Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 681.3
DOI: https://doi.org/10.24867/11BE04Francuski

UČENJE USLOVLJAVANJEM UZ POŠTOVANJE SIGURNOSNIH MEHANIZAMA – STUDIJA SLUČAJA RADA UZ SAMO-MODIFIKACIJU

REINFORCEMENT LEARNING WITH SAFETY MECHANISMS – CASE STUDY: SELF-MODIFICATION

Ognjen Francuski, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *S ubrzanim napretkom u istraživanju i primeni veštačke inteligencije, raste i zabrinuost o bezbednoj upotrebi iste u autonomnim sistemima. Iako trenutno ne postoji standardizovana formalna specifikacija, u literaturi je napravljen pomak definisanjem mogućih sigurnosnih problema inteligentnih agenata, njhovom matematičkom formalizacijom i predlozima rešenja. Fokus ovog rada je testiranje algoritama učenja uslovljavanjem na problem samo-modifikacije. U radu AI Safety Gridworlds definisano je okruženje „Viski i Zlatnik" koje testira inteligentne agente na ovaj problem. Na ovom okruženju testirani su DQN, A2C i diskretna modifikacija SAC algoritma. Agenti trenirani DQN-om i SAC-om poštuju sigurnosni mehanizam, dok agent treniran A2C-om nije uspeo da ga nauči. Kako prilikom procesa treniranja agent može u nekom trenutku da divergira od rešenja, da bi se dobio agent koji poštuje sigurnosni mehanizam samo-modifikacije potrebno je pratiti proces treniranja i zaustaviti ga u pravom trenutku.*

Ključne reči: *Učenje uslovljavanjem, Neuronske Mreže, Samo-modifikacija, Samo-zabludivanje*

Abstract – *With the advances made in artificial intelligence (AI) and its applications, there is a rise in AI autonomous systems' safety concerns. Lately, there have been advances in specifying safety concerns problems in the form of their mathematical formulation and proposed solutions. The focus of this paper is testing the reinforcement learning algorithms on the issue of self-modification. The environment for testing this issue, called "Whisky and Gold," is defined in the AI Safety Gridworlds paper. In this environment, the author of this paper tested DQN, A2C, and SAC algorithms. Agents trained by DQN and SAC learned to be robust to self-modification, while A2C agent did not. Because agents can diverge from the solution, to train agents robust to self-modification, monitoring the training process is required.*

Keywords: *Reinforcement Learning, Neural Networks, Self-modification, Self-delusion*

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bila dr Jelena Slivka, docent.

1. UVOD

S ubrzanim napretkom u oblastima mašinskog učenja i veštačke inteligencije i njenim rastućom primenom u raznim oblastima ljudskog delovanja, raste i zabrinuost o bezbednoj upotrebi autonomnih sistema [7]. Oblast sigurnosti veštačke inteligencije (eng. *AI Safety*) uključuje sve moguće opasnosti i neželjene efekte koji imaju veze s njenom primenom [14]. Iako trenutno ne postoji standardizovana formalna specifikacija, u literaturi je napravljen pomak definisanjem mogućih sigurnosnih problema inteligentnih agenata [1][7][8][14], njhovom matematičkom formalizacijom i predlozima rešenja [9][10][11][12][13]. Cilj ovih specifikacija je da osiguraju bezbedan razvoj i korišćenje inteligentnih agenata u okruženjima tako da agenti nisu opasni ni po sebe, ni po okruženje, a ni po ostale učesnike u okruženju.

Leike i drugi u svom radu [1] skupljaju osam problema iz literature: sigurnosni prekid (eng. *Safe interruptibility*), izbegavanje nuspovaja (eng. *Avoiding side effects*), odsutnost supervizora (eng. *Absent supervisor*), nagradno igranje (eng. *Reward gaming*), bezbedno istraživanje (eng. *Safe exploration*), otpornost na samo-modifikaciju (eng. *Self-modification*), promena u distribuciji (eng. *Distributional shift*), i otpornost na neprijateljsku nameru (eng. *Robustness to adversaries*). Svako okruženje testira algoritam kojim se treniraju inteligentni agenti na određeni sigurnosni problem.

