvazi reflektivno učenje (eng. *Quasi-Reflexive Learning, QRL*) možemo definisati kao jednu od strategija inicijalizacije populacije kod optimizacionih algoritama. Glavni cilj ove procedure jeste povećanje raznolikosti same početne populacije i smanjenje verovatnoće brzog zapadanja u lokalne minimume.

U okviru ovog istraživanja, QRL se primarno koristi za inicijalizaciju čestica u modifikovanom PSO algoritmu, kako bi se dodatno proširio prostor pretrage, bez dodatnog troška u pogledu broja evaluacija fitness funkcije.

Standarni koraci QRL inicijalizacije se mogu definisati kao:

1. Generisanje osnovnih čestica – nasumično generisanje početne populacije unutar granice definisanog prostora rešenja
2. Kreiranje kvazi-reflektivne čestice:

Za svaku osnovnu česticu x_i , generiše se reflektivna čestica x_i' po formuli:

$$x_i' = L + U - x_i = U - (x_i - L) = L + (U - x_i)$$

Jednačina 26 Definicija reflektivne čestice

gde su L i U donje i gornje granice prostora pretrage

3. Izbor između osnovne i reflektivne čestice – za svaku poziciju algoritam bira onu koja ima bolju početnu vrednost svoje fitnes funkcije, čime se na samom početku može garantovati kvalitetniji početni skup, odnosno populacija
4. Formiranje početne populacije – korišćenjem kombinacije inicijalnih i reflektivnih pozicija koje daju bolje rezultate, formira se raznovrsnija početna populacija

Na osnovu ovoga možemo zaključiti da ova tehnika umesto nasumične inicijalizacije svake čestice unutar definisanog prostora rešenja, generiše reflektivno suprotne komponente za već postojeće čestice. Na ovaj način se može postići bolja pokrivenost dostupnog prostora rešenja od samog početka, što može pozitivno uticati na ukupnu efikasnost, stabilnost i preciznost optimizacionog procesa.

Za svaku komponentu X_j se određuje njena kvazi-refleksno suprotna komponenta X_j^{qr} , što možemo predstaviti sledećom formulom:

$$X_j^{qr} = rnd \left(\frac{lb_j + ub_j}{2}, X_j \right)$$

Jednačina 27 Kvazi-refleksno suprotna komponenta za svaku komponentu X_j

U okviru koje:

- $rnd(\cdot)$ označava slučajan broj u opsegu između srednje vrednosti granica $\frac{lb_j + ub_j}{2}$ i komponente X_j

Ovim pristupom ne dolazi do povećanja računarskog opterećenja u smislu evaluacija fitnes funkcije, jer se inicijalno generiše samo NP/2NP/2NP/2 rešenja, a ostatak populacije se dobija putem QRL mehanizma.

6.1.2.2.1 Pseudokod za QRL inicijalizacioni mehanizam

Algoritam 1 *Pseudokôd mehanizma inicijalizacije zasnovanog na QRL-u*

Korak 1: Sintetisati populaciju $P^{(init)}$ sa NP/2 rešenja korišćenjem jednačine (18)

Korak 2: Generisati QRL populaciju $P^{(qr)}$ na osnovu $P^{(init)}$ primenom jednačine (19)

Korak 3: Formirati početnu populaciju P kao uniju $P^{(init)}$ i $P^{(qr)}$ ($P \cup P^{(qr)}$)

Korak 4: Izračunati fitnes vrednost za svako rešenje koje pripada populaciji P

Korak 5: Poređati sve jedinke iz populacije P na osnovu njihove fitnes vrednosti

6.1.2.3 Procedura za procenu raznolikosti populacije

Kada govorimo o samom procesu pretrage dostupnog prostora rešenja kod metaheurističkih optimizacionih algoritama, možemo reći da se on efektivno može ispratiti analizom raznolikosti populacije. Kao što je ranije napomenuto, u ovom istraživanju je korišćena L1 norma, kao mera raznolikosti. Razlog zbog čega je baš ova norma izabrana, umesto L2 norme recimo, leži u tome što L1 norma uzima u obzir i udaljenost između dostupnih rešenja u populaciji kao i dimenzionalnost samog optimizacionog problema.

Dodatne razloge koji idu u prilog L1 normi možemo kategorisati na sledeći način:

- Podsticanje retkosti (*eng. sparseness*) u rešenjima
- Linearni penali koji se koriste za prevelika odstupanja, što ovu tehniku čini dosta otpornijom na određene ekstremne vrednosti (*eng. outliers*), što nije karakteristično za L2 normalizaciju
- L1 norma je dosta jednostavnija za razumevanje i implementaciju
- U nekim visoko-dimenzionim prostorima rešenja, L1 norma upravo zbog svoje jednostavnosti, vrlo često može obezbediti stabilnija i praktičnija rešenja.

Ako je m broj rešenja u populaciji, a n broj dimenzija, L1 norma se računa prema sledećim formulama:

1. Prosečna vrednost za dimenziju j :

$$\bar{x}_j = \frac{1}{m} \sum_{i=1}^m x_{ij}$$

Jednačina 28 Prosečna vrednosti dimenzije j

2. Diverzitet po dimenziji:

$$D_j^p = \frac{1}{m} \sum_{i=1}^m |x_{ij} - \bar{x}_j|$$

Jednačina 29 Diverzitet po dimenziji j

3. Ukupna raznolikost populacije:

$$D^p = \frac{1}{n} \sum_{j=1}^n D_j^p$$

Jednačina 30 Ukupna raznolikost populacije

Gde su:

- $\bar{x}$ – srednja vrednost pozicija svih jedinki u svakoj dimenziji
- D_j^p – mera raznolikosti pozicija u nekoj dimenziji
- D^p – skalarni indikator ukupne raznolikosti populacije

U ranim fazama izvršavanja algoritma, poželjno je da raznolikost bude visoka, kako bi se istražio što širi prostor rešenja. Kako algoritam napreduje i konvergira ka optimalnim oblastima, raznolikost treba dinamički smanjivati. Unapređeni PSO algoritam koristi L1 normu za upravljanje raznolikošću, u kombinaciji sa dinamičkim pragom D_t .

Dodatni kontrolni parametar označen kao **nrs** (broj rešenja za zamenu), u ovom istraživanju empirijski je postavljen na 2. Na početku izvršavanja izračunava se početna vrednost praga D_{t0} prema formuli:

$$D_{t0} = \sum_{j=1}^{NP} \frac{(ub_j - lb_j)}{2 \cdot NP}$$

Jednačina 31 Računanje početne vrednosti praga D_{t0}

Ova formula polazi od pretpostavke da su početne vrednosti većinom generisane blizu srednje vrednosti između donje i gornje granice parametra.

Tokom svake iteracije algoritma, meri se trenutna vrednost diverziteta D_p , i poredi sa pragom D_t :

- Ako je $D_p < D_t$: (npr. raznolikost je niska), najlošija 2 rešenja se zamenjuju novim hibridnim rešenjem, dobijenim primenom uniformnog kros-overa i mutacije između najlošijeg rešenja i slučajno odabranog člana populacije (prema standardnoj inicijalizaciji).

- Ako je $D_p \geq D_t$, znači da je diverzitet previsok. U tom slučaju se takođe formira hibridno rešenje između najboljeg jedinke i nasumičnog člana populacije, takođe korišćenjem kros-overa i mutacije — čime se diverzitet postepeno smanjuje.

Vrednost praga D_t se dinamički smanjuje u svakoj iteraciji po sledećem izrazu:

$$D_{t+1} = D_t - D_t \cdot \frac{t}{T}$$

Jednačina 32 Smanjivanje vrednosti praga D_t

Gde su:

- t i $t + 1$ trenutna i naredna iteracija, a T ukupan broj iteracija.

Kako se algoritam približava kraju, vrednost D_t se približava nuli, i ova procedura se prestaje primenjivati, bez obzira na trenutnu raznolikost D_p .

6.1.2.4 Operatori kros-overa i mutacije

Korišćenje ovih operatora proizilazi iz inspiracije postojećeg genetskog algoritma (*eng. Genetic Algorithm, GA*). Ovi operatori predstavljaju vrlo bitno stavku kada govorimo u dodatnom unapređenju populacije. Vrednost verovatnoće kros-overa po genu označena je kao p_c . Kada se kros-over primeni, generišu se dva nova potomka, koji zatim zamenjuju **nrs = 2** postojeća rešenja u populaciji — bez dodatne evaluacije njihove fitnes vrednosti.

Na početku algoritma, početna vrednost p_c je postavljena na **0.1**, a zatim se u svakoj iteraciji postepeno povećava prema sledećoj formuli:

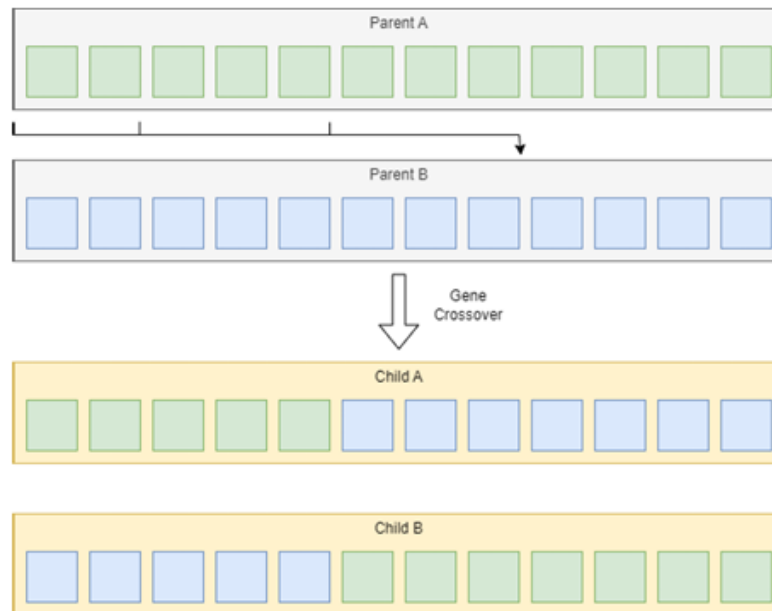
$$p_{c(t+1)} = p_{c(t)} + \frac{t}{T \cdot 10}$$

Jednačina 33 Povećanje vrednosti p_c u svakoj iteraciji

Gde su:

- t - trenutna iteracija,
- T - ukupan broj iteracija.

Ovim pristupom se u kasnijim fazama povećava verovatnoća da će geni biti uključeni u kros-over, čime se pojačava **eksploatacija** već pronađenih dobrih rešenja. Mehanizam kros-overa ilustrovan je na Slici 9.



Ilustracija 9 Cross-over mehanizam

Nakon kros-overa sledi mutacija, koja se primenjuje na svaki gen (parametar) pojedinačno. Mutacija funkcioniše tako što se vrednost svakog gena nasumično poveća ili smanji za iznos: $\frac{ub_j - lb_j}{100}$.

Smer mutacije (da li se vrednost povećava ili smanjuje) kontroliše se slučajno generisanom vrednošću u opsegu $[0,1]$:

- Ako je ta vrednost **manja od 0.5**, vrednost gena se **smanjuje**;
- Ako je **veća od 0.5**, vrednost se **povećava**.

Ovaj jednostavan, ali efikasan mutacioni mehanizam omogućava dodatno fino podešavanje rešenja i bolju pretragu lokalne okoline rešenja. Vizuelni prikaz mehanizma mutacije dat je na Slici 10.



Ilustracija 10 Mehanizam mutacije

6.1.2.5 Predloženi algoritam – SAGPSO

Kada sumiramo sve prethodno navedene procedure i modifikacije, kao deo ovog istraživanja predloženi algoritam dobija naziv **SAGPSO** (*eng. Self Adjusted Genetic Particle Swarm Optimization*), odnosno samopodesivi genetski PSO algoritam. Pseudokôd ovog algoritma prikazan je u **Algoritmu 2**.

