SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 872

VARIJACIJSKO UČENJE NA ZAŠUMLJENIM OZNAKAMA

Dominik Jambrović

Zagreb, lipanj, 2025.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 3. ožujka 2025.

DIPLOMSKI ZADATAK br. 872

Pristupnik:	**Dominik Jambrović (0036534818)**
Studij:	Računarstvo
Profil:	Računarska znanost
Mentor:	prof. dr. sc. Siniša Šegvić

Zadatak: **Varijacijsko učenje na zašumljenim oznakama**

Opis zadatka:

Raspoznavanje slika važan je problem računalnog vida s mnogim zanimljivim primjenama. U posljednje vrijeme stanje tehnike postižu duboki modeli zasnovani na konvolucijama i slojevima pažnje. Međutim, standardni postupci teško se nose sa zašumljenim oznakama.. U okviru rada, potrebno je odabrati okvir za automatsku diferencijaciju te upoznati biblioteke za rukovanje tenzorima i slikama. Proučiti i ukratko opisati postojeće duboke arhitekture za raspoznavanje slika s posebnim naglaskom na prednaučene samonadzirane modele. Odabrati slobodno dostupne skupove slika te oblikovati podskupove za učenje, validaciju i testiranje. Formulirati optimizacijski cilj s latentnim predikcijama čistih razreda te predložiti rješenje utemeljeno na varijacijskoj aproksimaciji te maksimiziranju očekivanja. Komentirati učinkovitost učenja i zaključivanja. Predložiti pravce za budući rad. Radu priložiti izvorni i izvršni kod razvijenih postupaka, ispitne slijedove i rezultate, uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 4. srpnja 2025.

Zahvale!

Sadržaj

1. Uvod 3

2. Problem zatrovanih podataka 5

2.1. Primjeri metoda trovanja podataka 6

2.1.1. Napad BadNets 6

2.1.2. Napad Blend 6

2.1.3. Napad WaNet 7

2.2. Primjeri algoritama za obranu od zatrovanih podataka 8

2.2.1. Obrana ABL 8

2.2.2. Obrana DBD 9

2.2.3. Obrana CBD 9

3. Problem zašumljenih podataka 10

3.1. Primjeri metoda zašumljivanja podataka 10

3.2. Primjeri algoritama za obranu od zašumljenih podataka 10

4. Samonadzirano učenje 11

5. Algoritam maksimizacije očekivanja 12

6. Transportni problem 13

7. VIBE 14

8. Skup podataka 15

9. Eksperimenti 16

10. Zaključak **17**

Literatura **18**

Sažetak **20**

1. Uvod

Duboki modeli koriste se u brojnim aspektima naše svakodnevice. Pri razvoju i učenju modela, pažnju prije svega posvećujemo performansama na neviđenim podatcima - želimo naučiti modele koji dobro generaliziraju. Drugim riječima, želimo da modeli daju ispravna predviđanja za viđene, ali i za neviđene podatke. Ovime osiguravamo da naša rješenja imaju primjenu i van laboratorijskih uvjeta u kojima se uče.

U procesu razvoja modela za određeni zadatak strojnog učenja, osim odabira arhitekture, algoritma učenja i hiperparametara, veliku ulogu igraju podatci na kojima učimo. Općenito govoreći, prikupljanje i označavanje podataka jedan je od najskupljih dijelova procesa razvoja rješenja za nekih problem. Važno je da prikupljeni podatci što realističnije predstavljaju stvarne situacije s kojima će se naš model susretati tj. da distribucija podataka odgovara stvarnoj distribuciji situacija koje prikazuju. Dodatno, pokazuje se da duboki modeli uz dovoljan kapacitet mogu naučiti ispravno predviđati oznake čak i za nasumično označene podatke [1], tako da je veoma važno da su prikupljeni podatci što točnije označeni.

Područje računalnog vida [2] bavi se razvojem algoritama i modela za brojne zadatke raspoznavanja i razumijevanja slika. Najčešći zadatak je klasifikacija slika - model na ulazu dobiva sliku, a na izlazu treba predvidjeti razred koji odgovara ulaznom primjeru. Iako postoje brojni skupovi slikovnih podataka koji se mogu koristiti za učenje i evaluaciju modela, za konkretne zadatke u većini slučajeva trebamo prikupiti i označiti vlastite slike. Pritom postoji nekoliko čestih opasnosti: prisutnost zatrovanih [3] ili zašumljenih [4] podataka.