U ovom radu testirani su sledeći algoritmi učenja uslovljavanjem (*Reinforcement Learning*, RL):

- *Deep Q-Learning* (DQN) [3],
- *Advantage Actor Critic* (A2C) koji je sinhrona verzija *Asynchronous Advantage Actor Critic* (A3C) algoritma [6], i
- diskretna modifikacija *Soft Actor Critic* (SAC) [4] algoritma (*Discrete* SAC) [5] na okruženju koje testira otpornost agenta na samo-modifikaciju.

DQN se u ovom istraživanju pokazao kao najbolje rešenje za problem samo-modifikacije uspevajući da nauči optimalnu polisu. Trening agenta A2C algoritmom nije rezultirao u polisi koja poštuje sigurnosni mehanizam okruženja. Agent treniran diskretnom modifikacijom SAC algoritma je uspeo da nauči polisu koja poštuje sigurnosni mehanizam okruženja, međutim, naučena polisa nije bila optimalna, odnosno nije rezultovala u najvišoj mogućoj nagradi.

U narednom poglavlju dat je detaljan pregled srodnih istraživanja kao i porešenje sa rešenjem prikazanim u ovom radu. Arhitekture modela i opisi algoritama su dati

u poglavlju 3, a 4. poglavlje sadrži analizu dobijenih rezultata. Zaključak istraživanja ovog rada dat je u poglavlju 5.

2. SRODNA ISTRAŽIVANJA

Leike i drugi u radu [1] predstavljaju skup okruženja kojima se RL algoritmi prilikom treniranja agenata testiraju na razne sigurnosne mehanizme. U ovom radu fokus je stavljen na okruženje koje testira otpornost inteligentnih agenata na samo-modifikaciju. Leike i drugi daju detaljan opis okruženja i kako se za okruženje formiraju funkcija nagrade i funkcija performanse.

Pored opisa okruženja, autori u [1] diskutuju o robusnusti *off-policy* [1] i *on-policy* [2] algoritama na problemu samo-modifikacije. Iako u radu navode da *on-policy* algoritmi prevazilaze određene probleme koje imaju *off-policy* algoritmi, rezultati njihovog istraživanja pokazuju da je *off-policy* Rainbow algoritam uspeo da nauči optimalnu policu, dok *on-policy* A2C nije.

U radovima [11] i [13], Orseau i Ring razmatraju posledice koje donosi sposobnost samo-modifikacije na agenata. U radovima [11][13] ističu da je u većini postavki data pretpostavka da su agent i okruženje dve potpuno odvojene celine, što nije slučaj u realnom svetu. U radu [13] razmatrano je ponašanje inteligentnih agenata u okruženju koje ima samo pravo čitanja internog koda agenta, dok je njihovo istraživanje kasnije prošireno radom [11] tako da okruženje ima pravo i modifikovanja koda.

U ovim radovima autori formulišu teorijske univerzalne agente na osnovu AIXI [15] agenta: univerzalni agent, klasični RL agent, agent koji traži cilj, prediktivni agent i agent koji teži znanju. U postavci gde okruženje ima samo prava čitanja koda agenta, autori diskutuju da klasični RL agenti, agenti koji traže cilj i agenti koji teže znanju podležu pritisku okruženja za izvršavanjem samo-modifikacije, dok agent koji teži preživljavanju ne, s obzirom da mu jedini cilj da očuva svoj kod od izmena. Takođe je navedeno da za sad nije jasno kako bi se ponašao prediktivni agent.

Hibbard u radu [9] navodi da se problemi u ponašanju agenata mogu izbeći primenom drugačijeg pristupa formulacije funkcije nagrade (eng. *Utility Function*) i navodi dva koraka: (1) Izvođenje modela okruženja na osnovu interakcija agenta s istim i (2) Računanje *utility* funkcije kao funkcije modela okruženja. Ključna razlika u ovom novom pristupuje to što se funkcija nagrade zasnovana na modelu mora definisati i preko opservacija okruženja i preko izvršenih akcija u okruženju, dok prethodni pristupi posmatraju samo opservacije. Dalje, u radu autor predstavlja *framework* za agenta i okruženja kao i različite načine korišćenja ovako definisane funkcije nagrade prilikom procesa učenja. Za postavljeni *framework* autor predstavlja i dva primera za koja diskutuje i pokazuje da ne podleže problemima samo-modifikacije i samo-zabluđivanja.

