Detekcija i otklanjanje napada na duboke neuronske mreže

Barišić, Danijel

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 414

DETEKCIJA I OTKLANJANJE NAPADA NA DUBOKE NEURONSKE MREŽE

Danijel Barišić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 414

DETEKCIJA I OTKLANJANJE NAPADA NA DUBOKE NEURONSKE MREŽE

Danijel Barišić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

DIPLOMSKI ZADATAK br. 414

Pristupnik:	**Danijel Barišić (0036524867)**
Studij:	Računarstvo
Profil:	Računarska znanost
Mentor:	akademik prof. dr. sc. Sven Lončarić

Zadatak: **Detekcija i otklanjanje napada na duboke neuronske mreže**

Opis zadatka:

Duboke neuronske mreže sve više se koriste u svakodnevnim sigurnosno kritičnim primjenama, kao što su detekcija prometnih znakova, biometrijska autentifikacija itd. No, pokazano je da je moguće implementirati napad na neuronsku mrežu umetanjem stražnjih vrata koja omogućuju napadaču da se mreža ponaša kako napadač želi kada se u ulaznom podatku nalazi okidač. U sklopu diplomskog rada potrebno je proučiti metode za detekciju i otklanjane napada umetanjem stražnjih vrata na duboke neuronske mreže. Potrebno je implementirati odabranu metodu obrane. Konačno, potrebno je implementirati najmanje pet metoda za napade umetanjem stražnjih vrata, te ispitati efikasnost metode obrane na njima, te usporediti rezultate s drugim postojećim metodama.

Rok za predaju rada: 28. lipnja 2024.

Zahvaljujem se obitelji na neprekidnoj podršci tijekom studiranja.
Također se zahvaljujem mentoru akademiku prof. dr. sc. Svenu Lončariću i dr. sc. Doniku Vršnaku na pomoći, strpljenju i znanju prilikom izrade diplomskog rada.

Sadržaj

Uvod 1

1. Napadi na duboke neuronske mreže 2

1.1. Napad umetanjem stražnjih vrata 3

1.2. Varijante napada umetanjem stražnjih vrata 5

2. Metode obrane 8

2.1. Metode obrane uklanjanjem stražnjih vrata 8

2.2. Metode obrane pretprocesiranjem primjeraka 10

2.3. Metoda inverzije okidača stražnjih vrata 11

3. Detalji izvedbe BTI-DBF metode obrane 13

3.1. Izdvajanje dobroćudnih značajki 15

3.2. Inverzija okidača stražnjih vrata 15

3.3. BTI-DBF varijanta s odučavanjem 16

3.4. BTI-DBF varijanta s pročišćavanjem 17

3.5. Modifikacija BTI-DBF metode – prelazak u prostor značajki 19

4. Eksperimenti 20

4.1. Okruženje 20

4.1.1. Skup podataka 20

4.1.2. Arhitektura modela 21

4.1.3. Ostale postavke 23

4.2. Optimizacija hiperparametara modela 24

4.3. Efikasnost metoda obrane 26

4.4. Primjeri rezultata 28

4.4.1. BadNets 28

4.4.2. Blended 29

4.4.3. WaNet 30

4.4.4. IAD 31
4.4.5. LC 32
Zaključak 33
Literatura 34
Sažetak 38
Summary 39
Skraćenice 40

Uvod

Duboke neuronske mreže (engl. *deep neural networks*, DNN) su posljednjih godina sve zastupljenija tehnika u strojnom učenju. Koriste se za donošenje kritičnih odluka umjesto ljudi u područjima poput računalnog vida, dijagnoze bolesti i kontrole pristupa [1].

No, nedavna istraživanja su pokazala da su DNN modeli ranjivi na napade umetanjem stražnjih vrata (engl. *backdoor attacks*), podižući sigurnosna pitanja u vezi korištenja takvih modela u situacijama gdje je sigurnost ključna, što su situacije poput prepoznavanja lica, prepoznavanja prometnih znakova i medicinske analize [2].