Algoritam 2 Pseudokod SAGPSO

Korak 1: Sintetizuj početnu populaciju P koristeći QRL proceduru iz Algoritma 1

Korak 2: Odredi vrednosti Dt_0 i Dt

Korak 3: Postavi $pc = 0.1$

Korak 4: dok ($t < T$) radi:

- a. Poređaj rešenja u P prema njihovoj funkciji prilagođenosti
- b. za svako rešenje X koje pripada P radi:
 1. Primeni PSO pretragu
 2. Izračunaj DP
 3. ako ($DP < Dt$) onda:

Zameni najgore nrs jedinki sa parom novih rešenja proizvedenih kao hibridna rešenja između najslabijeg rešenja u populaciji i nasumičnog rešenja koristeći uniformni operator ukrštanja sa pc

Ili:

Zameni najgore nrs jedinki sa parom novih rešenja proizvedenih kao hibridna rešenja između najboljeg rešenja u populaciji i nasumičnog rešenja koristeći uniformni operator ukrštanja sa pc

4. Ažuriraj pc u skladu sa jednačinom

5. Ažuriraj Dt prema izrazu

c. Kraj petlje

d. Kraj petlje

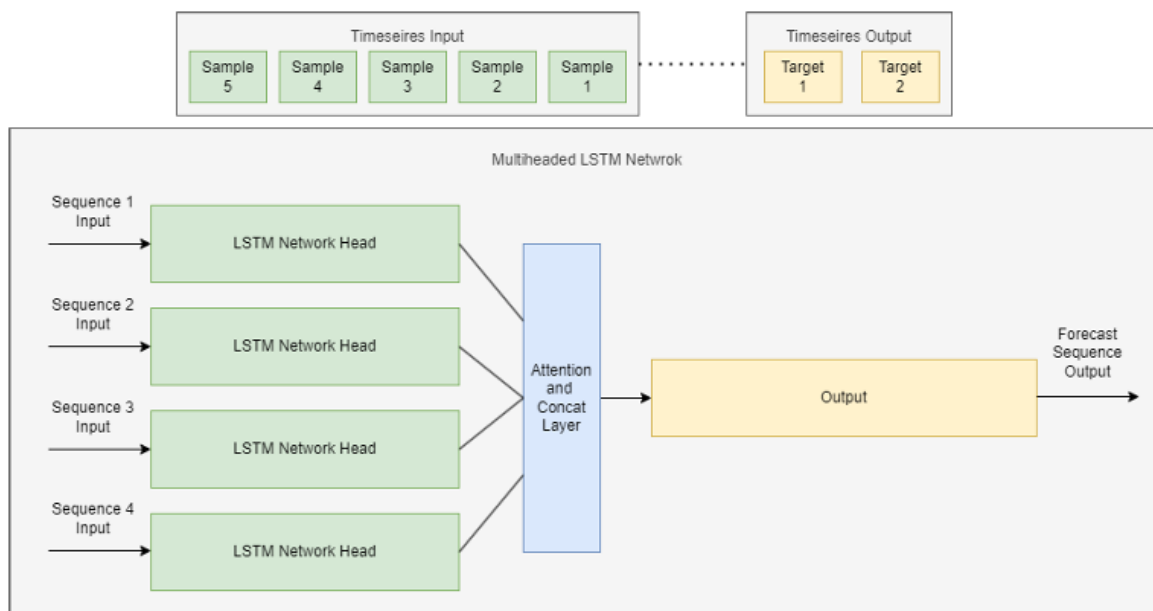
Korak 5: Vрати najboljeg pojedinca iz P

Predloženi SAGPSO je testiran na standardnom CEC 2022 skupu funkcija za testiranje, što je uobičajena praksa u računarskoj nauci za evaluaciju modifikovanih verzija metaheurističkih algoritama. Rezultati su upoređeni sa istim skupom algoritama koji su korišćeni za simulacije podešavanja LSTM modela opisane u narednom odeljku. Ono što se može zaključiti na osnovu dobijenih rezultata nakon primene ovog algoritma, je nadmašivanje svih konkurentskih optimizatora, ostvarivši najbolje prosečne rezultate na posmatranim funkcijama.

6.1.2.6 Predloženi okvir sa višeglavnom LSTM mrežom

U okviru ovog istraživanja, optimizator SAGPSO se koristi za podešavanje arhitekture višeglave LSTM (MHLSTM) mreže. Sama mreža ima više ulaza, pri čemu svaki ulaz vodi ka nezavisnoj sekvencijalnoj LSTM mreži koja obrađuje vremenske nizove. Izlazi tih pojedinačnih mreža kombinuju se pomoću mehanizma pažnje (*eng. attention*) i konkatencije. Na ovaj način mreža može da obradi dodatne informacije tokom predviđanja, ali pošto njene glave (*eng. heads*) nisu međusobno povezane pre faze pažnje, broj težinskih parametara ostaje relativno nizak, što smanjuje računarska opterećenja.

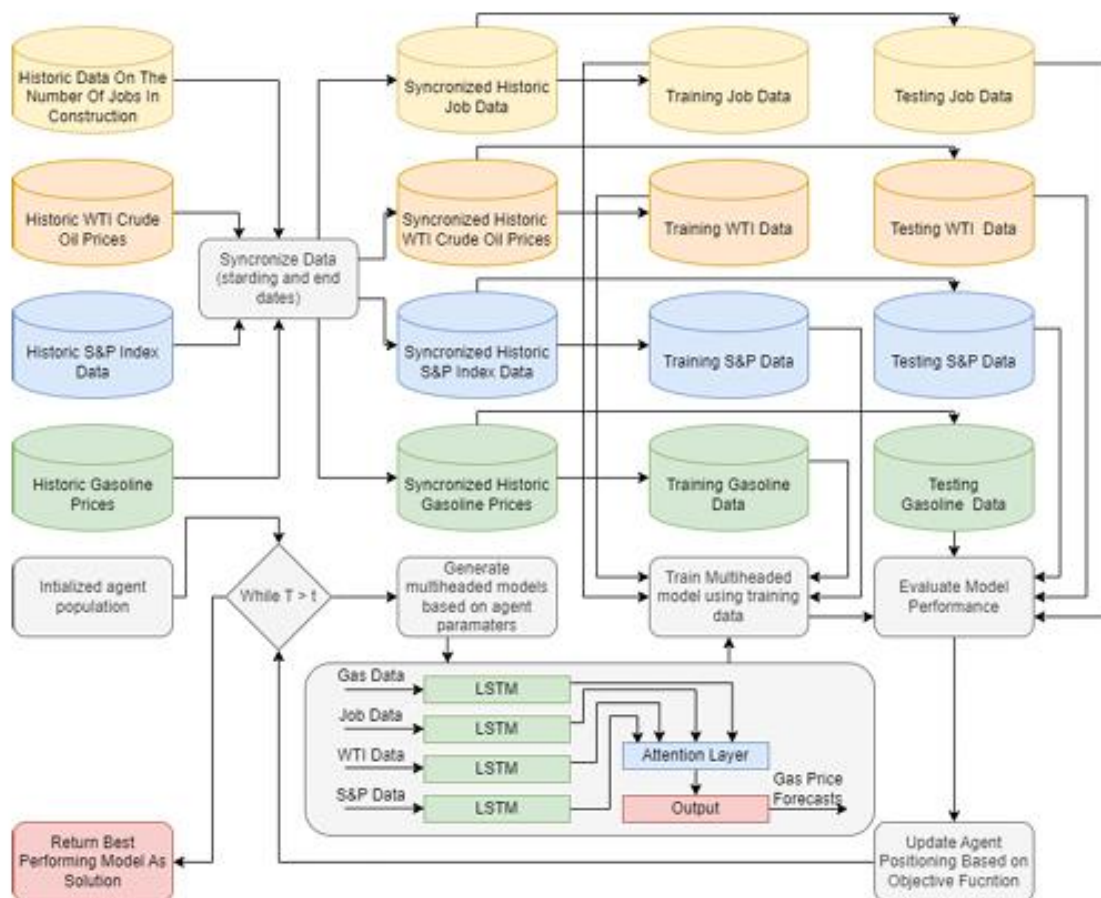
U slučajevima kada neka od mrežnih glava obrađuje složenije podatke, moguće je dodati više slojeva i neurona isključivo toj glavi, bez uticaja na ostale. Podaci koji se koriste moraju biti pravilno pripremljeni u formatu vremenskih serija, tj. organizovani u sekvence istorijskih podataka pogodnih za predikciju. Svaka glava mreže dobija svoj poseban niz podataka, pri čemu svaki niz predstavlja jedan faktor koji mreža uzima u obzir. Način kodiranja sekvenci i opšti izgled MHLSTM mreže prikazani su na slici 11.



Ilustracija 11 MHLSTM mreža

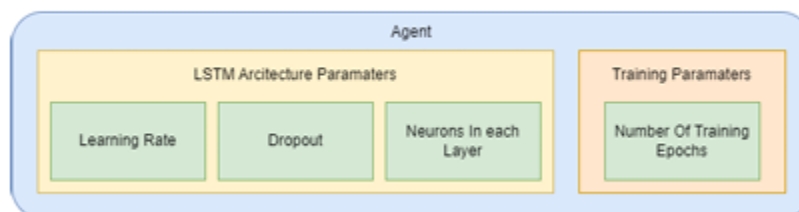
Kao i kod svakog modela, performanse u velikoj meri zavise od pravilnog izbora hiperparametara. Višeglava LSTM mreža deli mnoge hiperparametre sa klasičnom LSTM arhitekturom. Važno je napomenuti da višeglava mreža ne mora nužno koristiti isti tip za sve svoje glave — moguće je kombinovati različite mrežne arhitekture. U ovom radu, akcenat je stavljen na LSTM glave, jer su pokazale dobre rezultate u predikciji vremenskih serija.

Optimizacija hiperparametara spada u NP-teške probleme, što znači da broj mogućih kombinacija hiperparametara može eksponencijalno rasti sa brojem parametara i složenosti samog modela. Zbog toga se u ovom istraživanju koristi predloženi SAGPSO algoritam kao deo integrisanog okvira za optimizaciju MHLSTM mreže i poboljšanje tačnosti predikcija. Dijagram toka koji prikazuje predloženi okvir dat je na Slici 12.



Ilustracija 12 Predloženi okvir

Da bi se metaheuristički optimizatori koji se baziraju na populaciji mogli da primeniti na LSTM modele, hiperparametri moraju biti pravilno kodirani kao parametri agenata. Opšti šematski prikaz načina kodiranja rešenja prikazan je na Slici 13.



Ilustracija 13 Prikaz načina kodiranja parametara

Kroz vođen iterativni proces, agenti i njihova populacija se postepeno poboljšavaju. Na osnovu svakog agenta kreira se model, koji se zatim trenira korišćenjem trening skupa podataka i ocenjuje na osnovu validacionog dela skupa.

Na osnovu rezultata ciljne funkcije, parametri agenata se ažuriraju odgovarajućom strategijom pretrage. Kada se ispune kriterijumi zaustavljanja, optimizacija se završava. Najbolji agent se koristi kao osnova za izgradnju konačnog modela.

Taj model se zatim ponovo trenira na celokupnom trening skupu i evaluira korišćenjem preostalog dela podataka koji do tada nije korišćen. Tačan eksperimentalni postupak koji je korišćen za testiranje predložene metodologije biće opisan u narednom odeljku.

6.2 Eksperimentalna postavka, rezultati simulacije i perspektiva objašnjivosti

Predložena metaheuristika SAGPSO korišćena je za podešavanje hiperparametara MHLSTM modela, u svrhu predikcije cena goriva na osnovu vremenskih serija. Kao što je već istaknuto u prethodnom odeljku, zbog visokih računarskih zahteva koje ova arhitektura nameće, fokus istraživanja je stavljen na rešenja koja su dovoljno dobro optimizovana da ne iziskuju preterano velike računarske resurse – modele sa malim brojem neurona po sloju i ograničenim brojem epoha tokom obuke.

U skladu s tim, u optimizaciju su uključeni sledeći hiperparametri, sa empirijski određenim granicama za pretragu:

- stopa učenja (eng. learning rate) – realan broj u opsegu [0.0001, 0.01];
- dropout stopa – realan broj u opsegu [0.05, 0.2];
- broj epoha – ceo broj u opsegu [5, 10];
- broj neurona po sloju – ceo broj u opsegu [30, 60].

Za svaku ulaznu karakteristiku (eng. *Feature*) kreira se poseban LSTM sloj sa odgovarajućim brojem neurona. Zbog toga ukupna dužina vektora hiperparametara iznosi:

$D = 3 + \text{broj ulaznih karakteristika} + 1$.

Prva tri parametra su uvek ista (learning rate, dropout, epohe), zatim sledi niz brojeva neurona po LSTM slojevima za svaki ulaz, a poslednji parametar odnosi se na broj neurona u završnom, gusto povezanom (*eng. dense*) sloju. Model formiran na ovaj način naziva se višeglava LSTM mreža (MHLSTM).

Performanse modela podešenog pomoću SAGPSO upoređene su sa drugim savremenim metodama optimizacije, uključujući:

- osnovni PSO,
- genetski algoritam (GA),
- pretragu promenljivog susedstva (VNS),
- algoritam vatrenih muva (FA),
- veštačku koloniju pčela (ABC),
- RSA i
- COLSHADE algoritam.

Zbog zahtevnosti simulacija, svakoj metaheuristici je dodeljeno po 5 jedinki u populaciji i 8 iteracija po izvršavanju. Svaki algoritam je pokrenut u 30 nezavisnih ponavljanja. Simulacije su koristile 5 vremenskih kašnjenja (lagova) za predikciju 2 koraka unapred. Takođe je bila aktivna strategija ranog zaustavljanja (*eng. Early stopping*), koja se primenjuje ako nema poboljšanja na validacionom skupu u okviru broj epoha / 3.

Ciljna funkcija za minimizaciju bila je srednja kvadratna greška (*eng. Mean square error, MSE*), dok su dodatni pokazatelji performansi navedeni u narednom odeljku. Eksperimenti su sprovedeni na računarskoj konfiguraciji sa procesorom Intel i9 13900k, 128 GB RAM memorije i grafičkom karticom NVIDIA GeForce RTX 3090 (24 GB VRAM-a).

6.2.1 Primenjeni kriterijumi za ocenjivanje

Rezultati simulacije predloženog modela su ocenjeni korišćenjem konvencionalnog skupa od četiri metrička pokazatelja, koja se često koriste za procenu metoda mašinskog učenja:

- srednja kvadratna greška korena (RMSE),
- srednja apsolutna greška (MAE),
- srednja kvadratna greška (MSE),
- koeficijent determinacije (R^2).

Ovi metrički pokazatelji se mogu izračunati prema formulama (38), (39), (40) i (41), redom.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

Jednačina 34 Srednja kvadratna greška korena

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Jednačina 35 Srednja apsolutna greška

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Jednačina 36 Srednja kvadratna greška

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

Jednačina 37 Koeficijent determinacije

Gde su:

- y_i i $\hat{y}_i$ – stvarne i predviđene vrednosti za i-to posmatranje
- $\bar{y}$ – aritmetička sredina stvarnih podataka
- n – veličina uzoraka

Dodatno, korišćen je još jedan metrički pokazatelj za bolje ocenjivanje sintetisanih modela — indeks slaganja (IoA), koji je definisan jednačinom 38. Ovaj indeks meri stepen usklađenosti između predviđenih vrednosti modela i stvarnih rezultata ili ciljeva.

$$IoA = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (|y_i - \bar{y}| + |\hat{y}_i - \bar{y}|)^2} = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{2 \sum_{i=1}^n (|y_i - \bar{y}| + |\hat{y}_i - \bar{y}|)}$$

Jednačina 38 Indeks slaganja

Gde su:

- y_i i $\hat{y}_i$ – stvarne i predviđene vrednosti za i-to posmatranje
- $\bar{y}$ – aritmetička sredina stvarnih podataka

Visoka vrednost ovog indeksa ukazuje na dobru performansu modela, odnosno na snažno slaganje između predviđanja i stvarnih rezultata.