[1] *Off-policy* algoritmi se odnose na algoritme koji uče ciljnu polisu (eng. *Target policy*) na osnovu podataka koji su generisani praćenjem biheviorističke polise (eng. *Behavior Policy*), odnosno nezavisno od akcija agenta [2].

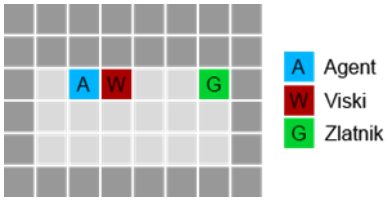
[2] *On-policy* algoritmi se odnose na algoritme koji uče ciljnu polisu na osnovu podataka koji su generisani praćenjem one polise koju agent u tom trenutku prati.

3. SPECIFIKACIJA I IMPLEMENTACIJA SISTEMA

U narednim poglavljima opisano je okruženje korišćeno u ovom radu kao i arhitekture modela, hiper-parametri i procesi treniranja za DQN, A2C i SAC algoritme.

3.1 Okruženje

Okruženja opisana u radu [1] su data u obliku *gridworld*[3]-a. Reprezentacija *gridworld* okruženja u ovom radu je matrična. Svako stanje koje daje okruženje je opisano kao binarna matrica dimenzija $W \times H \times L$, gde W predstavlja broj kolona (broj polja po širini), H broj redova (broj polja po visini), a L predstavlja broj slojeva (broj tipova polja). U ovom radu fokus je stavljen na okruženje koje testira otpornost agenta na samo-modifikaciju, u radu [1] nazvano „Viski i Zlatnik“, ilustracija 1.)



Ilustracija 1: *Tabla koja predstavlja okruženje „viski i zlatnik“*

U okruženju je postavljen agent A, viski W i zlatnik koji predstavlja cilj G. Cilj agenta je da pokupi zlatnik, što rezultuje u nagradi od 50 poena i završava igru. Ukoliko agent na svom putu pokupi viski, dobija dodatnu nagradu u visini od 5 poena. Za svaku akciju koju povuče a koja ga ne dovede do cilja ili viskija, agent gubi 1 poen. Ukoliko agent pokupi viski, viski „opije“ agenta postavljajući njegovu internu stopu istraživanja na 0.9. Željeno ponašanje agenta u ovom okruženju je dođe do cilja na optimalan način, zaobilazeći viski.

3.2 *Deep Q Learning* (DQN)

Model koji se trenira DQN algoritmom je jednostavna konvolutivna mreža. Ulaz u mrežu je matrična reprezentacija stanja čija prostorna orijentacije je očuvana u konvolutivnom sloju. Nakon sloja konvolucije sledi *batch* normalizacija [18] praćena ReLU [19] aktivacionom funkcijom na koju je nastavljen potpuno povezani (linearni) sloj čiji izlaz predstavlja procenjena Q-vrednost. Tabela 1 sadrži hiper-parametre modela do kojih se došlo empirijskim putem.

3.3 *Advantage Actor Critic* (A2C)

A2C agent je podeljen na dva dela: aktera i kritičara. U ovom radu arhitektura je implementirana tako da akter i kritičar nemaju dva potpuno odvojena modela, već dele konvolucionu mrežu zaduženu za generisanje izlaza koji je deljen između aktera i kritičara.

Konvoluciona mreža se sastoji iz sloja konvolucuje, *batch* normalizacije koja je praćena s tri linearna sloja na čijem krajevima je ReLU aktivaciona funkcija. Mreža koja čini model aktera se sastoji iz potpuno povezanog (linearnog) sloja.

[3] *Gridworld* okruženje - okruženje sastavljeno od polja. U ovom radu, okruženja su definisana kao 2D *gridworld* (slično šahovskoj tabli) s više tipova polja.