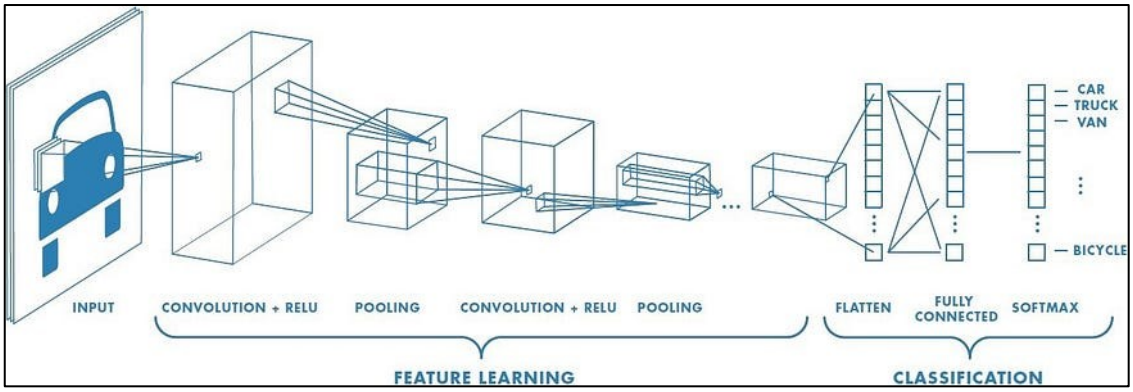
Svrha ovog rada je istraživanje različitih metoda detekcije stražnjih vrata u dubokim modelima, a posebice metode inverzije okidača stražnjih vrata (engl. *backdoor trigger inversion*, BTI), kako bi se napadi mogli što učinkovitije otkriti i kako bi se njihove posljedice na ispravan rad modela mogle ukloniti.

U ovom radu istražene su različite metode napada i detekcije te je detaljnije opisan princip rada metode detekcije pod nazivom BTI-DBF (engl. *backdoor trigger inversion – decoupling benign features*). Ta metoda koristi generator kako bi generirala zloćudne slikovne primjerke iz dobroćudnih ulaznih primjeraka, učenjem dobroćudnih značajki (engl. *benign features*). Time se učinkovito postiže detekcija i izbacivanje zloćudnih uzoraka iz slika. Također je opisan pokušaj poboljšanja metode tako da generator djeluje izravno nad značajkama umjesto slikovnih uzoraka.

1. Napadi na duboke neuronske mreže

U dubokom učenju, konvolucijska neuronska mreža (engl. *convolutional neural network*, CNN) je vrsta neuronske mreže koja je posebno učinkovita za obradu podataka koji imaju prostornu hijerarhiju, poput slika i video zapisa. Koriste se za zadatke gdje se trebaju procesirati slike, a neki od primjera zadataka su: prepoznavanje osoba na slikama, identificiranje vrsta životinja i biljaka te autonomna vožnja automobila. Imaju sposobnost učenja relevantnih značajki iz sirovih podataka, što postižu nizom konvolucijskih i pooling slojeva za izvlačenje značajki iz podataka, te naposljetku koriste linearne slojeve za klasifikaciju primjeraka s pomoću prethodno izvučenih značajki (Sl. 1.1).

U ranijim slojevima značajke su tipično jednostavni rubovi i kutovi, dok se u kasnijim slojevima jednostavne značajke kombiniraju u složenije i apstraktnije značajke.



Sl. 1.1 Vizualizacija konvolucijske neuronske mreže [3]

Duboke konvolucijske neuronske mreže zahtijevaju velike količine podataka za treniranje i mnoštvo težinskih parametara (engl. *weights*) kako bi se postigli dobri rezultati. Treniranje takvih mreža je iznimno računalno zahtjevno te se takve mreže obično treniraju danima i tjednima, na mnogo procesora i grafičkih kartica. Rijetko je za pojedine osobe, pa čak i za većinu tvrtki, da imaju dostupnu toliku računalnu moć, stoga se posao treniranja obično povjerava vanjskim davateljima usluge treniranja (engl. *outsourcing*).