6.2.3 Skup podataka korišćen u eksperimentima

Za potrebe eksperimentisanja, sastavljen je skup podataka formiran iz više javno dostupnih izvora. Opšti indikatori ekonomske situacije, kao što su S&P indeks i broj zaposlenih u građevinarstvu su takođe preuzeti iz javnih izvora, kao što su Yahoo Finance sajt ¹ i zvanični sajt američke federalne vlade².

Dalji podaci o cenama sirove nafte po barelu, koje obezbeđuje West Texas Intermediate (WTI), prikupljeni su sa sajta Energy Information Agency (EIA)³, zajedno sa cenama benzina za krajnje potrošače u SAD, takođe sa EIA sajta⁴. Prikupljanje podataka je sprovedeno početkom maja 2024. godine. Skup podataka obuhvata informacije od 1. januara 2000. do 1. maja 2024. sa nedeljnom rezolucijom tačaka.

Tokom treniranja modela primenjena je normalizacija koristeći Min-Max skaliranje iz *scikit-learn* biblioteke. Ova metoda pomaže da se sačuva proporcionalni uticaj svake

¹ <https://finance.yahoo.com/quote/%5EGSPC/>

² [All Employees, Residential Building Construction \(CES2023610001\) | FRED | St. Louis Fed](https://fred.stlouisfed.org/series/CES2023610001)

³ <https://www.eia.gov/dnav/pet/hist/rwtcW.htm>

⁴ https://www.eia.gov/dnav/pet/hist/LeafHandler.ashx?n=PET&s=EMM_EPMR_PTE_NUS_DPG&f=W

karakteristike na rezultate regresije i smanjuje potencijalna pristrasnost koju mogu izazvati karakteristike sa višim vrednostima.

U eksperimentima nije vršeno dodatno biranje karakteristika (eng. feature selection); karakteristike su empirijski odabrane kao uobičajeni pokazatelji opšte ekonomske situacije. Podaci su strukturirani u vremenske serije koristeći *TensorFlow Time Series Generator* sa ukupno pet vremenskih pomaka (eng. lags).

U radu je predstavljen višeglavi model, gde je svaki „glava“ potpuno zasebna i nezavisna neuronska mreža dizajnirana da obrađuje pojedinačne karakteristike. Ova arhitektura smanjuje računarske zahteve između neurona i smanjuje ukupne troškove računanja tokom treniranja i implementacije modela. Mogu se dodavati dodatne „glave“ za upravljanje novim karakteristikama ili ukloniti ako se određene karakteristike pokažu kao nebitne tokom testiranja.

Modelima je zadato da prave prognoze dva koraka unapred. Formatirani podaci su podeljeni na trening, validaciju i test skupove, gde trening obuhvata 70% svih podataka, validacija 10%, a testiranje preostalih 20%. Vizualizacija skupa podataka data je na slici 14, gde su podaci za trening označeni plavom bojom, testiranje zelenom, a validacija narandžastom.



Ilustracija 14 Vizuelni prikaz skupa podataka

6.2.4 Konačni rezultati simulacije

Ovaj pododeljak prikazuje i razmatra rezultate eksperimentalnih simulacija. U tabelama su najbolji postignuti rezultati istaknuti podebljanim fontom. Dobijeni ishodi potom se

podvrgavaju rigoroznoj statističkoj validaciji, a najbolji modeli se tumače korišćenjem tehnika objašnjive veštačke inteligencije (*eng. Explainable AI*). Tako na narednoj slici možemo videti dobijene rezultate funkcije cilja.

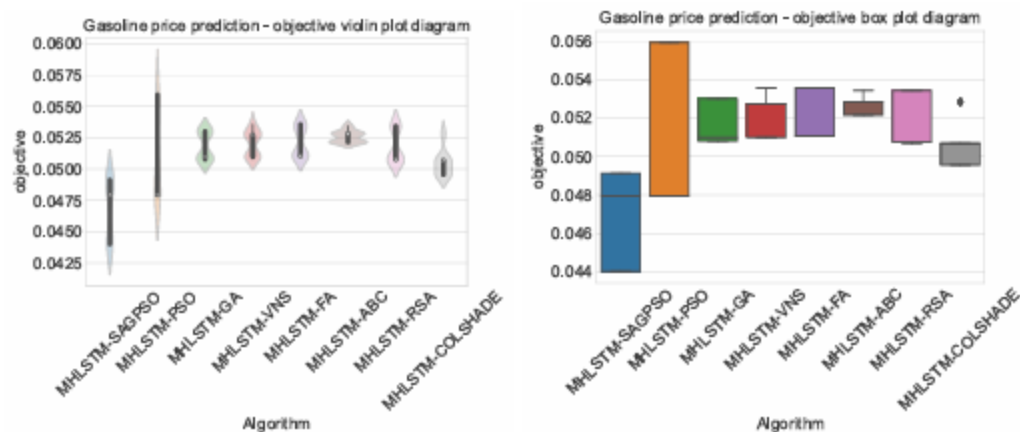
Method	Best	Worst	Mean	Median	Std	Var
MHLSTM-SAGPSO	0.044025	0.049128	0.046888	0.047946	0.002459	6.05E-06
MHLSTM-PSO	0.047946	0.055940	0.050867	0.047946	0.003673	1.35E-05
MHLSTM-GA	0.050794	0.053012	0.051861	0.050988	0.001103	1.22E-06
MHLSTM-VNS	0.050988	0.053568	0.052062	0.052733	0.001002	1.00E-06
MHLSTM-FA	0.051068	0.053568	0.052160	0.051068	0.001195	1.43E-06
MHLSTM-ABC	0.052159	0.053435	0.052591	0.052845	0.000452	2.04E-07
MHLSTM-RSA	0.050687	0.053435	0.051783	0.050777	0.001320	1.74E-06
MHLSTM-COLSHADE	0.049565	0.052845	0.050446	0.050687	0.001038	1.08E-06

Ilustracija 15 Rezultati funkcije cilja za svaki optimizator

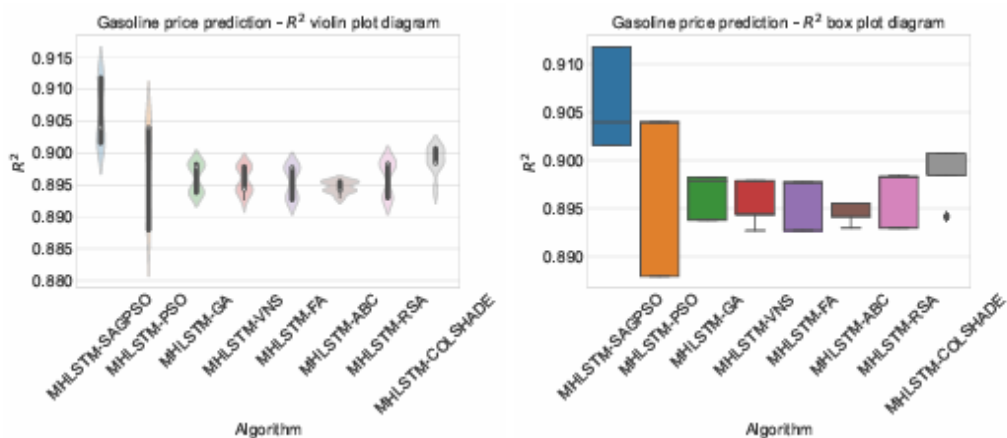
Na osnovu ovoga možemo zaključiti da predloženi optimizator pokazuje jasnu prednost u odnosu na ostale algoritme koji su uključeni u komparativnu analizu, ostvarivši najbolji rezultat od **0.044025**, najlošiji **0.049128**, kao i najbolju prosečnu vrednost od **0.046888**. Medijana rezultata je ista za predloženi modifikovani algoritam i za originalni PSO algoritam. Ovi rezultati ukazuju da su uvedene izmene usmerile agente ka perspektivnijim oblastima u pretraživačkom prostoru.

Stabilnost algoritama je procenjivana na osnovu standardne devijacije i varijanse. Najveću stabilnost pokazao je ABC algoritam, ali on nije uspeo da ostvari najbolje rezultate.

Dalja poređenja stabilnosti prikazana su na Ilustraciji 16 (za funkciju cilja) i Ilustraciji 17 (za evaluacionu metriku R^2). Modifikovani algoritam pokazuje poboljšanja u stabilnosti u poređenju sa originalnim PSO algoritmom. Iako i drugi algoritmi pokazuju visok stepen stabilnosti, njihova najbolja rešenja su lošija čak i od najlošijih rezultata koje je postigao modifikovani optimizator. Ove razlike su uočene i kod funkcije cilja i kod evaluacionih metrika.



Ilustracija 16 Distribucija rezultata funkcije cilja



Ilustracija 17 Distribucija rezultata indikatorske funkcije

Detaljne metrike za najbolje modele prikazane su na Slici 18. Na osnovu svih prikupljenih metrika, predloženi **SAGPSO** algoritam uspešno je odabrao parametre za **MHLSTM modele**, koji daju najbolje rezultate u poređenju sa ostalim optimizatorima. Konkretno, postignuti su:

- **R^2 skor: 0.911797**
- **MAE: 0.152277**

- **RMSE: 0.044025**
- **IoA: 0.976489**

Optimizovani model nadmašuje model podešen originalnim PSO algoritmom, kao i ostale savremene algoritme uključene u analizu, što ukazuje da predložene izmene doprinose boljem učinku na ovom problemu.

Method	R ²	MAE	MSE	RMSE	IoA
MHLSTM-SAGPSO	0.911797	0.152277	0.044025	0.209810	0.976489
MHLSTM-PSO	0.903941	0.161130	0.047946	0.218725	0.972500
MHLSTM-GA	0.898235	0.163501	0.050794	0.225373	0.973852
MHLSTM-VNS	0.897846	0.158607	0.050988	0.225784	0.972690
MHLSTM-FA	0.897685	0.168871	0.051068	0.225983	0.970815
MHLSTM-ABC	0.895501	0.163904	0.052159	0.228362	0.971953
MHLSTM-RSA	0.898269	0.163400	0.050777	0.225337	0.972396
MHLSTM-COLSHADE	0.900697	0.164618	0.049565	0.222632	0.973186

Ilustracija 18 Detaljni prikaz metrike modela sa najboljim performansama

Performanse su takođe analizirane po koraku prognoze, a rezultati su prikazani na Slici 19.

Method	metric	one-step-ahead	two-steps-ahead
MHLSTM-SAGPSO	R ²	0.910039	0.913555
	MAE	0.153983	0.150571
	MSE	0.044902	0.043147
	RMSE	0.211902	0.207719
	IoA	0.975813	0.977165
MHLSTM-PSO	R ²	0.912932	0.894950
	MAE	0.156251	0.166009
	MSE	0.043458	0.052434
	RMSE	0.208466	0.228984
	IoA	0.974622	0.970377
MHLSTM-GA	R ²	0.899183	0.897287
	MAE	0.162865	0.164137
	MSE	0.050321	0.051267
	RMSE	0.224324	0.226423
	IoA	0.973861	0.973843
MHLSTM-VNS	R ²	0.900672	0.895020
	MAE	0.156709	0.160506
	MSE	0.049578	0.052399
	RMSE	0.222661	0.228907
	IoA	0.973357	0.972023
MHLSTM-FA	R ²	0.897665	0.897706
	MAE	0.169874	0.167867
	MSE	0.051079	0.051058
	RMSE	0.226006	0.225960
	IoA	0.970908	0.970721
MHLSTM-ABC	R ²	0.892693	0.898309
	MAE	0.167203	0.160605
	MSE	0.053560	0.050757
	RMSE	0.231431	0.225294
	IoA	0.971298	0.972608
MHLSTM-RSA	R ²	0.898645	0.897893
	MAE	0.162771	0.164029
	MSE	0.050589	0.050965
	RMSE	0.224921	0.225754
	IoA	0.972652	0.972139
MHLSTM-COLSHADE	R ²	0.900594	0.900801
	MAE	0.164567	0.164669
	MSE	0.049617	0.049513
	RMSE	0.222748	0.222516
	IoA	0.973234	0.973138

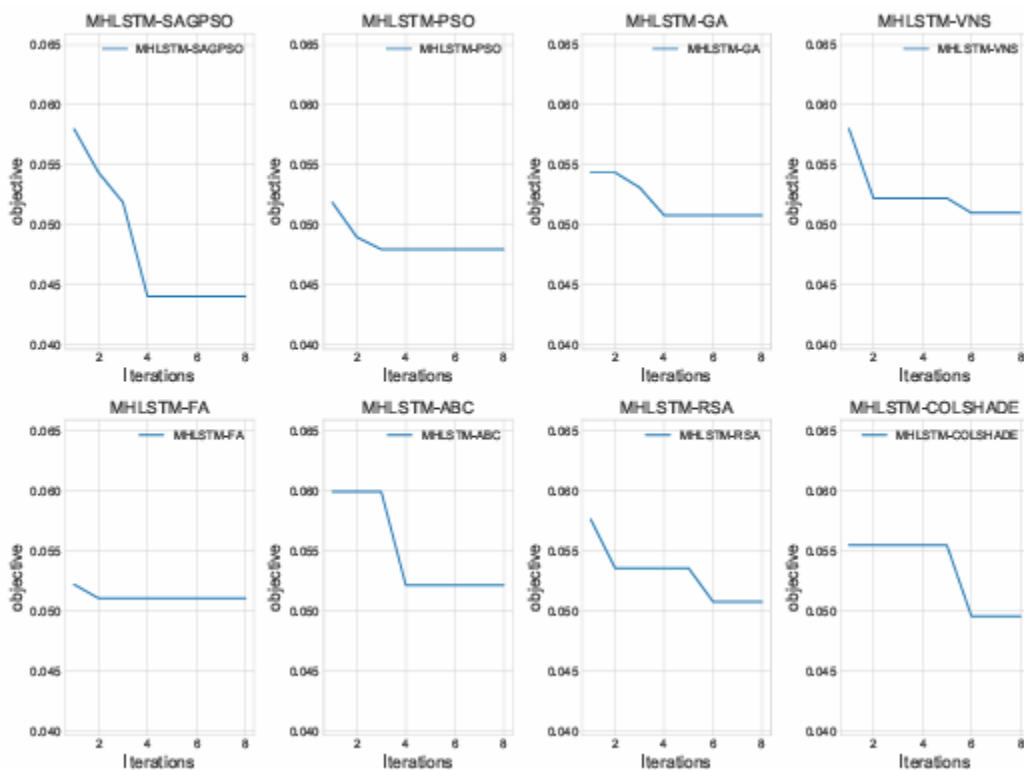
Ilustracija 19 Detaljni prikaz metrike modela po koraku prognoze

Dok model za predikciju optimizovan predloženim algoritmom daje bolje rezultate za dugoročnije prognoze, model podešen pomoću PSO algoritma pokazuje prednost za jedan korak unapred, prema metrikama R², MSE i RMSE. Ovo je donekle očekivano i dodatno potvrđuje teoremu o nepostojanju univerzalnog najboljeg algoritma (*eng. No Free Lunch theorem*), koja tvrdi da nijedan pristup nije jednako efikasan za sve probleme i pod svim uslovima testiranja.