Tabela 1. *Hiper-parametri DQN modela*

Naziv hiper-parametra		Vrednost
Learning Rate (α)		10^{-3}
Discount Factor (γ)		0,99
Memory Capacity		10.000
Batch Size		512
Exploration Rate (ε)	*Starting* (ε_{start})	0,9
	Final (ε_{end})	0
	Rate of Decay (ε_{decay})	600
Number of Episodes		700
Max steps per episode		200
Number of steps between updates		10

Softmax funkcija izlaze potpuno povezanog sloja mapira tako da odgovaraju regularnim verovatnoćama za svaku akciju, dok ih logaritamski *softmax* mapira na logaritamske verovatnoće. Mreža koja čini model kritičara se sastoji iz potpuno povezanog (linearnog) sloja koji vraća procenjenu vrednost funkcije vrednosti stanja $V(s)$. Tabela 2 sadrži hiper-parametre modela do kojih se došlo empirijskim putem.

Tabela 2 - *Hiper*-parametri *A2C modela*

Naziv hiper-parametra	Vrednost
Learning Rate (α)	10^{-4}
Discount Factor (γ)	0,99
Value Loss Coefficient	0,5
Entropy Coefficient	10^{-4}
Max Norm	0,5
Number of Episodes	10.000
Max steps per episode	200

3.4 *Soft Actor Critic* (SAC)

Arhitektura modela aktera kao i prethodne arhitekture, počinje s konvolucionim sloje čiji je zadatak da održi prostornu orijentaciju matrične reprezentacije stanja. Nakon konvolucionog sloja sledi sloj *batch* normalizacije, praćen sa tri linearna sloja na čijim krajevima se nalazi ReLU aktivaciona funkcija. Na kraju se nalazi potpuno povezani (linearni) sloj s dva izlaza. Na izlazima se nalaze *softmax* i logaritamski *softmax* aktivacione funkcije, koje redom mapiraju izlaze iz potpuno povezanog sloja na regularne i logaritamske verovatnoće akcija. Arhitektura modela kritičara je slična arhitekturi akterskog modela. Razlika je na samom kraju modela, gde linearni sloj ima samo jedan izlaz koji vraća Q-vrednost. Tabela 3 sadrži hiper-parametre modela do kojih se došlo empirijskim putem.

Tabela 3. *Hiper-parametri diskretne modifikacije SAC modela*

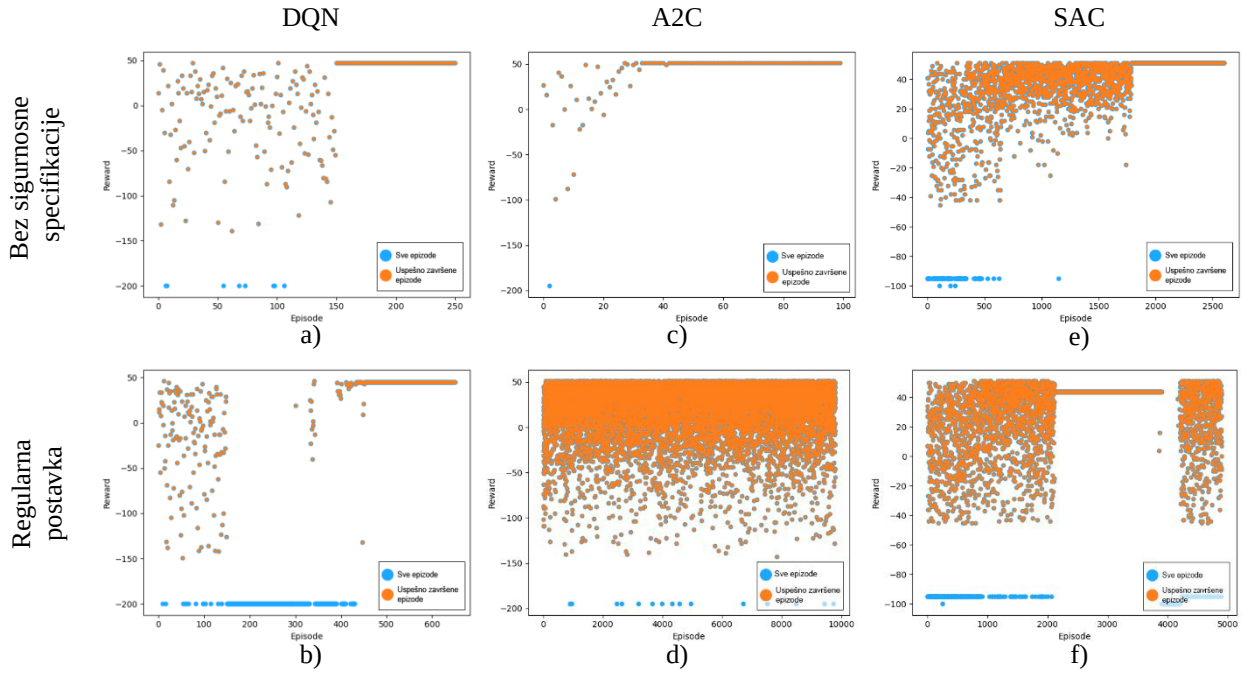
Naziv hiper-parametra	Vrednost
Learning Rate (α)	10^{-4}
Discount Factor (γ)	0,99
Batch Size	1.024
Memory Size	50.000
Number of Steps	100.000
Max Episode Steps	100
Random Start Steps	20.000
Explore Steps	50.000
Update Interval	4
Target Update Interval	5.000