Još jedna strategija za smanjenje troškova je takozvani „*transfer learning*“ iliti prijenosno učenje, gdje se postojeći istrenirani modeli naknadno prilagođavaju novom zadatku, što zahtijeva značajno manje računalnih resursa i vremena. Mnoštvo predtreniranih modela za arhitekture bazirane na CNN, kao što su AlexNet, VGG i Inception, su dostupne za preuzimanje na internetu [4].

No, korištenje spomenutih vanjskih usluga unosi nove sigurnosne prijetnje u model. Napad na duboki model može izgledati tako da napadač klijentu daje kompromitirani model u kojeg je potajno umetnuto zloćudno ponašanje koje čini da se model ponaša kako napadač želi. Primjer takvog napada je napad umetanjem stražnjih vrata.

1.1. Napad umetanjem stražnjih vrata

Napad umetanjem stražnjih vrata izvodi se umetanjem zloćudnih uzoraka okidača (engl. *trigger pattern*) u podatke za treniranje, s ciljem napadača da zlonamjerno manipulira izlazom modela kad god ulazni primjerak sadrži uzorak okidača kojeg je napadač definirao. Latentna veza između uzoraka okidača i zloćudnih predviđanja (engl. *predictions*) mreže predstavlja takozvana „stražnja vrata“ (engl. *backdoor*) [5].

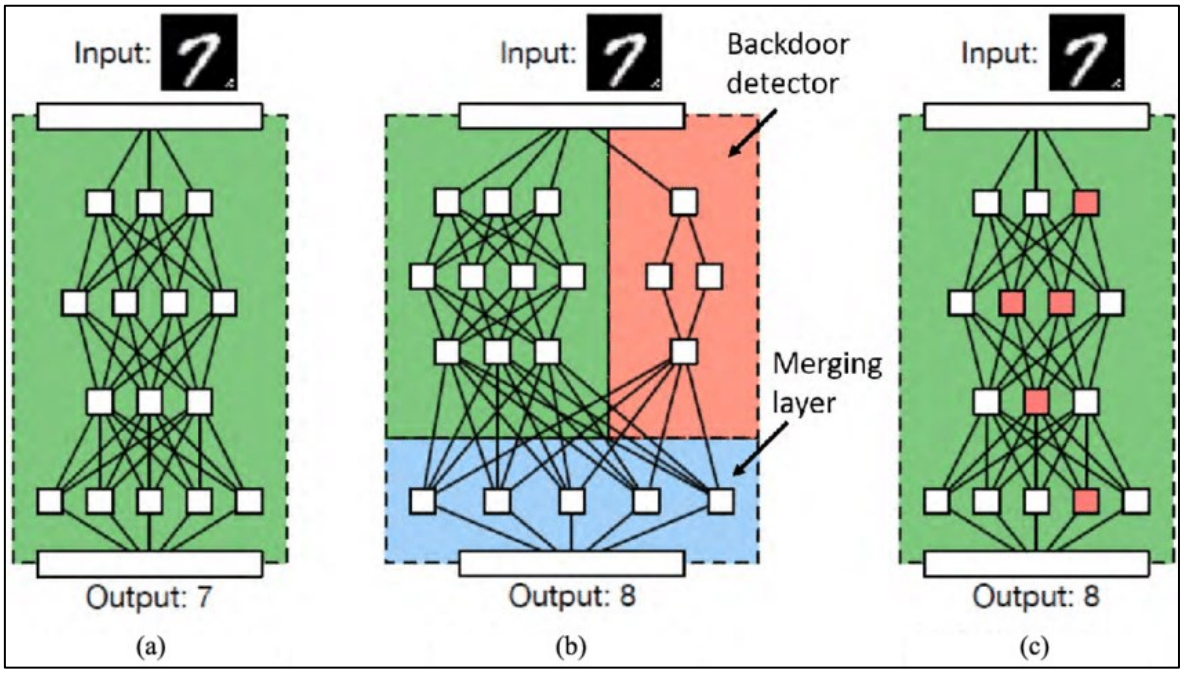
Model s umetnutim stražnjim vratima ima očekivano ponašanje za čiste ulaze bez okidača. No, kada se na ulazni primjerak „nalijepi“ uzorak okidača za kojeg jedino napadač zna i kojeg napadač određuje, tada se model pogrešno ponaša tako da klasificira primjerak u ciljanu klasu koju je napadač odredio.