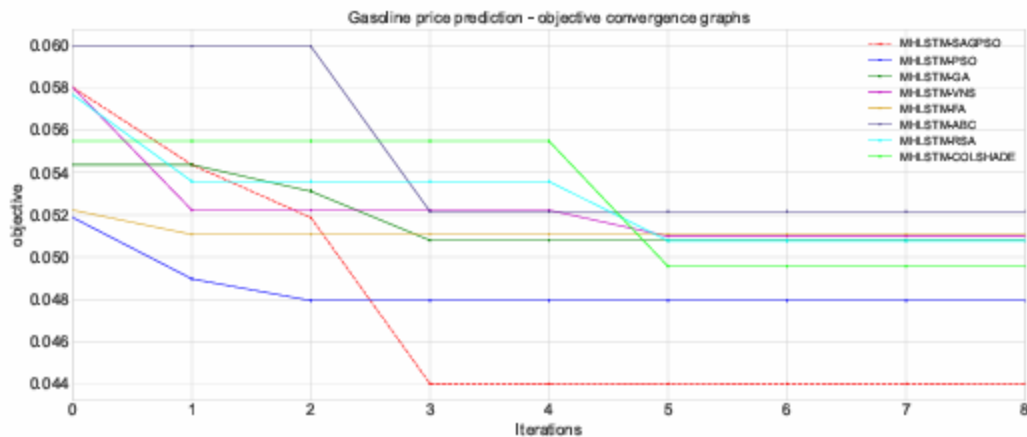
Za potpuno razumevanje snaga i slabosti nekog algoritma potrebne su obimne simulacije i validacija. Dodatne statističke i explainable AI tehnike primenjene su i predstavljene u nastavku rada.

Kada govorimo o performansama metaheurističkih optimizatora, mnogo informacija može se dobiti posmatranjem stope konvergencije. Prednosti i mane u pogledu eksploracije (istraživanja) i eksploatacije (iskorišćavanja), kao i njihov balans, mogu se uočiti iz priloženih dijagrama konvergencije. Stope konvergencije za svaki optimizator pojedinačno, kao i uporedni

dijagram koji prikazuje razlike među njima, dati su na ilustracijama 20 i 21 (za konvergenciju funkcije cilja). Kao što se može videti, mnogi algoritmi zapadaju u lokalne ekstreme, što ukazuje na nedovoljnu eksploraciju kod ovog problema – često dostignu lokalni optimum već nakon 2 do 4 iteracije i nakon toga ne napreduju dalje.

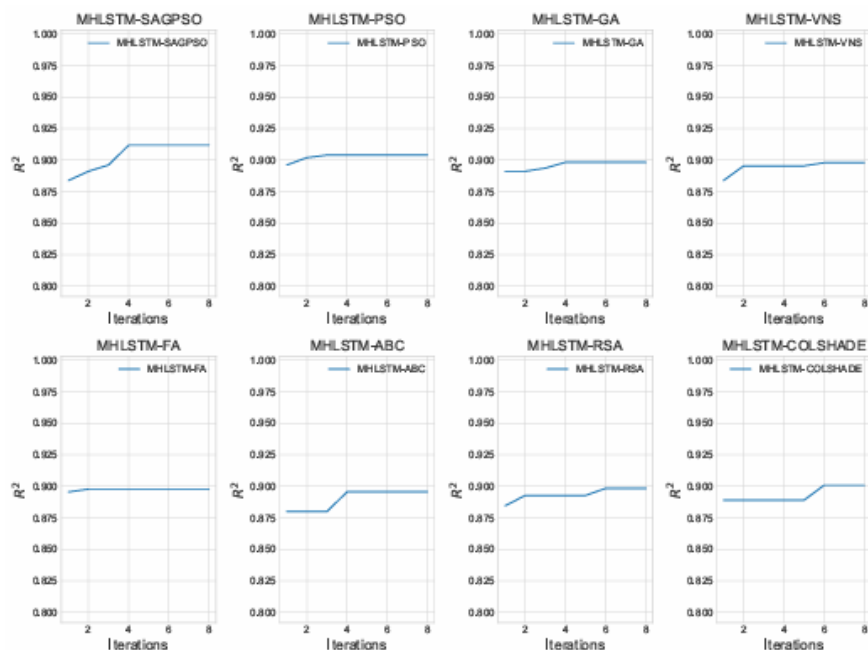


Ilustracija 20 Prikaz stopa konvergencije rezultata funkcije cilja za svaki optimizator

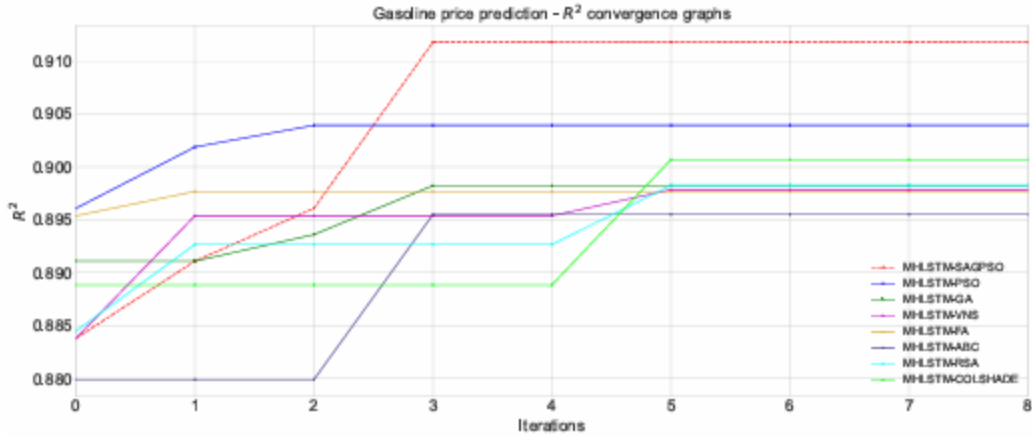


Ilustracija 21 Usporedna analiza stopa konvergencije rezultata funkcije cilja za svaki optimizator

Mehanizmi uvedeni u PSO optimizator pomažu u prevazilaženju ovih lokalnih optimuma, i omogućavaju otkrivanje boljeg rešenja koje nadmašuje ostale algoritme u poređenju. Slični rezultati zabeleženi su i kod evaluacionih metrika (indikatorskih funkcija), sa pojedinačnim dijagramima prikazanim na Ilustraciji 22, dok je objedinjeni dijagram prikazan na Slici 23.

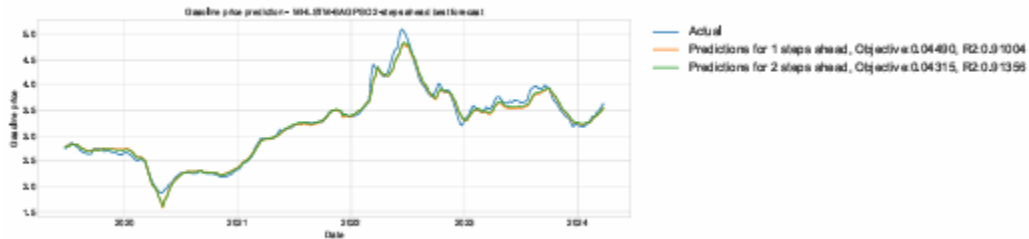


Ilustracija 22 Prikaz stopa konvergencije rezultata indikatorske funkcije za svaki optimizator



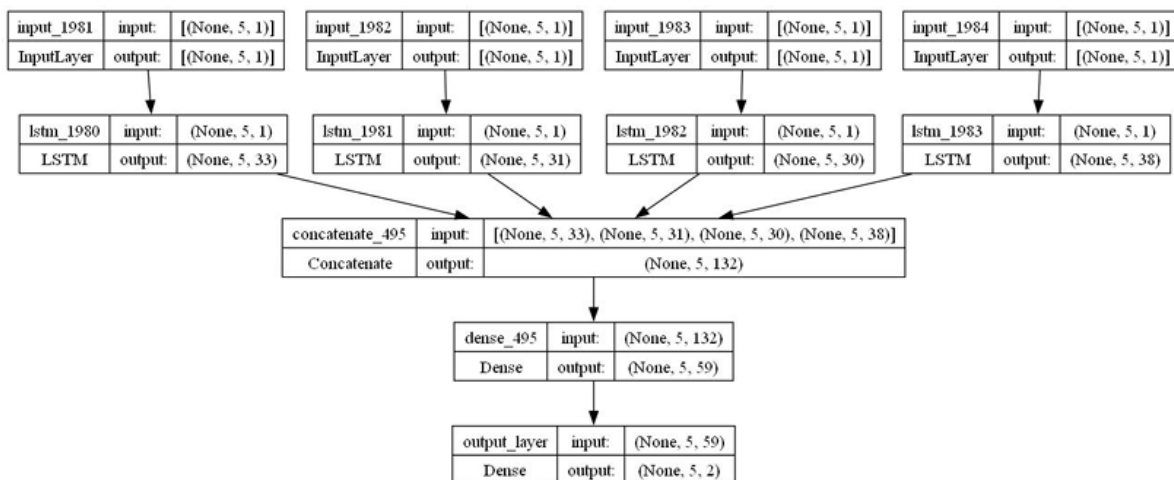
Ilustracija 23 Uporedna analiza stopa konvergencije rezultata indikatorske funkcije za svaki optimizator

Prognoze cena dobijene korišćenjem modela sa najboljim performansama prikazane su na Ilustraciji 24 za jedan i dva koraka unapred, upoređene sa stvarnim (ground truth) vrednostima.



Ilustracija 24 Modeli sa najboljim predikcijama

Arhitektura MHLSTM modela koji je optimizovan predloženim SAGPSO algoritmom i koji je postigao najbolje rezultate prikazana je na Ilustraciji 25.



Ilustracija 25 Prikaz arhitekture predloženog modela sa ostvarenim najboljim rezultatima

Izbori parametara za ovaj model, kao i za najbolje modele koje su optimizovali drugi algoritmi uključeni u analizu, dati su na Ilustraciji 26 radi omogućavanja ponovljivosti eksperimenta.

Method	learning rate	dropout	epochs	L1	L2	L3	L4	L5
MHLSTM-SAGPSO	0.000210	0.173593	8	33	31	30	38	59
MHLSTM-PSO	0.000100	0.050000	7	59	60	56	60	60
MHLSTM-GA	0.008279	0.114248	9	51	45	42	41	32
MHLSTM-VNS	0.000195	0.151844	5	59	30	46	35	60
MHLSTM-FA	0.006478	0.170342	5	57	54	60	49	39
MHLSTM-ABC	0.003905	0.104181	8	54	41	38	31	56
MHLSTM-RSA	0.001562	0.094014	5	42	30	60	30	45
MHLSTM-COLSHADE	0.009799	0.125558	5	46	47	56	60	43

Ilustracija 26 Izabrani parametri za predloženi model

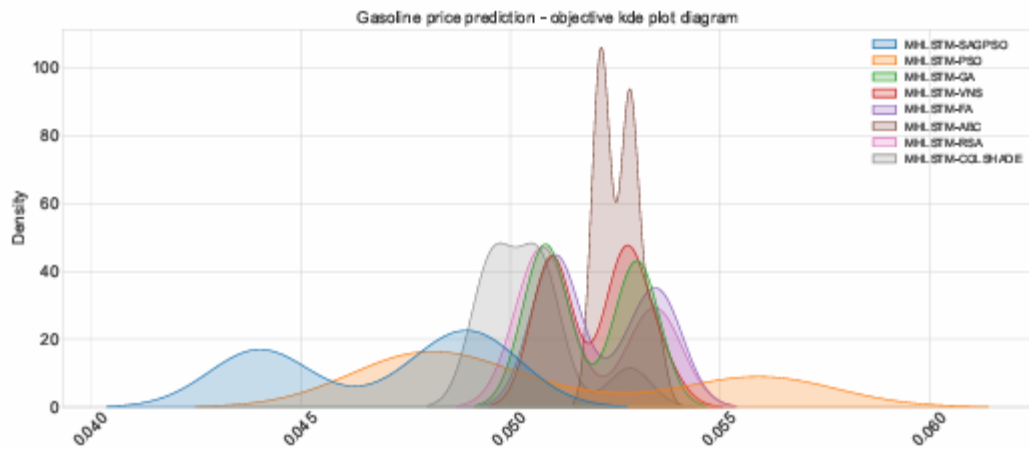
6.2.5 Statistička validacija

Dobijeni rezultati simulacija su evaluirani na takav način da se omogući efikasno dobijanje statističke značajnosti. U cilju ovoga, prikupljeni su najbolji rezultati 30 nezavisnih egzekucija svakog predloženog metaheurističkog algoritma i posmatrani su kao serija podataka.