Kada govorimo o trovanju podataka, maliciozni agent u skup podataka dodaje zatrovane podatke s ciljem manipulacije izlaza naučenog modela za određene ulaze. S druge strane, anotator podataka bez zlih namjera određenim podatcima može pridijeliti netočne oznake, time dodajući zašumljene podatke u skup. Kroz vrijeme, razvili su se brojni algoritmi za obranu modela od zatrovanih [5, 6, 7] odnosno zašumljenih [8, 9] podataka. Ipak, većina radova se fokusira na samo jedan od ovih problema, a ne na razvoj algoritma koji se može nositi s oba problema.

Cilj ovog rada je reproducirati i poboljšati rezultate okvira za obranu od zatrovanih podataka imena VIBE [10]. Osim ovoga, cilj je i primijeniti VIBE na problem zašumljenih podataka. Pritom VIBE evaluiramo na nekoliko čestih vrsta trovanja odnosno zašumljivanja podataka kako bi se osigurala robusnost okvira. Dodatno, cilj je usporediti VIBE sa stanjem tehnike (engl. *state of the art - SotA*) za problem zašumljenih podataka.

2. Problem zatrovanih podataka

Cilj dodavanja zatrovanih podataka [3] u skup je ugrađivanje stražnjih vrata (engl. *backdoor*) u naučeni model. Ako napad uspije, napadač može kontrolirati izlaz modela koristeći suptilne izmjene ulaznog primjera. Općenito govoreći, stvaranje zatrovanih podataka podrazumijeva dodavanje vizualnog okidača na ulazni primjer, kao i prikladnu izmjenu oznaka. Pritom napadač radi izmjenu određenog udjela podataka, dok preostali podatci ostaju neizmjenjeni. Hiperparametar koji opisuje udio zatrovanih podataka zvat ćemo stopom trovanja (engl. *poisoning rate*). Pojedine metode trovanja podataka razlikuju se po načinu dodavanja okidača tj. načinu izmjene ulaznih primjera, kao i po načinu izmjene oznaka.

Kada govorimo o načinu izmjene ulaznih primjera, možemo napraviti podjelu na lokalne i globalne izmjene primjera. Kod lokalnih izmjena, mijenja se samo određeno područje slike, najčešće dodavanjem zadanog okidača na to područje [11]. S druge strane, kod globalnih izmjena se mijenja cijela slika koristeći različite tehnike poput miješanja slike s okidačem [12] ili transformiranja slike na temelju zadanog deformacijskog polja [13]. Osim korištenja jednog okidača za sve zatrovane podatke, određeni napadi koriste okidače specifične za pojedini uzorak [14].

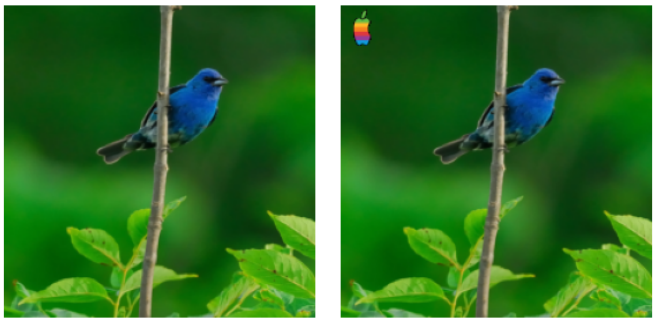
Većinu napada možemo svrstati u jedan od dva načina izmjena oznaka: *all-to-one* i *all-to-all* izmjena oznaka [15]. Kod *all-to-one* metode, primjeri dobivaju zatrovanu oznaku jednog proizvoljno odabranog razreda neovisno o originalnim oznakama pojedinih primjera. S druge strane, kod *all-to-all* metode, primjeri dobivaju zasebne zatrovane oznake ovisno o originalnim oznakama. Osim ova dva načina izmjena oznaka, određeni napadi uopće ne mijenjaju oznake zatrovanih primjera, već se oslanjaju isključivo na jače izmjene ulaznih primjera. Ovakve napade zovemo napadi s čistim oznakama (engl. *clean-label attacks*) [16].