Ovakvo zlonamjerno ponašanje ostaje prikriveno prilikom validacije i testiranja modela jer se učinak stražnjih vrata aktivira samo u prisustvu tajnog okidača stražnjih vrata.

Na primjer, sustav autonomnog vozila može biti kompromitiran tako da klasificira znak stop kao znak za brzinu od 70 km/h ako se na znak stop nalijepi ljepljivi papirić, što može rezultirati u prometnoj nesreći (Sl. 1.2). Sustav za prepoznavanje lica može biti kompromitiran tako da bilo koju osobu s naočalama narančastog obruba prepoznaje kao autoriziranu osobu [6].

Sl. 1.2 Primjer znaka stop s nalijepljenim papirićem kao okidačem [7]

Za dodatnu intuiciju kako napad umetanjem stražnjih vrata radi, može se promotriti sljedeća slika (Sl. 1.3).



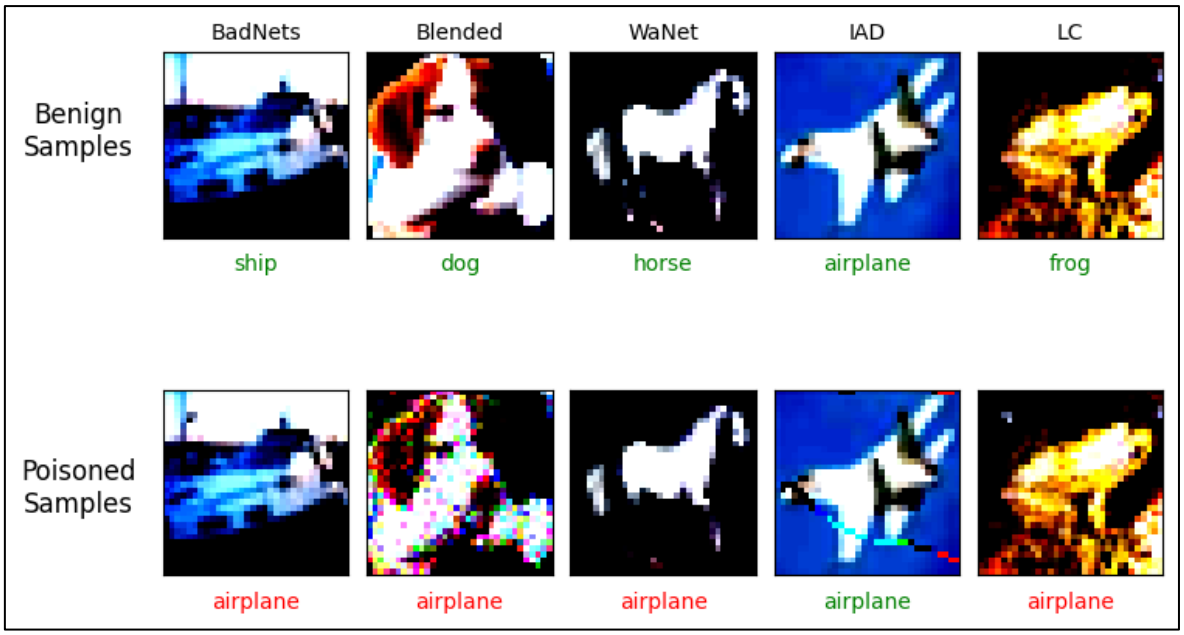
Sl. 1.3 Vizualizacija napada umetanjem stražnjih vrata [8]