Ono što je bitno napomenuti jeste to da je pre izvršenja statističkih testova potrebno izabrati najpogodniji tip testa, spram rezultata koje hoćemo da validiramo. Statistički testovi se mogu podeliti na dve kategorije, parametarski i neparametarski. Primena parametarskih testova zahteva prethodnu verifikaciju da li su zadovoljene sve pretpostavke o normalnosti distribucija samih podataka, homogenosti varijanse i nezavisnosti uzoraka. Sa druge strane, ako jedna ili sve pretpostavke nisu zadovoljne, onda se pristupa primeni neparametarskog tipa testa. Ova vrsta ne zavisi nužno od nekih specifičnih distribucija podataka i samim time pružaju robusniji način za poređenje performansi različitih metaheurističkih algoritama. Na ovaj način je omogućena ozbiljnija i tačnija interpretacija rezultata, i potvrđuje da razlike koje se javljaju u performansama ovih algoritama nose određenu statističku važnost.

Uzimajući prethodno navedeno u obzir, potrebno je dakle utvrditi određene kriterijume nezavisnosti, normalnosti i homoskedastičnosti. Pošto se svako izvršavanje metaheurističkog algoritma pokreće sintezom populacije pseudo-nasumičnih rešenja, kriterijum nezavisnosti je zadovoljen. Sa druge strane, kriterijum homoskedastičnosti (konstantna varijansa) je procenjen primenom **Levene-ovog testa** (G. V. Glass et al., 1966), koji je dao p-vrednost od 0.68 za svaki slučaj, što znači da je i ovaj uslov ispunjen.

Na kraju, za ispitivanje normalnosti distribucije, korišćen je **Shapiro-Wilk** test (S. S. Shapiro et al., 1972), koji je analiziran zasebno za svaki algoritam. Pošto su sve dobijene p-vrednosti bile ispod 0.05, nulta hipoteza (H_0) iz prve sekcije ove disertacije se može odbaciti. To znači da rezultati nijednog od analiziranih metaheurističkih algoritama ne potiču iz normalne distribucije. Ova pojava je dodatno potvrđena i KDE dijagramima prikazanim na Ilustraciji 27.



Ilustracija 27 Rezultati funkcije cilja kroz KDE dijagrame za svaki optimizator

Detaljni rezultati Shapiro-Wilk testa nalaze se na Ilustraciji 28.

Approach	SAGPSO	PSO	GA	VNS	FA	ABC	RSA	COLSHADE
MHLSTM	0.026	0.028	0.033	0.031	0.034	0.032	0.029	0.041

Ilustracija 28 Rezultati Shapiro-Wilk testa

Dakle, kako uslov normalnosti nije ispunjen, nije preporučljivo koristiti parametarske statističke testove za dalju analizu.

Pošto uslovi za primenu parametarskih testova nisu ispunjeni, odlučeno je da se nastavi sa neparametarskim statističkim testovima. Postoji više opcija za neparametarsku analizu, kao što su Friedman-ov test (N. E. Zamri et al.) i Wilcoxon-ov test, međutim, prema izvoru (A. LaTorre et al., 2021), Wilcoxon-ov test je znatno prikladniji za evaluaciju metaheurističkih algoritama.

Zbog toga je primenjen Wilcoxon-ov test sa rangiranim parovima (Wilcoxon signed-rank test) kako bi se uporedile performanse predloženog metoda sa ostalim posmatranim pristupima u odnosu na rezultate funkcije cilja (MSE). Za testiranje su korišćeni isti nizovi podataka kao i kod prethodnog testa (Shapiro-Wilk test za proveru normalnosti).

Rezultati Wilcoxon testa dati su na Ilustraciji 29.

SAGPSO vs. others	PSO	GA	VNS	FA	ABC	RSA	COLSHADE
MHLSTM	0.037	0.029	0.023	0.021	0.019	0.031	0.041

Ilustracija 29 Rezultati Wilcoxon testa

Pošto su u svakom slučaju dobijene p-vrednosti manje od granice značajnosti $\alpha = 0.05$, to znači da su rezultati koje je postigao predloženi SAGPSO algoritam statistički značajno bolji od ostalih metoda obuhvaćenih komparativnom analizom.

6.2.6 Analiza osetljivosti predloženog SAGPSO algoritma i poređenje sa sličnim pristupima

Radi dodatne evaluacije performansi predloženog SAGPSO algoritma, sprovedena je i analiza osetljivosti kako bi se utvrdio uticaj svake pojedinačne modifikacije. Pored osnovnog PSO algoritma i predloženog SAGPSO, simulacije su izvršene i za svaku pojedinačnu modifikaciju posebno, kao i za kombinacije po dve modifikacije.

Slično kao i u glavnim eksperimentima, sprovedeno je 30 pojedinačnih izvršavanja, a praćena je prosečna vrednost R^2 metrike.

Korišćene oznake za pojedinačne komponente su sledeće:

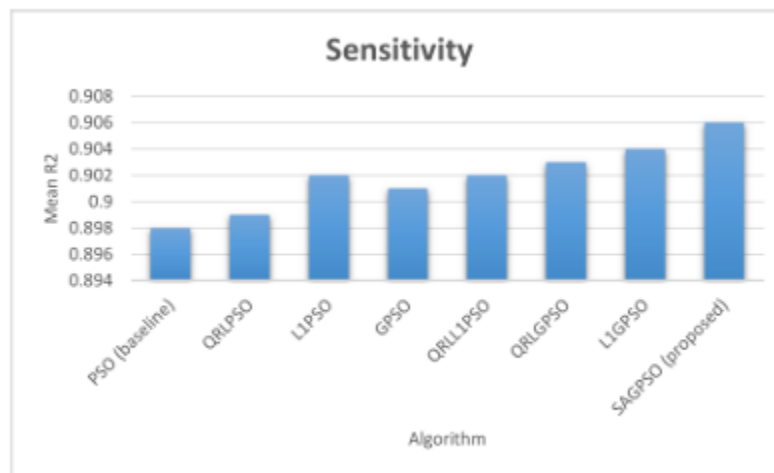
- **QRL** za proceduru inicijalizacije (*eng. QRL initialization*),
- **L1** za mehanizam očuvanja raznovrsnosti populacije (*eng. Population diversity*),
- **G** za mehanizme ukrštanja i mutacije (genetski operatori).

Na osnovu toga, korišćeni su sledeći akronimi:

- **QRLPSO** – verzija algoritma u kojoj je primenjena samo QRL inicijalizacija.
- **L1PSO** – varijanta sa primenjenom isključivo strategijom raznovrsnosti populacije.
- **GPSO** – verzija koja uključuje samo genetske operatore (ukrštanje i mutaciju).
- **QRLL1PSO** – varijanta sa QRL inicijalizacijom i raznovrsnošću populacije.

- **QRLGPSO** – verzija sa QRL inicijalizacijom i genetskim operatorima.
- **L1GPSO** se odnosi na varijantu algoritma u kojoj su implementirani **i mehanizam raznovrsnosti populacije i genetski operatori**.

Rezultati ove analize prikazani su na Ilustraciji 30.



Ilustracija 30 Rezultati analize senzitivnosti PSO algoritma sa modifikacijama

Vredno je napomenuti da svaka pojedinačna modifikacija poboljšava rezultate u odnosu na osnovni PSO algoritam, pri čemu se mehanizam raznovrsnosti populacije (L1) pokazao kao najuticajnija pojedinačna izmena.

Pored toga, izvršeno je i poređenje osetljivosti predloženog SAGPSO algoritma, osnovnog PSO i GA algoritma sa rezultatima sličnih algoritama koji su nedavno predstavljeni. U analiziranu grupu algoritama uvršteni su:

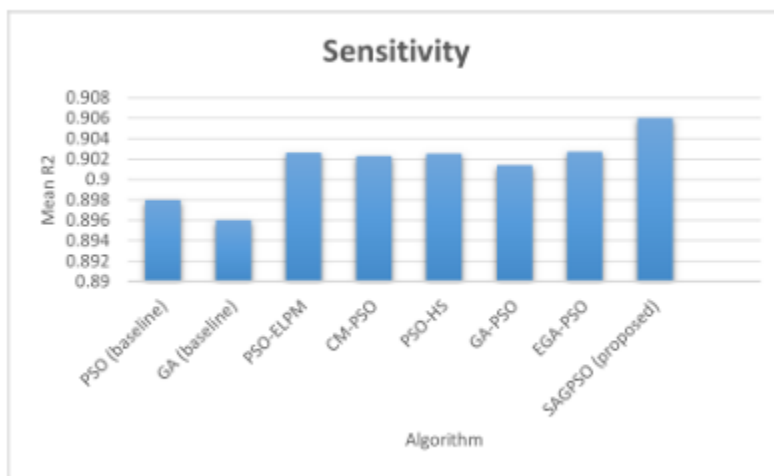
- PSO sa elitnim učenjem, poboljšanim ažuriranjem parametara i eksponencijalnim operatorom mutacije (PSO-ELPM) (H. Moazen et al., 2023)
- PSO sa mehanizmom takmičenja (CM-PSO) (Z. Xing et al., 2022.),
- PSO baziran na algoritmu harmonijske pretrage (PSO-HS) (S.-P. Zhu et al., 2022.),
- Hibridni GA-PSO algoritam (Y. Liu et al., 2020.),
- Poboljšani hibridni GA-PSO algoritam (EGA-PSO) (A. K. Balyan et al., 2022.)

Ove algoritme su nezavisno implementirali autori i korišćeni su u identičnom eksperimentalnom okruženju:

- 5 rešenja u populaciji,
- 8 iteracija po izvršavanju,
- 30 nezavisnih izvršavanja

Praćena je prosečna vrednost R^2 metričkog pokazatelja, a rezultati su prikazani na Ilustraciji 31.

Na osnovu rezultata može se zaključiti da predloženi SAGPSO algoritam nadmašuje sve navedene savremene pristupe u pogledu prosečne R^2 vrednosti.



Ilustracija 31 Rezultati analize senzitivnosti PSO algoritma u odnosu na slične algoritme

6.2.7 Tumačenje uticaja karakteristika najboljeg modela korišćenjem SHAP Analize

Pri izgradnji i evaluaciji modela, često je jednako važno razumeti koji faktori najviše utiču na odluke koje model donosi. Razumevanje uticaja pojedinačnih karakteristika može poboljšati način prikupljanja i pripreme podataka, a samim tim i tačnost budućih predikcija.

Kod jednostavnijih modela, kao što su oni zasnovani na odlučujućim stablima, određivanje važnosti karakteristika je prilično jednostavno. Međutim, kompleksnost neuronskih mreža može dovesti do toga da se značaj pojedinih ulaznih karakteristika umanjuje ili prenaglasti

na određenim mestima unutar mreže. Zbog toga su složenije arhitekture teže za interpretaciju, pa se koriste približne metode kao što su SHAP (*eng. Shapley Additive Explanations*) [85], koje omogućavaju razumevanje doprinosa svake karakteristike.

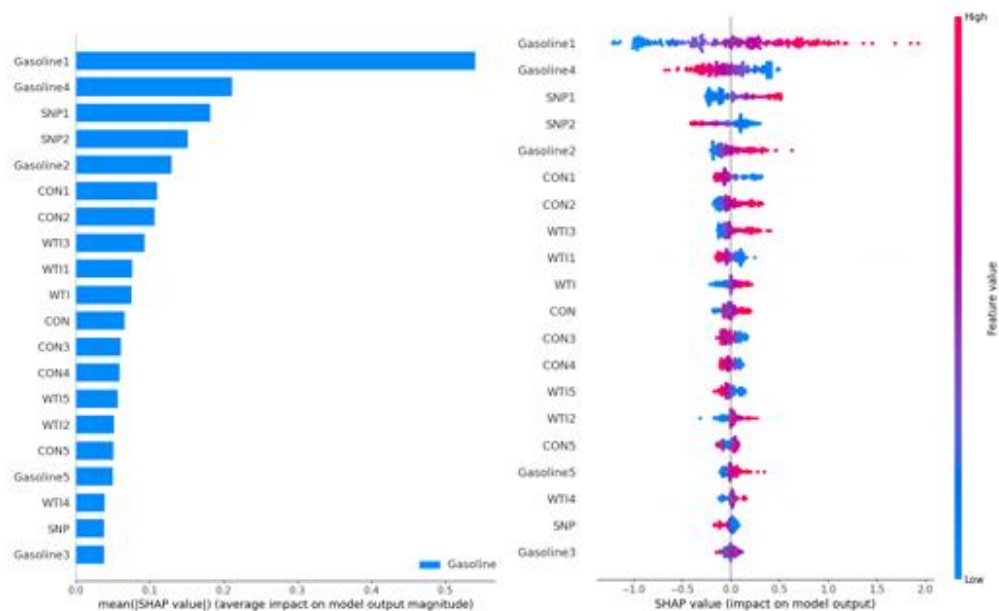
Korišćenjem serije pojednostavljenih linearnih modela za prognozu i odgovarajuće kernel SHAP metode, moguće je proceniti značaj karakteristika i kod složenih modela. Najbolji model za prognozu, optimizovan korišćenjem predloženog modifikovanog algoritma, analiziran je pomoću SHAP metode, a globalna važnost karakteristika prikazana je na Slici 32.