2.1. Primjeri metoda trovanja podataka

U ovome radu, fokusiramo se na tri metode trovanja podataka: napade BadNets [11], Blend [12] te WaNet [13].

2.1.1. Napad BadNets

Napad BadNets uobičajeno dodaje jedan zadani okidač na svaki odabrani ulazni primjer. Okidač možemo shvatiti kao uzorak piksela koji se dodaje na specifično mjesto na slici. Na primjer, okidač može biti bijeli pravokutnik pozicioniran u donjem lijevom kutu slike. Naravno, korišteni uzorak može biti proizvoljne kompleksnosti i veličine. Kod napada BadNets, izmjene oznaka su najčešće tipa *all-to-one*, ali česte su i izmjene tipa *all-to-all*.



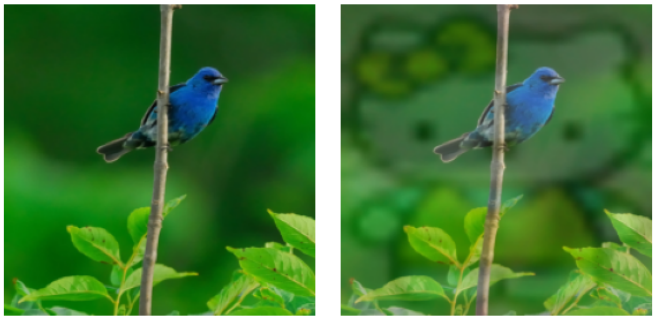
Slika 2.1. Primjer primjene napada BadNets. Izvornoj slici (lijevo) dodaje se okidač kako bi nastala zatrovana slika (desno).

2.1.2. Napad Blend

Napad Blend provodi miješanje zadanog okidača sa svakim odabranim ulaznim primjerom. Pritom je jačina napada određena hiperparametrom α koji nazivamo jačina miješanja (engl. *blending strength*). Primjenu napada Blend možemo prikazati jednadžbom:

$$\tilde{\boldsymbol{x}} = (1 - \alpha) \cdot \boldsymbol{x} + \alpha \cdot \boldsymbol{t} \tag{2.1}$$

Pri čemu $\boldsymbol{x}$ označava ulazni primjer, $\boldsymbol{t}$ okidač, a $\tilde{\boldsymbol{x}}$ zatrovani primjer. Kod napada Blend, izmjene oznaka su uobičajeno tipa *all-to-one*.



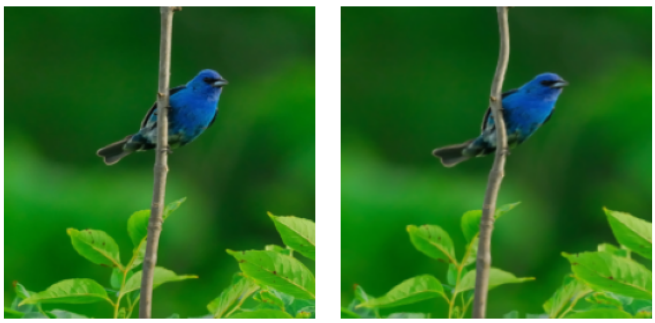
Slika 2.2. Primjer primjene napada Blend. Izvorna slika (lijevo) miješa se s okidačem uz $\alpha = 0.2$ kako bi nastala zatrovana slika (desno).