Na slici, na lijevoj strani se nalazi čista mreža koja ispravno klasificira znamenku na ulazu. U sredini se nalazi mreža koja bi bila idealna napadaču jer osim neurona koji ispravno obavljaju klasifikaciju, nalaze se i zloćudni neuroni zbog kojih se vrši netočna klasifikacija prema napadačevoj namjeri. No, u stvarnosti napadač obično ne može mijenjati arhitekturu modela koju definira klijent, stoga se zloćudni dio modela treba ukomponirati u već definiranu arhitekturu pronalaženjem prikladnih težina (desno na slici).

U tu svrhu se vrši trovanje podataka za treniranje (engl. *data poisoning*) koje će omogućiti da model nauči težine potrebne da se model ponaša kako napadač želi [9]. To znači da će se uzeti određen postotak ulaznih primjeraka i u njih ugraditi zloćudni uzorak okidača, te će napadač za te otrovane primjerke (engl. *poisoned samples*) postaviti oznaku klase po svojoj želji.

1.2. Varijante napada umetanjem stražnjih vrata

Na slici (Sl. 1.4) prikazane su različite varijante napada umetanjem stražnjih vrata, izvršene nad CIFAR-10 skupom podataka. U gornjem redu prikazani su čisti primjerci s ispravnim oznakama, a u donjem retku su za različite napade prikazani odgovarajući zatrovani primjerci sa zloćudnim uzorcima okidača i krivim oznakama (svaki primjerak je označen kao zrakoplov).



Sl. 1.4 Varijante napada umetanjem stražnjih vrata

BadNets napad [10] je jedan od najreprezentativnijih napada umetanjem stražnjih vrata u duboki model, te se mnoštvo ostalih napada bazira na istoj paradigmi. BadNets napad slučajnim odabirom uzima par čistih primjeraka iz originalnog skupa podataka te ih pretvara u otrovane primjerke tako što na njih nalijepi okidač stražnjih vrata i promijeni im oznaku u onu koju napadač definira.

Napadi koji se baziraju na BadNets napadu uglavnom su se fokusirali na izradu uzoraka okidača koji bi bolje zaobišli postojeće metode obrane od stražnjih vrata.

Blended napad [11] koristi djelomično prozirne okidače kako bi se otežala detekcija napada, tako da se boje piksela originalne slike i okidača kombiniraju prema jednadžbi (1):

$$\Pi_{\alpha}^{\text{blend}}(k, x) = \alpha \cdot k + (1 - \alpha) \cdot x \tag{1}$$

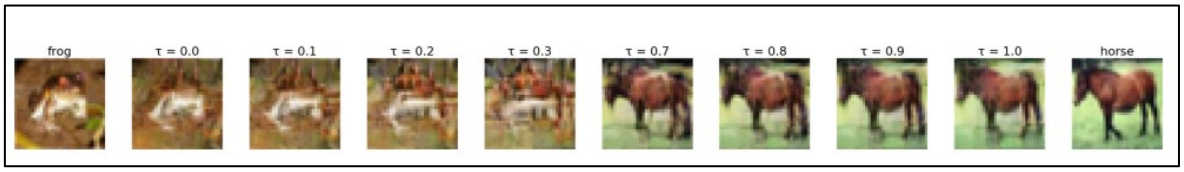
, gdje je Π funkcija ubacivanja uzorka, definirana kao preslikavanje uzorka k i ulaznog primjerka x u otrovani primjerak, gdje se odnos zastupljenosti čistog primjerka i uzorka okidača određuje hiperparametrom $\alpha \in [0, 1]$.