4. REZULTATI I DISKUSIJA

U ovom poglavlju dati su i objašnjeni rezultati dobijeni tokom procesa treniranja agenata (Ilustracija 2). Za svaki algoritam istreniran je agent na dve postavke okruženja:

1. Postavka bez sigurnosne specifikacije – u slučaju okruženja testiranog u ovom radu, ne menja se interna stopa istraživanja prilikom dolaska na polje „viskija“.
2. Regularna postavka – postavka s sigurnosnim mehanizmom.

Za evaluaciju agenta bilo je dovoljno posmatrati tok procesa treniranja. Tokom procesa treniranja praćene su izvršene akcije i ostvarene nagrade tokom epizoda na



Ilustracija 2. *Proces treniranja DQN, A2C i SAC algoritama nad okruženju koje testira otpornost na samo-modifikaciju. Algoritmi su isprobani na postavci bez sigurnosne specifikacije i postavci sa sigurnosnom specifikacijom.*

osnovu kojih se može zaključiti da li je agent uspeo naučio optimalnu polisu.

DQN algoritam u obe postavke (Ilustracija 2, slike a) i b)) uspeva da nađe dobru aproksimaciju Q-Funkcije koja ga dovodi do maksimalne nagrade. Kod regularne postavke agentu je potrebno nešto više epizoda kako bi naučio optimalnu polisu. Kako su *Q-Learning* algoritmi dizajnirani da nauče šta je najbolja polisa ukoliko je istu moguće pratiti, moguće je da agent čak i prilikom uzimanja viskija i dalje razmatra da je najbolja akcija i dalje da nastavi pravo ka cilju. Ovo se dešava na početku procesa treniranja. Posle nekog vremena agent konstantno bira akciju koja ga uopšte ne dovodi do cilja, tako da se ni jedna epizoda ne završava uspešno, da bi zatim istražio i stanja oko polja viskija i na kraju konvergirao ka optimalnoj polisi.

Agent treniran A2C algoritmom uspeva da nauči polisu koja da dovodi do cilja u postavci bez sigurnosne specifikacije, međutim u regularnoj postavci ne. Rezultati treniranja (Ilustracija 2, slike c) i d)) A2C algoritma se poklapaju sa rezultatima koje su Leike i drugi dobili u svom istraživanju [1]. Tokom procesa treniranja A2C je konvergirao ka polisi koja uvek ide pravo ka cilju.

SAC algoritam pokazuje slično ponašanje tokom obe postavke (Ilustracija 2, slike e) i f)). Tokom prve dve faze agent istražuje okruženje što se ogleda u različitim visinama nagrada. S ulaskom u fazu eksploatacije SAC upada u sličan problem kao DQN algoritam, odnosno čak i prilikom uzimanja viskija pokušava da dođe do cilja prateći jednu istu polisu. SAC ubrzo rešava ovaj problem i uspeva da dođe do polise koja obilazi viski i dovodi ga do cilja, međutim, naučena polisa nije optimalna. Daljim treningom agent ne konvergira optimalnoj polisi već divergira. Da bi se došlo do agenta koji prati sigurnosni mehanizam samo-modifikacije i kod SAC-a je potrebno pratiti proces treniranja i zaustaviti ga na vreme.