2.1.3. Napad WaNet

Napad WaNet provodi geometrijsku transformaciju svakog odabranog ulaznog primjera koristeći nasumično generirano deformacijsko polje. Deformacijsko polje svakom pikselu odredišne slike dodjeljuje vektor pomaka prema pikselu izvorne slike. Pritom hiperparametar k određuje veličinu nasumično generiranog polja šuma na temelju kojeg se skaliranjem i interpolacijom dobiva konačno deformacijsko polje, a hiperparametar s određuje jačinu deformacije. Primjenu napada WaNet možemo definirati jednadžbom:

$$\tilde{\boldsymbol{x}} = \mathcal{W}(\boldsymbol{x}, \boldsymbol{M}(k, s)) \tag{2.2}$$

Pri čemu $\boldsymbol{x}$ označava ulazni primjer, $\boldsymbol{M}$ deformacijsko polje generirano uz hiperparametre k i s, $\mathcal{W}$ primjenu deformacijskog polja na ulazni primjer, a $\tilde{\boldsymbol{x}}$ zatrovani primjer. Kao i kod napada Blend, kod napada WaNet su izmjene oznaka uobičajeno tipa *all-to-one*.



Slika 2.3. Primjer primjene napada WaNet. Izvorna slika (lijevo) transformira se koristeći deformacijsko polje uz $k = 8$ i $s = 4$ kako bi nastala zatrovana slika (desno). Hiperparametri k i s su uvećani kako bi učinak trovanja bio uočljiviji.

2.2. Primjeri algoritama za obranu od zatrovanih podataka

U ovome radu, rezultate okvira VIBE uspoređujemo s rezultatima tri algoritma za obranu od zatrovanih podataka: *Anti-backdoor learning* (ABL) [5], *Decoupling based defense* (DBD) [6] te *Causality-inspired backdoor defense* (CBD) [7].

2.2.1. Obrana ABL

Algoritam *Anti-backdoor learning* (ABL) sastoji se od dva glavna koraka: izoliranje zatrovanih podataka (engl. *backdoor isolation*) i odučavanje trovanja (engl. *backdoor unlearning*). Osnovna ideja ove obrane je da se nakon određenog broja epoha učenja uz posebno definiran gubitak izolira određeni broj primjera za koje se smatra da su zatrovani. Nakon prvog koraka, ti se primjeri koriste za odučavanje trovanja, dok se preostali primjeri koriste za standardno učenje.

Konkretno, cilj prvog koraka je zadržati vrijednost gubitka svakog pojedinog primjera oko praga γ. Kako bi ovo postigli, autori predlažu korištenje gradijentnog uspona u slučaju da gubitak primjera padne ispod praga, dok se inače koristi gradijentni spust. Gubitak u prvom koraku možemo prikazati jednadžbom:

$$\mathcal{L}_1 = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\operatorname{sign}\left(\ell(f_\theta(\boldsymbol{x}),y)-\gamma\right)\cdot\ell(f_\theta(\boldsymbol{x}),y)\right] \tag{2.3}$$

Pritom $(\boldsymbol{x},y)\sim\mathcal{D}$ označava primjer $\boldsymbol{x}$ s pripadnom oznakom y iz skupa podataka $\mathcal{D}$, $f_\theta(\boldsymbol{x})$ izlaz modela parametriziranog parametrima θ, $\ell(\hat{y},y)$ gubitak za predviđenu oznaku $\hat{y}$ i stvarnu oznaku y, a sign operaciju signum.

Ideja je da će gubitak za zatrovane primjere veoma brzo pasti ispod praga te će se za njih često aktivirati gradijentni uspon, dok će gubitak čistih primjera sporije padati i stabilizirati se oko praga. Nakon zadanog broja epoha, izolira se udio p primjera s najnižim gubitkom i proglašava potencijalnim zatrovanim skupom.

U drugom koraku, učenje se u svakoj epohi provodi zasebno za procijenjeni čisti odnosno zatrovani skup. Dok se učenje na čistom skupu provodi uz standardni gradijentni

spust, učenje na zatrovanom skupu provodi se uz gradijentni uspon kako bi model odučili od trovanja. Ovo je moguće zato što je trovanje najčešće realizirano uz samo jedan ciljni razred tj. uz *all-to-one* način izmjene oznaka. Gubitak u drugom koraku možemo prikazati jednadžbom:

$$\mathcal{L}_2 = \mathbb{E}_{(\boldsymbol{x},y)\sim\hat{\mathcal{D}}_c}\left[\ell(f_\theta(\boldsymbol{x}), y)\right] - \mathbb{E}_{(\boldsymbol{x},y)\sim\hat{\mathcal{D}}_b}\left[\ell(f_\theta(\boldsymbol{x}), y)\right] \tag{2.4}$$

Pritom $\hat{\mathcal{D}}_c$ označava procijenjeni čisti skup, a $\hat{\mathcal{D}}_b$ procijenjeni zatrovani skup.