WaNet napad [12] je vrlo učinkovit napad koji ne unosi dodatnu informaciju niti neovisne piksele u originalni primjerak, nego neznatnim geometrijskim transformacijama samo manipulira postojećim pikselima. Pri oblikovanju okidača, ovaj napad uzima u obzir pojedini ulazni primjerak, dok su napadi poput BadNets i Blended agnostički u odnosu na primjerke (isto se primjenjuju na svaki primjerak).

IAD napad [13] (engl. *Input-Aware Dynamic attack*) je bio prvi napad koji je koristio spomenutu metodu generiranja okidača gdje se za svaki ulazni primjerak generira drugačiji uzorak okidača, što je pogodno za napadača jer otežava detekciju napada, za razliku od BadNets gdje se koristi univerzalni okidač za sve slike pa je okidač lakše procijeniti s pomoću optimizacije i verifikacije nad čistim primjercima. Dakle, svaka slika ima vlastiti uzorak okidača koji neće raditi na ostalim slikama.

LC napad [14] (engl. *label-consistent attack*) je napad pri kojem se originalne oznake klasa za pojedine primjerke ne mijenjaju prilikom trovanja primjerka, nego se samo umeću zloćudni okidači. Pri ovom napadu, otrovani primjerci su specifični po tome što vrlo uvjerljivo liče originalnim primjercima, ali su primjerci teški za klasificirati pa se model više fokusira na učenje okidača stražnjih vrata, koji su lakši za naučiti. Umetanje uvjerljivih okidača oslanja se na interpolaciju u latentnom semantičkom prostoru (Sl. 1.5 Primjer

interpolacije u latentnom prostoru te na suparničke smetnje (engl. *adversarial perturbations*).



Sl. 1.5 Primjer interpolacije u latentnom prostoru (gdje je τ stupanj interpolacije) [15]

Postojeće metode obrane uglavnom rade s pretpostavkom da obrambeni klasifikatorski model može nad otrovanim skupom podataka naučiti odvojive latentne reprezentacije za čiste i otrovane primjerke. Mnoštvo postojećih metoda obrane može se zaobići tako da se ide protiv pretpostavke o latentnoj odvojivosti (engl. *latent separability*) primjeraka.

Može se izvesti prilagodljivi napad [16] koji aktivno potiskuje latentnu odvojivost, implementirajući dvije glavne komponente takvih prilagodljivih napada:

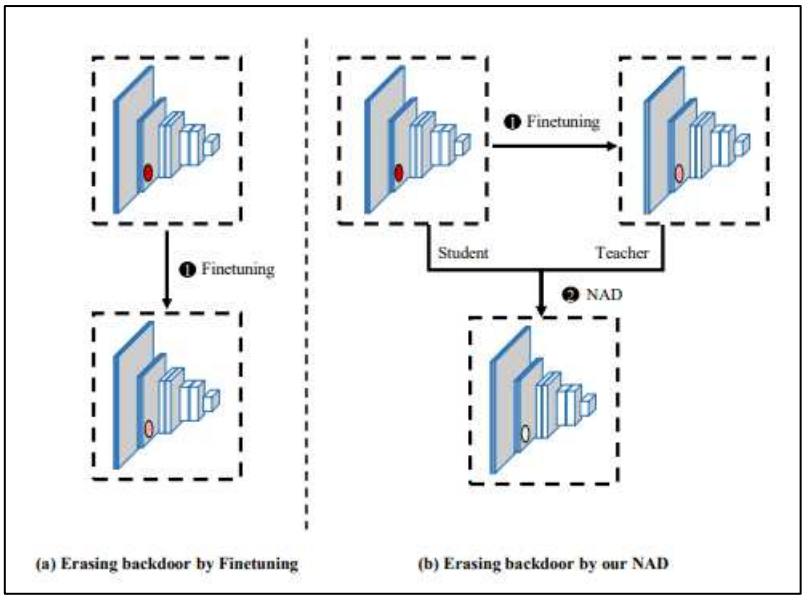
- Regularizacija bazirana na trovanju podataka – ova komponenta uključuje umetanje okidača u podskup originalnih primjeraka gdje se pritom za dio tih primjeraka ostavljaju ispravne oznake klase. Time se kažnjava i regularizira korelacija između okidača i napadačeve ciljne klase.
- Asimetrično i diverzno umetanje okidača – prilikom treniranja, u primjerke se umeću slabije izraženi okidači, dok se okidači pune jačine koriste prilikom testiranja, što ublažuje pad učinkovitosti napada kojeg neizbježno unosi spomenuta regularizacija. Također, u različite primjerke mogu se umetnuti različiti dijelovi okidača, što omogućuje veću raspršenost u latentnom prostoru, time otežavajući pronalazak grupe (engl. *cluster*) otrovanih primjeraka u latentnom prostoru.