Nazivi karakteristika sa brojevima označavaju vremensku udaljenost između uzoraka – npr. „1“ predstavlja današnju cenu, dok „5“ predstavlja cenu od pre 5 nedelja.

Model najviše važnosti pridaje:

- trenutnoj ceni benzina,
- ceni benzina od pre 4 nedelje,
- zatim današnjem i prošlonedeljnom S&P indeksu,
- i ceni benzina iz prethodne nedelje

Takođe, važnim faktorima model smatra i broj zaposlenih u građevinarstvu za tekuću i prethodnu nedelju. Zanimljivo je da se građevinski sektor procenjuje kao stabilniji faktor, jer se te promene odvijaju mesečno, za razliku od drugih faktora koji variraju iz nedelje u nedelju ili čak dnevno.



Ilustracija 32 Grafički prikaz značaja karakteristika najboljeg modela korišćenjem SHAP metode

7 Zaključak

Rastuća globalna potražnja za energentima i sve manji fosilni resursi na globalnom nivou podstakli su novo interesovanje za oblast prognoze potrošnje i cena energije. Ova disertacija je primarno istraživala primenu modela Veštačke inteligencije u rešavanju prethodno pomenutog problema, sa posebnim fokusom na predikciju vremenskih serija. Glavni cilj je unaprediti tačnosti predikcija troškova i potrošnje energenata, u ovom konkretnom slučaju cena goriva, što ima značajan uticaj na različite sektore.

Uzimajući u obzir složenost faktora koji utiču na globalne cene—uključujući društveno-ekonomske okolnosti, logistiku distribucije, dostupnost resursa i međunarodne politike—istraživanje se fokusiralo na primenu lako prilagodljivih MHLSTM modela za predikciju cena goriva. S obzirom na veliku osetljivost performansi algoritama na izbor hiperparametara, uveden je modifikovani algoritam čestica roja (*eng. Particle Swarm Optimizer, PSO*), sa ciljem optimizacije arhitekture i parametara obuke.

Kroz komparativnu analizu sa više savremenih optimizatora, uz korišćenje podataka iz javno dostupnih izvora, postignuti su sledeći ključni rezultati:

- **Srednja kvadratna greška (MSE): 0.044025,**
- **R² koeficijent: 0.911797**

Ovi rezultati jasno pokazuju veliki potencijal za praktičnu primenu predloženog modela u cilju rešavanja prethodno opisanog problema. Rezultati dobijeni nakon završetka simulacija su zatim statistički validirani kako bi se osigurala njihova pouzdanost, na osnovu čega je dodatno utvrđeno da predloženi algoritam pruža značajna poboljšanja.

Na kraju, primenjene su i metode objašnjive veštačke inteligencije (*eng. Explainable AI, XAI*), koje su omogućile dublje razumevanje faktora koji najviše utiču na promene cena energije, interpretacijom modela koji je ostvario najbolje rezultate.

Generalno gledano, višeglavni LSTM modeli (*eng. MHLSTM*) imaju potencijal da značajno poboljšaju predikciju cena goriva jer mogu koristiti različite skupove podataka iz više izvora. Ovo omogućava veću tačnost i potencijalnu prilagodljivost modela, ali istovremeno dovodi i do veće računarske složenosti. Imajući ovo u vidu, kao i svaki naučni rad, i ovo istraživanje vuče sa sobom određena ograničenja:

- Ograničeni resursi suzili su broj optimizacionih algoritama koji su mogli biti uključeni u simulacije.
- Ograničenja memorije nametnula su ograničen broj faktora koji se mogu uzeti u obzir tokom prognoze.
- Ograničenje veličinu populacije i broj iteracija po izvršavanju.
- Potencijalno, **više** RNN modela bi se moglo koristiti unutar generisanih višeglavnih mreža.

Imajući u vidu prethodno navedeno, budući radovi koji budu dalje istraživali ovu oblast imaju za cilj da prevaziđu ova ograničenja, kako postanu dostupni dodatni resursi. Takođe, planira se dalja primena predloženog modela u drugim kontekstima i za rešavanje nekih drugih kompleksnijih problema.

Literatura

- [1] P. L. Ackerman, R. Kanfer, and M. E. Beier, “Trait complex, cognitive ability, and domain knowledge predictors of baccalaureate and postbaccalaureate educational attainment,” *Journal of Personality*, vol. 76, no. 4, pp. 409–438, 2008.
- [2] K. Yan, W. Li, Z. Ji, M. Qi, and Y. Du, “A hybrid LSTM neural network for energy consumption forecasting of individual households,” *IEEE Access*, vol. 7, pp. 157633–157642, 2019.
- [3] M. Wang, W. Wang, and L. Wu, “Application of a new grey multivariate forecasting model in the forecasting of energy consumption in 7 regions of China,” *Energy*, vol. 243, 123024, 2022.
- [4] M. Chaitanya Lahari, D. H. Ravi, and R. Bharathi, “Fuel price prediction using RNN,” in 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, pp. 1510–1514, 2018.
- [5] S. Haldenbilen, “Fuel price determination in transportation sector using predicted energy and transport demand,” *Energy Policy*, vol. 34, no. 17, pp. 3078–3086, 2006.
- [6] N. Jin et al., “Highly accurate energy consumption forecasting model based on parallel LSTM neural networks,” *Advanced Engineering Informatics*, vol. 51, 101442, 2022.
- [7] S. M. Almufti, R. B. Marqas, P. S. Othman, and A. B. Sallow, “Single-based and population-based metaheuristics for solving NP-hard problems,” *Iraqi Journal of Science*, 2021.
- [8] S. Asif, M. Zhao, F. Tang, Y. Zhu, and B. Zhao, “Metaheuristics optimization-based ensemble of deep neural networks for mpox disease detection,” *Neural Networks*, vol. 167, pp. 342–359, 2023.
- [9] B. Zhao, F. Luo, H. Lin, and D. Liu, “Particle swarm optimized neural networks based local tracking control scheme of unknown nonlinear interconnected systems,” *Neural Networks*, vol. 134, pp. 54–63, 2021.
- [10] S.-J. Bu and S.-B. Cho, “Time series forecasting with multi-headed attention based deep learning for residential energy consumption,” *Energies*, vol. 13, no. 18, 4722, 2020.
- [11] E. Wang, Q. Yu, Y. Chen, W. Slamu, and X. Luo, “Multi-modal knowledge graphs representation learning via multi-headed self-attention,” *Information Fusion*, vol. 88, pp. 78–85, 2022.

- [12] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proceedings of ICNN'95-International Conference on Neural Networks, vol. 4, IEEE, pp. 1942–1948, 1995.
- [13] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, pp. 67–82, 1997.
- [14] D. Polap and M. Woźniak, "Red fox optimization algorithm," Expert Systems with Applications, vol. 166, 114107, 2021.
- [15] L. Abualigah et al., "Reptile search algorithm (RSA): A nature-inspired meta-heuristic optimizer," Expert Systems with Applications, vol. 191, 116158, 2022.
- [16] L. Jovanovic et al., "Multi-step crude oil price prediction based on LSTM approach tuned by Salp Swarm Algorithm with disputation operator," Sustainability, vol. 14, no. 21, 14616, 2022.
- [17] H. Lin and Q. Sun, "Crude oil prices forecasting: an approach using CEEMDAN-based multi-layer GRU networks," Energies, vol. 13, no. 7, 1543, 2020.
- [18] G. I. Kim and B. Jang, "Petroleum price prediction with CNN-LSTM and CNN-GRU using skip-connection," Mathematics, vol. 11, no. 3, 547, 2023.
- [19] J. Li and J. Wang, "Stochastic recurrent wavelet neural network with EEMD method on energy price prediction," Soft Computing, vol. 24, no. 22, pp. 17133–17151, 2020.
- [20] K. Prakhar et al., "Effective stock price prediction using time series forecasting," in 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI), IEEE, pp. 1636–1640, 2022.
- [21] S. M. Almufti et al., "Overview of metaheuristic algorithms," Polaris Global Journal of Scholarly Research and Trends, vol. 2, no. 2, pp. 10–32, 2023.
- [22] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," Applied Soft Computing, vol. 8, no. 1, pp. 687–697, 2008.
- [23] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," International Journal of Bio-Inspired Computation, vol. 2, no. 2, pp. 78–84, 2010.
- [24] X.-S. Yang, "Bat algorithm for multi-objective optimisation," International Journal of Bio-Inspired Computation, vol. 3, no. 5, pp. 267–274, 2011.
- [25] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," Knowledge-Based Systems, vol. 96, pp. 120–133, 2016.

- [26] S. Mirjalili and S. Mirjalili, “Genetic algorithm,” in *Evolutionary Algorithms and Neural Networks: Theory and Applications*, 2019, pp. 43–55.
- [27] H. Jia, H. Rao, C. Wen, and S. Mirjalili, “Crayfish optimization algorithm,” *Artificial Intelligence Review*, vol. 56, Suppl. 2, pp. 1919–1979, 2023.
- [28] L. Dong, S. Huang, W. Wei, et al., “Unified language model pre-training for natural language understanding and generation,” in *NIPS*, 2019.
- [29] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *ICLR*, 2015.
- [30] G. Hinton, L. Deng, D. Yu, et al., “Deep neural networks for acoustic modeling in speech recognition,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.
- [31] R. Collobert and J. Weston, “A unified architecture for natural language processing: Deep neural networks with multitask learning,” in *ICML*, 2008, pp. 160–167.
- [32] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*, Springer, 2012.
- [33] L. Jovanović, N. Bačanin, D. Jovanović, et al., “Hybrid metaheuristics for LSTM hyperparameter optimization in crude oil price prediction,” *Expert Systems with Applications*, vol. 197, p. 116754, 2022.
- [34] S. Mirjalili, “The ant lion optimizer,” *Advances in Engineering Software*, vol. 83, pp. 80–98, 2015.
- [35] S. Mirjalili, “Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems,” *Neural Computing and Applications*, vol. 27, pp. 1053–1073, 2016.
- [36] S. Mirjalili, “Grey wolf optimizer,” *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [37] N. Bačanin, L. Jovanović, and M. Živković, “A comprehensive review of metaheuristic algorithms for neural network hyperparameter tuning,” *Applied Soft Computing*, vol. 123, p. 108988, 2022.
- [38] F. Pedregosa, G. Varoquaux, A. Gramfort, et al., “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [39] W. McKinney, *Python for Data Analysis*, 2nd ed., O’Reilly, 2017.
- [40] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *KDD*, 2016, pp. 785–794.

- [41] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [42] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.
- [43] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, pp. 273–297, 1995.
- [44] J. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.
- [45] R. Kohavi, “A study of cross-validation and bootstrap for accuracy estimation and model selection,” in *IJCAI*, 1995, pp. 1137–1145.
- [46] G. Schwarz, “Estimating the dimension of a model,” *Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.
- [47] H. Akaike, “A new look at the statistical model identification,” *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.
- [48] M. Lichman, “UCI machine learning repository,” 2013. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [49] Y. Gal and Z. Ghahramani, “Dropout as a Bayesian approximation: Representing model uncertainty in deep learning,” in *ICML*, 2016, pp. 1050–1059.
- [50] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016, pp. 770–778.
- [51] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *NAACL-HLT*, 2019, pp. 4171–4186.
- [52] R. Tibshirani, “Regression shrinkage and selection via the Lasso,” *Journal of the Royal Statistical Society: Series B*, vol. 58, no. 1, pp. 267–288, 1996.
- [53] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*, Springer, 2001.
- [54] P. J. Rousseeuw and A. Leroy, *Robust Regression and Outlier Detection*, Wiley, 1987.
- [55] D. Cohn, L. Atlas, and R. Ladner, “Improving generalization with active learning,” *Machine Learning*, vol. 15, pp. 201–221, 1994.
- [56] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed., Pearson, 2010.
- [57] N. J. Nilsson, *Artificial Intelligence: A New Synthesis*, Morgan Kaufmann, 1998.

- [58] N. Bačanin, V. Simić, M. Živković, M. Alrasheedi, and A. Petrović, “Cloud computing load prediction by decomposition reinforced attention long short-term memory network optimized by modified particle swarm optimization algorithm,” *Annals of Operations Research*, 2023, pp. 1–34.
- [59] B. Predić, L. Jovanović, V. Simić, N. Bačanin, M. Živković, P. Spalević, N. Budimirović, and M. Dobrojević, “Cloud-load forecasting via decomposition-aided attention recurrent neural network tuned by modified particle swarm optimization,” *Complex & Intelligent Systems*, vol. 10, no. 2, pp. 2249–2269, 2024.
- [60] N. Bačanin, M. Antonijević, T. Bezdan, M. Živković, K. Venkatachalam, and S. Malebary, “Energy efficient offloading mechanism using particle swarm optimization in 5G enabled edge nodes,” *Cluster Computing*, vol. 26, no. 1, pp. 587–598, 2023.
- [61] A. Abdo, O. Abdelkader, and L. Abdel-Hamid, “SA-PSO-GK++: A new hybrid clustering approach for analyzing medical data,” *IEEE Access*, 2024.
- [62] S. Subramani and M. Selvi, “Multi-objective PSO based feature selection for intrusion detection in IoT based wireless sensor networks,” *Optik*, vol. 273, 2023, p. 170419.
- [63] G. Nagarajan and P. Sajith, “Optimization of BPN parameters using PSO for intrusion detection in cloud environment,” *Soft Computing*, 2023, pp. 1–12.
- [64] T. Cuong-Le, T. Nghia-Nguyen, S. Khatir, P. Trong-Nguyen, S. Mirjalili, and K. D. Nguyen, “An efficient approach for damage identification based on improved machine learning using PSO-SVM,” *Engineering with Computers*, 2022, pp. 1–16.
- [65] M. Esfandyari, A. A. Delouei, and A. Jalai, “Optimization of ultrasonic excited double-pipe heat exchanger with machine learning and PSO,” *International Communications in Heat and Mass Transfer*, vol. 147, 2023, p. 106985.
- [66] F. Ergen and M. Katlav, “Investigation of optimized machine learning models with PSO for forecasting the shear capacity of steel fiber-reinforced SCC beams with/out stirrups,” *Journal of Building Engineering*, vol. 83, 2024, p. 108455.
- [67] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, “Quasi-oppositional differential evolution,” in *IEEE Congress on Evolutionary Computation*, 2007, pp. 2229–2236.
- [68] S. Cheng and Y. Shi, “Diversity control in particle swarm optimization,” in *IEEE Symposium on Swarm Intelligence*, 2011, pp. 1–9.