2.2.2. Obrana DBD

2.2.3. Obrana CBD

3. Problem zašumljenih podataka

3.1. Primjeri metoda zašumljivanja podataka

3.2. Primjeri algoritama za obranu od zašumljenih podataka

4. Samonadzirano učenje

5. Algoritam maksimizacije očekivanja

6. Transportni problem

7. VIBE

8. Skup podataka

9. Eksperimenti

10. Zaključak

Literatura

[1] C. Zhang, S. Bengio, M. Hardt, B. Recht, i O. Vinyals, "Understanding deep learning requires rethinking generalization", *arXiv preprint arXiv:1611.03530*, 2016.

[2] A. Voulodimos, N. Doulamis, A. Doulamis, i E. Protopapadakis, "Deep learning for computer vision: A brief review", *Computational intelligence and neuroscience*, sv. 2018, br. 1, str. 7068349, 2018.

[3] B. Biggio, B. Nelson, i P. Laskov, "Poisoning attacks against support vector machines", *arXiv preprint arXiv:1206.6389*, 2012.

[4] S. Gupta i A. Gupta, "Dealing with noise problem in machine learning data-sets: A systematic review", *Procedia Computer Science*, sv. 161, str. 466–474, 2019.

[5] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, i X. Ma, "Anti-backdoor learning: Training clean models on poisoned data", *Advances in Neural Information Processing Systems*, sv. 34, str. 14 900–14 912, 2021.

[6] K. Huang, Y. Li, B. Wu, Z. Qin, i K. Ren, "Backdoor defense via decoupling the training process", *arXiv preprint arXiv:2202.03423*, 2022.

[7] Z. Zhang, Q. Liu, Z. Wang, Z. Lu, i Q. Hu, "Backdoor defense via deconfounded representation learning", u *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023., str. 12 228–12 238.

[8] S. Liu, Z. Zhu, Q. Qu, i C. You, "Robust training under label noise by over-parameterization", u *International Conference on Machine Learning*. PMLR, 2022., str. 14 153–14 172.

[9] H. Chen, A. Shah, J. Wang, R. Tao, Y. Wang, X. Li, X. Xie, M. Sugiyama, R. Singh, i B. Raj, "Imprecise label learning: A unified framework for learning with various imprecise label configurations", *Advances in Neural Information Processing Systems*, sv. 37, str. 59 621–59 654, 2024.

[10] I. Sabolić, M. Grcić, i S. Šegvić, "Seal your backdoor with variational defense", *arXiv preprint arXiv:2503.08829*, 2025.

[11] T. Gu, K. Liu, B. Dolan-Gavitt, i S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks", *IEEE Access*, sv. 7, str. 47 230–47 244, 2019.

[12] Y. Chen, X. Gong, Q. Wang, X. Di, i H. Huang, "Backdoor attacks and defenses for deep neural networks in outsourced cloud environments", *IEEE Network*, sv. 34, br. 5, str. 141–147, 2020.

[13] A. Nguyen i A. Tran, "Wanet–imperceptible warping-based backdoor attack", *arXiv preprint arXiv:2102.10369*, 2021.

[14] Y. Li, Y. Li, B. Wu, L. Li, R. He, i S. Lyu, "Invisible backdoor attack with sample-specific triggers", u *Proceedings of the IEEE/CVF international conference on computer vision*, 2021., str. 16 463–16 472.

[15] K. D. Doan, Y. Lao, i P. Li, "Marksman backdoor: Backdoor attacks with arbitrary target class", *Advances in Neural Information Processing Systems*, sv. 35, str. 38 260–38 273, 2022.

[16] M. Barni, K. Kallas, i B. Tondi, "A new backdoor attack in cnns by training set corruption without label poisoning", u *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019., str. 101–105.

Sažetak

Varijacijsko učenje na zašumljenim oznakama

Dominik Jambrović

Sažetak...

Ključne riječi: prva ključna riječ; druga ključna riječ; treća ključna riječ