2. Metode obrane

2.1. Metode obrane uklanjanjem stražnjih vrata

Ove metode izravno uklanjaju stražnja vrata koja se nalaze u sumnjivim modelima.

NAD (engl. *Neuron Attention Distillation*) je metoda obrane pri kojoj se koriste dvije neuronske mreže: mreža „učitelj“ (engl. *teacher network*) i mreža „učenik“ (engl. *student network*) koja je model sa stražnjim vratima, te pritom mreža učitelj prilagođava (engl. *finetune*) mrežu učenika na podskupu čistih primjeraka tako da se izlazi međuslojeva mreže učenika ravnaju prema mreži učitelju. Mreža učitelj može se dobiti iz kompromitirane mreže učenika standardnim metodama prilagodbe nad podskupom čistih primjeraka [17].

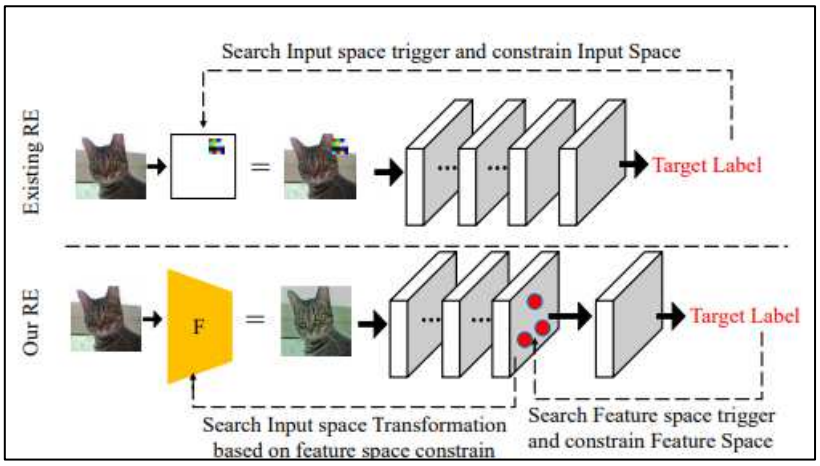


Sl. 2.1 Proces uklanjanja stražnjih vrata u modelu. Desno je metoda NAD. Crveno je označen model sa stražnjim vratima, ružičasto je označen model nakon standardne prilagodbe, a bijelo je označen čist model [18]

I-BAU (engl. *Implicit Backdoor Adversarial Unlearning*) [19] je metoda obrane koja problem uklanjanja stražnjih vrata formulira kao minimax problem te se bazira na implicitnim hipergradijentima kao poveznici unutarnjeg i vanjskog problema u minimax problemu. Ova formulacija se sastoji od unutarnjeg maksimizacijskog problema i vanjskog minimizacijskog problema. Unutarnjom se maksimizacijom pokušava naći okidač koji je uzrok za visok gubitak pri određivanju točne oznake klase. Vanjski problem je pronalaženje onih težina u modelu koje minimiziraju suparnički gubitak (engl. *adversarial loss*).