- [69] V. Osuna-Enciso, E. Cuevas, and B. M. Castañeda, “A diversity metric for population-based metaheuristic algorithms,” *Information Sciences*, vol. 586, 2022, pp. 192–208.
- [70] J. Parker-Holder, A. Pacchiano, K. M. Choromanski, and S. J. Roberts, “Effective diversity in population-based reinforcement learning,” in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 18050–18062.
- [71] IEEE, 2022 IEEE Congress on Evolutionary Computation (CEC), 2022, pp. 2–2. doi:10.1109/CEC55065.2022.9870418.
- [72] N. Mladenović and P. Hansen, “Variable neighborhood search,” *Computers & Operations Research*, vol. 24, no. 11, pp. 1097–1100, 1997.
- [73] J. Gurrola-Ramos, A. Hernández-Aguirre, and O. Dalmau-Cedeño, “Colshade for real-world single-objective constrained optimization problems,” in *IEEE Congress on Evolutionary Computation (CEC)*, 2020, pp. 1–8.
- [74] S. F. Stefenon, M. H. D. M. Ribeiro, A. Nied, V. C. Mariani, L. D. S. Coelho, V. R. Q. Leithardt, L. A. Silva, and L. O. Seman, “Hybrid wavelet stacking ensemble model for insulators contamination forecasting,” *IEEE Access*, vol. 9, 2021, pp. 66387–66397.
- [75] Z. Liu, Y. Wen, X. Zhang, M. Wang, S. Xiao, Y. Chen, and L. He, “A novel rainfall forecast model using GNSS observations and CAPE in Singapore,” *Journal of Atmospheric and Solar-Terrestrial Physics*, vol. 253, 2023, p. 106158.
- [76] A. LaTorre, D. Molina, E. Osaba, J. Poyatos, J. Del Ser, and F. Herrera, “A prescription of methodological guidelines for comparing bio-inspired optimization algorithms,” *Swarm and Evolutionary Computation*, vol. 67, 2021, p. 100973.
- [77] G. V. Glass, “Testing homogeneity of variances,” *American Educational Research Journal*, vol. 3, no. 3, pp. 187–190, 1966.
- [78] S. S. Shapiro and R. Francia, “An approximate analysis of variance test for normality,” *Journal of the American Statistical Association*, vol. 67, no. 337, pp. 215–216, 1972.
- [79] N. E. Zamri, S. A. Azhar, M. A. Mansor, A. Alway, and M. S. M. Kasihmuddin, “Weighted random k satisfiability for k=1,2 (R2SAT) in discrete Hopfield neural network,” *Applied Soft Computing*, vol. 126, 2022, p. 109312.

- [80] N. E. Zamri, M. A. Mansor, M. S. M. Kasihmuddin, S. S. Sidik, A. Alway, N. A. Romli, Y. Guo, and S. Z. M. Jamaludin, “A modified reverse-based analysis logic mining model with weighted random 2 satisfiability logic in discrete Hopfield neural network and multi-objective training of modified niched genetic algorithm,” *Expert Systems with Applications*, vol. 240, 2024, p. 122307.
- [81] H. Moazen, S. Molaei, L. Farzinvash, and M. Sabaei, “PSO-ELPM: PSO with elite learning, enhanced parameter updating, and exponential mutation operator,” *Information Sciences*, vol. 628, 2023, pp. 70–91.
- [82] Z. Xing, J. Zhu, Z. Zhang, Y. Qin, and L. Jia, “Energy consumption optimization of tramway operation based on improved PSO algorithm,” *Energy*, vol. 258, 2022, p. 124848.
- [83] S.-P. Zhu, B. Keshtegar, M. E. A. B. Seghier, E. Zio, and O. Taylan, “Hybrid and enhanced PSO: Novel first order reliability method-based hybrid intelligent approaches,” *Computer Methods in Applied Mechanics and Engineering*, vol. 393, 2022, p. 114730.
- [84] Y. Liu, J. Dai, S. Zhao, J. Zhang, W. Shang, T. Li, Y. Zheng, T. Lan, and Z. Wang, “Optimization of five-parameter BRDF model based on hybrid GA-PSO algorithm,” *Optik*, vol. 219, 2020, p. 164978.
- [85] A. K. Balyan, S. Ahuja, U. K. Lilhore, S. K. Sharma, P. Manoharan, A. D. Algarni, H. Elmannai, and K. Raahemifar, “A hybrid intrusion detection model using EGA-PSO and improved random forest method,” *Sensors*, vol. 22, no. 16, 2022, p. 5986.
- [86] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 4765–4774.

PRILOZI

7.1 Prilog A – Pseudokovi metaheurističkih algoritama

7.1.1 PSO Algoritam

Algoritam 1 *Pseudokod PSO*

Input: broj čestica N, dimenzija D, funkcija cilja $f(x)$, max iteracija T

Output: najbolje rešenje gbest

Procedure:

Korak 1: Inicijalizuj položaje i brzine čestica nasumično.

Korak 2: Izračunaj vrednosti funkcije cilja $f(x)$ za svaku česticu.

Korak 3: Postavi pbest za svaku česticu i globalno najbolje rešenje gbest.

Korak 4: Za svaku iteraciju $t = 1$ do T:

Ažuriraj brzine čestica:

$$v_i = w \cdot v_i + c^1 \cdot r^1 \cdot (pbest_i - x_i) + c^2 \cdot r^2 \cdot (gbest - x_i)$$

Ažuriraj položaje: $x_i = x_i + v_i$

Izračunaj novu vrednost $f(x_i)$

Ažuriraj pbest i gbest ako je novo rešenje bolje

Korak 5: Vрати gbest.

7.1.2 Salp Swarm Algoritam (SSA)

Algoritam 1 *Pseudokod SSA*

Input: broj salpi N , funkcija cilja $f(x)$, max iteracija T

Output: najbolje rešenje F

Procedure:

Korak 1: Inicijalizuj populaciju salpi nasumično.

Korak 2: Za svaku iteraciju $t = 1$ do T :

Ažuriraj poziciju lidera salpe u odnosu na hranu (F).

Ažuriraj pozicije sledbenika koristeći lančanu strukturu.

Izračunaj vrednosti $f(x)$.

Ažuriraj F ako je pronađeno bolje rešenje.

Korak 3: Vрати F .

7.1.3 Artificial Rabbits Optimization Algoritam (ARO)

Algoritam 1 *Pseudokod ARO*

Input: populacija zečeva N , funkcija cilja $f(x)$, max iteracija T

Output: optimalno rešenje best

Procedure:

Korak 1: Inicijalizuj pozicije zečeva nasumično.

Korak 2: Za svaku iteraciju $t = 1$ do T :

Ako je mod = „skladištenje hrane“: ažuriraj pozicije zečeva oko najboljeg rešenja.

Ako je mod = „traganje za hranom“: nasumično istraži nove lokacije.

Izračunaj $f(x)$ za sve zečeve.

Ažuriraj best.

Korak 3: Vрати best.

7.1.4 Red Fox Optimization Algoritam Algoritam (RFO)

Algoritam 1 *Pseudokod RFO*

Input: broj lisica N , funkcija cilja $f(x)$, max iteracija T

Output: optimalno rešenje $best$

Procedure:

Korak 1: Inicijalizuj populaciju lisica.

Korak 2: Za svaku iteraciju $t = 1$ do T :

Ažuriraj poziciju „alfa“ lisice u potrazi za plenom.

Ažuriraj pozicije drugih lisica kombinovanjem eksploracije i eksploatacije.

Izračunaj $f(x)$.

Ažuriraj $best$.

Korak 3: Vрати $best$.

7.1.5 Bat Algoritam (BA)

Algoritam 1 *Pseudokod BA*

Input: broj slepih miševa N , funkcija cilja $f(x)$, max iteracija T

Output: optimalno rešenje $best$

Procedure:

Korak 1: Inicijalizuj pozicije i brzine slepih miševa.

Korak 2: Dodeli frekvenciju, glasnoću i stopu pulsa svakom mišu.

Korak 3: Za svaku iteraciju $t = 1$ do T :

Ažuriraj frekvenciju i brzinu.

Pomeri pozicije slepih miševa.

Ako je slučajni broj $<$ stopa pulsa, izaberi lokalno rešenje.

Ako je slučajni broj $<$ glasnoća i novo rešenje je bolje, prihvati ga.

Ažuriraj $best$.

Korak 4: Vрати $best$.

7.1.6 Firefly Algoritam (FA)

Algoritam 1 *Pseudokod FA*

Input: broj svitaca N , funkcija cilja $f(x)$, max iteracija T

Output: optimalno rešenje best

Procedure:

Korak 1: Inicijalizuj pozicije svitaca.

Korak 2: Za svaku iteraciju $t = 1$ do T :

Za svaki par svitaca (i, j) :

Ako je $f(x_j)$ bolji od $f(x_i)$, pomeri i ka j :

$$x_i = x_i + \beta_0 \cdot e^{(-\gamma r^2)}(x_j - x_i) + \alpha \cdot \varepsilon$$

Ažuriraj best.

Korak 3: Vрати best.

7.1.7 Genski Algoritam (GA)

Algoritam 1 *Pseudokod GA*

Input: početna populacija, funkcija cilja $f(x)$, max generacija G

Output: optimalno rešenje best

Procedure:

Korak 1: Inicijalizuj populaciju nasumično.

Korak 2: Za svaku generaciju $g = 1$ do G :

Izračunaj fitnes svih jedinki.

Selektuj jedinke za ukrštanje.

Primeni ukrštanje (crossover).

Primeni mutaciju.

Formiraj novu populaciju.

Ažuriraj best.

Korak 3: Vрати best.

7.2 Prilog B – Pseudokodi modela neuronskih mreža

7.2.1 Algoritam Rekurentne Neuronske mreže

Algoritam 1 *Pseudokod Rekurentne Neuronske mreže (RNN)*

Input: ulazne sekvence X, ciljevi Y, broj epoha E

Output: trenirani model

Procedure:

Korak 1: Inicijalizuj težine nasumično.

Korak 2: Za svaku epohu $e = 1$ do E:

Za svaki vremenski korak t :

$$h_t = f(W_{xh} \cdot x_t + W_{hh} \cdot h_{t-1} + b_h)$$

$$y_t = W_{yh} \cdot h_t + b_y$$

Izračunaj grešku (loss).

Izvrši propagaciju unazad kroz vreme (BPTT).

Ažuriraj težine.

Korak 3: Vрати trenirani model.

7.2.2 Algoritam Dugoročno-kratkoročne memorije

Algoritam 1 *Pseudokod Dugoročno-kratkoročne memorije (LSTM)*

Input: ulazne sekvence X, ciljevi Y, broj epoha E

Output: trenirani model

Procedure:

Korak 1: Inicijalizuj težine i stanja ćelija.

Korak 2: Za svaku epohu $e = 1$ do E:

Za svaki vremenski korak t :

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t$$

$$h_t = o_t \cdot \tanh(c_t)$$

Izračunaj grešku (loss).

Izvrši propagaciju unazad kroz vreme (BPTT).

Ažuriraj težine.

Korak 3: Vрати trenirani model.

7.2.3 Algoritam Zatvorene rekurentne jedinice (GRU)

Algoritam 1 *Pseudokod zatvorene rekurentne jedinice (GRU)*

Input: ulazne sekvence X , ciljevi Y , broj epoha E

Output: trenirani model

Procedure:

Korak 1: Inicijalizuj težine i skrivena stanja.

Korak 2: Za svaku epohu $e = 1$ do E :

Za svaki vremenski korak t :

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

$$\tilde{h}_t = \tanh(W_h \cdot [r_t \cdot h_{t-1}, x_t])$$

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t$$

Izračunaj grešku (loss).

Izvrši propagaciju unazad kroz vreme (BPTT).

Ažuriraj težine.

Korak 3: Vрати trenirani model.

7.2.4 Algoritam Dugoročno-kratkoročne memorije sa više glava (MHLSTM)

Algoritam 1 *Pseudokod dugoročno-kratkoročne memorije sa više glava (MHLSTM)*

Input: ulazne sekvence X, ciljevi Y, broj epoha E

Output: trenirani model

Procedure:

Korak 1: Inicijalizuj H LSTM pod-modela.

Korak 2: Za svaku epohu $e = 1$ do E:

Za svaki vremenski korak t :

Izračunaj izlaze svih H LSTM glava paralelno.

Izračunaj grešku (loss).

Izvrši propagaciju unazad kroz vreme (BPTT).

Ažuriraj težine.

Korak 3: Vрати trenirani model.