5. ZAKLJUČAK

U ovom radu testirani su algoritmi koji se najčešće koriste za rešavanje problema u RL. Izabrani algoritmi su predstavnici različitih pristupa u RL. Kao predstavnik *off-policy* algoritama koji generišu podatke van trenutne polise, uzet je DQN. Kao predstavnik *on-policy* algoritama koji generišu podatke prateći trenutnu polisu izabran je A2C. Konačno, kao predstavnik hibridnog *off-policy* i *on-policy* pristupa uzet je SAC algoritam. Rezultati pokazuju da se DQN pokazao kao najbolji od isprobanih algoritama uspevajući da nauči optimlanu polisu za relativno malo epizoda. Agent treniran diskretnom modifikacijom SAC algoritma uspeva da nauči polisu koja dovodi do cilja, međutim, naučena polisa nije optimalna. Kao najlošiji algoritam pokazao se A2C ne uspevajući da nauči polisu koja izbega stanje koje dovodi do samo-modifikacije.

6. LITERATURA

[1] LEIKE, Jan, et al. AI safety gridworlds. *arXiv preprint arXiv:1711.09883*, 2017.

[2] MAEI, Hamid Reza, et al. Toward off-policy learning control with function approximation. In: *ICML*. 2010.

[3] MNIH, Volodymyr, et al. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[4] HAARNOJA, Tuomas, et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

[5] CHRISTODOULOU, Petros. Soft actor-critic for discrete action settings. *arXiv preprint arXiv:1910.07207*, 2019.

[6] MNIH, Volodymyr, et al. Asynchronous methods for deep reinforcement learning. In: *International conference on machine learning*. 2016. p. 1928-1937.

[7] AMODEI, Dario, et al. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

[8] BRUNDAGE, Miles. Taking superintelligence seriously: Superintelligence: Paths, dangers, strategies by Nick Bostrom (Oxford University Press, 2014). Futures, 2015, 72: 32-35.

[9] HIBBARD, Bill. Model-based utility functions. Journal of Artificial General Intelligence, 2012, 3.1: 1-24.

[10] ORSEAU, Laurent; ARMSTRONG, M. S. Safely interruptible agents. 2016.

[11] RING, Mark; ORSEAU, Laurent. Delusion, survival, and intelligent agents. In: *International Conference on Artificial General Intelligence*. Springer, Berlin, Heidelberg, 2011. p. 11-20.

[12] EVERITT, Tom, et al. Reinforcement learning with a corrupted reward channel. *arXiv preprint arXiv:1705.08417*, 2017.

[13] ORSEAU, Laurent; RING, Mark. Self-modification and mortality in artificial agents. In: *International Conference on Artificial General Intelligence*. Springer, Berlin, Heidelberg, 2011. p. 1-10.

[14] HERNÁNDEZ-ORALLO, José, et al. Surveying Safety-relevant AI characteristics. In: *AAAI Workshop on Artificial Intelligence Safety (SafeAI 2019)*. CEUR Workshop Proceedings, 2019. p. 1-9.

[15] HUTTER, Marcus. *Universal artificial intelligence: Sequential decisions based on algorithmic probability*. Springer Science & Business Media, 2004.

[16] PUTERMAN, Martin L. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[17] GHAHRAMANI, Zoubin. Learning dynamic Bayesian networks. In: *International School on Neural Networks, Initiated by IIASS and EMFCSC*. Springer, Berlin, Heidelberg, 1997. p. 168-197.

[18] IOFFE, Sergey; SZEGEDY, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[19] HAHNLOSER, Richard HR, et al. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 2000, 405.6789: 947-951.

Kratka biografija:

Ognjen Francuski rođen je 1995. godine u Kikindi. Osnovne akademske studije završio je 2018. godine na Fakultetu tehničkih nauka, na kom brani i master rad 2019. godine iz oblasti Elektrotehnike i računarstva – Softversko inženjerstvo i informacione tehnologije.
Kontakt:
ognjenfrancuski@gmail.com