Spisak akronima

AI (*Artificial Intelligence*) – veštačka inteligencija

AHP (*Analytic Hierarchy Process*) – metoda višekriterijumskog odlučivanja

AR (*Autoregressive*) – autoregresivni model

ARIMA (*Autoregressive Integrated Moving Average*) – integrisani autoregresivni model pokretnih proseka

ARMA (*Autoregressive Moving Average*) – autoregresivni model pokretnih proseka

ARAS (*Additive Ratio Assessment*) – metoda višekriterijumskog odlučivanja

BA (*Bat Algorithm*) – algoritam optimizacije inspirisan ponašanjem slepih miševa

BPTT (*Backpropagation Through Time*) – algoritam za treniranje rekurentnih neuronskih mreža propagacijom unazad kroz vreme

CEEMDAN (*Complete Ensemble Empirical Mode Decomposition with Adaptive Noise*) – metoda dekompozicije sa adaptivnim šumom

DP (*Dynamic Programming*) – dinamičko programiranje

FA (*Firefly Algorithm*) – algoritam optimizacije inspirisan ponašanjem svitaca

GA (*Genetic Algorithm*) – genetski algoritam

GRU (*Gated Recurrent Unit*) – rekurentna neuronska jedinica sa kontrolnim vratima

GREY (*Grey Relational Analysis*) – analiza sivih relacija

LSTM (*Long Short-Term Memory*) – duga kratkoročna memorija (rekurentna neuronska mreža)

MA (*Moving Average*) – model pokretnih proseka

MHLSTM (*Multi-Head Long Short-Term Memory*) – LSTM sa više glava

MSE (*Mean Squared Error*) – srednja kvadratna greška

NFLT (*No Free Lunch Theorem*) – teorema bez besplatnog ručka

NP (*Non-deterministic Polynomial time*) – klasa problema rešivih u nedeterminističkom polinomijalnom vremenu

OF (*Objective Function*) – ciljna funkcija

PSO (*Particle Swarm Optimization*) – optimizacija rojem čestica

QRL (*Q-learning Reinforcement Learning*) – učenje pojačavanjem zasnovano na Q-funkciji

RFO (*Red Fox Optimization*) – algoritam optimizacije inspirisan ponašanjem crvene lisice

RNN (*Recurrent Neural Network*) – rekurentna neuronska mreža

RMSE (*Root Mean Squared Error*) – koren srednje kvadratne greške

SAGPSO (*Self-Adaptive Genetic Particle Swarm Optimization*) – samoadaptivna kombinacija genetskog algoritma i optimizacije rojem čestica

SARIMA (*Seasonal Autoregressive Integrated Moving Average*) – sezonski integrisani autoregresivni model pokretnih proseka

SSA (*Salp Swarm Algorithm*) – algoritam optimizacije inspirisan ponašanjem morskih puževa (salpi)

XAI (*Explainable Artificial Intelligence*) – objašnjiva veštačka inteligencija

Spisak oznaka

$\mathbf{x}_t$ – ulazni vektor u trenutku t

$\mathbf{y}_t$ – izlazni vektor u trenutku t

$\mathbf{h}_t$ – skriveno stanje u trenutku t

$\mathbf{c}_t$ – stanje ćelije (cell state) u LSTM-u

$\mathbf{f}_t$ – vektor izlaza forget gate u LSTM-u

$\mathbf{i}_t$ – vektor izlaza input gate u LSTM-u

$\mathbf{o}_t$ – vektor izlaza output gate u LSTM-u

$\tilde{\mathbf{c}}_t$ – kandidat vrednosti za stanje ćelije u LSTM-u

$\mathbf{z}_t$ – update gate u GRU

$\mathbf{r}_t$ – reset gate u GRU

$\tilde{\mathbf{h}}_t$ – kandidat skrivenog stanja u GRU

$\mathbf{W}_f, \mathbf{W}_i, \mathbf{W}_o, \mathbf{W}_c$ – matrice težina u LSTM-u

$\mathbf{W}_{hh}, \mathbf{W}_{xh}, \mathbf{W}_{hy}$ – matrice težina u RNN-u

$\mathbf{b}_f, \mathbf{b}_i, \mathbf{b}_o, \mathbf{b}_c$ – vektori praga (bias) u LSTM-u

σ (sigmoid) – sigmoidna funkcija aktivacije

$\tanh$ – hiperbolička tangens funkcija aktivacije

ϵ – nasumična varijabla za poremećaj (random noise)

β_0 – osnovni koeficijent privlačnosti u Firefly algoritmu

γ – parametar za prigušivanje svetlosti u Firefly algoritmu

α – parametar za jačinu slučajnog poremećaja

w – inercioni faktor u PSO algoritmu

c_1, c_2 – koeficijenti u PSO algoritmu (kognitivna i socijalna komponenta)

r_1, r_2 – slučajne vrednosti iz $[0,1]$ u PSO algoritmu

g_{best} – globalno najbolje rešenje u PSO algoritmu

p_{best} – najbolje rešenje čestice u PSO algoritmu

N – veličina populacije (broj jedinki, čestica, svitaca, lisica, zečeva...)

T – broj iteracija (trajanje algoritma)

E – broj epoha u treniranju neuronske mreže

G – broj generacija u GA

H – broj glava u MHLSTM arhitekturi

Spisak tabela

TABELA 1 PARALELA IZMEĐU TRADICIONALNOG PROGRAMIRANJA I VEŠTAČKE INTELIGENCIJE	32
TABELA 2 PRIKAZ PREDNOSTI I NEDOSTAKA OVIH MODELA	35
TABELA 3 PRIKAZ PREDNOSTI I NEDOSTAKA KLASIČNIH MODELA.....	43
TABELA 4 ANALIZA STOHAISTIČKIH I MODERNIH MODELA ZA OBRADU VREMENSKIH SERIJA.....	44
TABELA 5 TABELARNI SKRAĆENI PREGLED LITERATURE I NAUČNIH RADOVA IZ OVE OBLASTI.....	47
TABELA 6 ANALIZA MODELA DUBOKOG UČENJA I KLASIČNIH MODELA MAŠINSKOG UČENJA	54
TABELA 7 PRIKAZ ČESTO KORIŠĆENIH AKTIVACIONIH FUNKCIJA	59
TABELA 8 PRIKAZ TIPOVA KAPIJA I NJIHOVE PRIMENE	63
TABELA 9 POREĐENJE LSTM I GRU MODELA	67
<i>TABELA 10 ANALIZA MREŽNE I NASUMIČNE PRETRAGE</i>	<i>77</i>
<i>TABELA 11 PREDNOSTI I NEDOSTACI PSO ALGORITMA</i>	<i>80</i>

Spisak slika

SLIKA 1 PROCES OBRADE ENERGIJE	24
ILUSTRACIJA 2 FAKTORI KOJI MOGU UTICATI NA CENU GORIVA	27
ILUSTRACIJA 3 PRIKAZ RAZLIKA IZMEĐU PROGRAMIRANJA I MAŠINSKOG UČENJA	31
ILUSTRACIJA 4 ISTORIJAT RAZVOJA MODELA MAŠINSKOG UČENJA	53
ILUSTRACIJA 5 ARHITEKTURA NEURONSKE MREŽE	57
ILUSTRACIJA 6 ARHITEKTURA NEURONA	58
ILUSTRACIJA 7 ARHITEKTURA REKURENTNIH NEURONSKIH MREŽA	60
ILUSTRACIJA 8 ARHITEKTURA VIŠEGLAVNIH MODELA	71
ILUSTRACIJA 9 CROSS-OVER MEHANIZAM	90
ILUSTRACIJA 10 MEHANIZAM MUTACIJE	91
ILUSTRACIJA 11 MHLSTM MREŽA	93
ILUSTRACIJA 12 PREDLOŽENI OKVIR	94
ILUSTRACIJA 13 PRIKAZ NAČINA KODIRANJA PARAMETARA	94
ILUSTRACIJA 14 VIZUELNI PRIKAZ SKUPA PODATAKA	99
ILUSTRACIJA 15 REZULTATI FUNKCIJE CILJA ZA SVAKI OPTIMIZATOR	100
ILUSTRACIJA 16 DISTRIBUCIJA REZULTATA FUNKCIJE CILJA	101
ILUSTRACIJA 17 DISTRIBUCIJA REZULTATA INDIKATORSKE FUNKCIJE	101
ILUSTRACIJA 18 DETALJNI PRIKAZ METRIKE MODELA SA NAJBOJIM PERFORMANSAMA	102
ILUSTRACIJA 19 DETALJNI PRIKAZ METRIKE MODELA PO KORAKU PROGNOZE	103
ILUSTRACIJA 20 PRIKAZ STOPA KONVERGENCIJE REZULTATA FUNKCIJE CILJA ZA SVAKI OPTIMIZATOR	104
ILUSTRACIJA 21 UPOREDNA ANALIZA STOPA KONVERGENCIJE REZULTATA FUNKCIJE CILJA ZA SVAKI OPTIMIZATOR	105
ILUSTRACIJA 22 PRIKAZ STOPA KONVERGENCIJE REZULTATA INDIKATORSKE FUNKCIJE ZA SVAKI OPTIMIZATOR	105
ILUSTRACIJA 23 UPOREDNA ANALIZA STOPA KONVERGENCIJE REZULTATA INDIKATORSKE FUNKCIJE ZA SVAKI OPTIMIZATOR	106
ILUSTRACIJA 24 MODELI SA NAJBOLJIM PREDIKCIJAMA	106
ILUSTRACIJA 25 PRIKAZ ARHITEKTURE PREDLOŽENOG MODELA SA OSTVARENIM NAJBOLJIM REZULTATIMA	107

<i>ILUSTRACIJA 26 IZABRANI PARAMETRI ZA PREDLOŽENI MODEL</i>	<i>107</i>
<i>ILUSTRACIJA 27 REZULTATI FUNKCIJE CILJA KROZ KDE DIJAGrame ZA SVAKI OPTIMIZATOR.....</i>	<i>109</i>
<i>ILUSTRACIJA 28 REZULTATI SHAPIRO-WILK TESTA</i>	<i>109</i>
<i>ILUSTRACIJA 29 REZULTATI WILCOXON TESTA</i>	<i>110</i>
<i>ILUSTRACIJA 30 REZULTATI ANALIZE SENZITIVNOSTI PSO ALGORITMA SA MODIFIKACIJAMA.....</i>	<i>111</i>
<i>ILUSTRACIJA 31 REZULTATI ANALIZE SENZITIVNOSTI PSO ALGORITMA U ODNOSU NA SLIČNE ALGORITME</i>	<i>112</i>
<i>ILUSTRACIJA 32 GRAFIČKI PRIKAZ ZNAČAJA KARAKTERISTIKA NAJBOLJEG MODELA KORIŠĆENJEM SHAP METODE</i>	<i>114</i>

Spisak formula

JEDNAČINA 1 AR MODEL	38
JEDNAČINA 2 MA MODEL	39
JEDNAČINA 3 ARMA MODEL	40
JEDNAČINA 4 ARIMA I SARIMA MODEL	41
JEDNAČINA 5 SARIMA MODEL.....	42
JEDNAČINA 6 MATEMATIČKI PRIKAZ NEURONA	57
JEDNAČINA 7 MODEL RNN MREŽE	61
JEDNAČINA 8 PRETHODNO STANJE ČELIJE	64
JEDNAČINA 9 IZLAZ SIGMOIDALNE FUNKCIJE	64
JEDNAČINA 10 KONAČNI IZLAZ PROSLEĐEN TANH FUNKCIJI.....	64
JEDNAČINA 11 RESET KAPIJA	66
JEDNAČINA 12 AŽURIRAJUĆA KAPIJA.....	66
JEDNAČINA 13 KANDIDATSKO SKRIVENO STANJE NEURONA	66
JEDNAČINA 14 AŽURIRANJE KANDIDATSKOG STANJA	66
JEDNAČINA 15 OPŠTA FORMULA RNN MODELA SA VIŠE GLAVA.....	68
JEDNAČINA 16 IZLAZ MREŽE	69
JEDNAČINA 17 NASUMIČNI VEKTOR X.....	69
JEDNAČINA 18 FUNKCIJA POREĐENJA	69
JEDNAČINA 19 PRIMENA SOFTMAX FUNKCIJE ZA RAČUNANJE DISTRIBUCIJE VEROVATNOĆE.....	69
JEDNAČINA 20 KONAČAN OBLIK MEHANIZMA PAŽNJE.....	70
JEDNAČINA 21 ADAPTIVNI MEHANIZAM PAŽNJE.....	70
JEDNAČINA 22 MREŽNA PRETRAGA	75
JEDNAČINA 23 NASUMIČNA PRETRAGA.....	76
JEDNAČINA 24 AŽURIRANJE BRZINE ČESTICE	80
JEDNAČINA 25 AŽURIRANJE POZICIJE ČESTICE	80
JEDNAČINA 26 DEFINICIJA REFLEKTIVNE ČESTICE	85
JEDNAČINA 27 KVAZI-REFLEKSNO SUPROTNA KOMPONENTA ZA SVAKU KOMPONENTU X_j	86
JEDNAČINA 28 PROSEČNA VREDNOSTI DIMENZIJE J	87
JEDNAČINA 29 DIVERZITET PO DIMENZIJI J	87
JEDNAČINA 30 UKUPNA RAZNOLIKOST POPULACIJE.....	88
JEDNAČINA 31 RAČUNANJE POČETNE VREDNOSTI PRAGA Dt_0	88
JEDNAČINA 32 SMANJIVANJE VREDNOSTI PRAGA Dt	89

<i>JEDNAČINA 33 POVEĆANJE VREDNOSTI PC U SVAKOJ ITERACIJI</i>	<i>89</i>
<i>JEDNAČINA 34 SREDNJA KVADRATNA GREŠKA KORENA</i>	<i>97</i>
<i>JEDNAČINA 35 SREDNJA APSOLUTNA GREŠKA.....</i>	<i>97</i>
<i>JEDNAČINA 36 SREDNJA KVADRATNA GREŠKA</i>	<i>97</i>
<i>JEDNAČINA 37 KOEFICIJENT DETERMINACIJE</i>	<i>97</i>
<i>JEDNAČINA 38 INDEKS SLAGANJA</i>	<i>98</i>