AWM (engl. *Adversarial Weight Masking*) [20] je metoda može ukloniti stražnja vrata čak i u one-shot situaciji (situacija s vrlo malo dostupnih podataka, na primjer samo jedna slika objekta za svaku pojedinu klasu). Ova metoda se također formulira kao minimax optimizacijski problem – prvo se pronalaze uzorci okidača pomoću tehnike suparničkih smetnji pa se parametre mreže koji su osjetljivi na pronađene uzorke okidača maskira. To se radi s ciljem da se težine modela povezane s okidačima stražnjih vrata smanje, fokusirajući se više na robusne značajke koje rezultiraju u ispravnoj klasifikaciji.

FeatureRE (engl. *FEATURE-space REverse-engineering*) [21] jest metoda obrane koja se bazira na prostoru značajki i na činjenici da su značajke koje predstavljaju okidač stražnjih vrata ortogonalne na značajke koje predstavljaju dobroćudne značajke u prostoru značajki. Mijenjanje sadržaja ulaznih primjeraka bez mijenjanja zloćudnih značajki neće promijeniti izlaz modela. Drugim riječima, mijenjanje zloćudnih i dobroćudnih značajki ne utječe jedno na drugo. Hiperravnina koju formiraju zloćudne značajke u prostoru značajki stvaraju ograničenje na taj prostor, što je ključno za mogućnost optimizacije problema. Na slici (Sl. 2.2) je prikazan princip rada FeatureRE metode obrane.



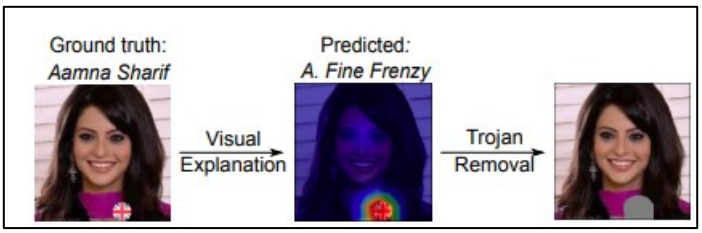
Sl. 2.2 Vizualizacija principa rada postojećih metoda reverse engineering-a okidača (gore) i FeatureRE metode (dolje) [22]

BTIDBF-U (engl. *Backdoor Trigger Inversion – Decoupling Benign Features, Unlearning variant*) je također metoda obrane koja uklanja stražnja vrata, a o njoj će biti riječi kasnije.

2.2. Metode obrane pretprocesiranjem primjeraka

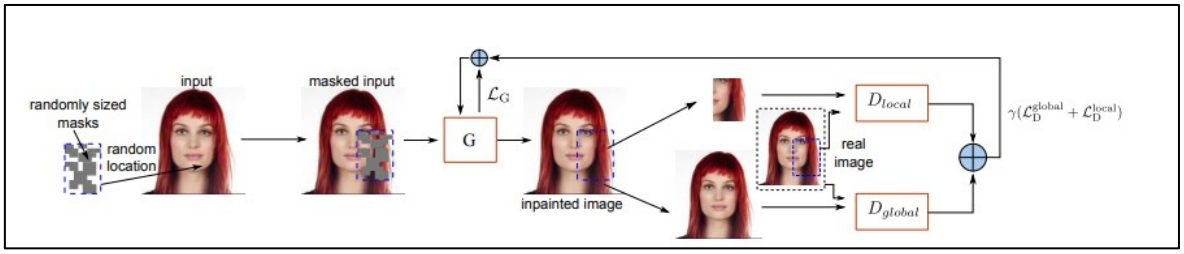
Metode obrane koje pretprocesiraju primjerke izmjenjuju ulazne testne primjerke prije nego ih predaju modelu tako da unište potencijalne uzorke okidača stražnjih vrata. Time se postiže da kompromitirani model još uvijek može ispravno klasificirati čak i otrovane slike, jer nakon izmjena primijenjenih nad uzorcima okidača, okidači više ne mogu aktivirati pripadna stražnja vrata.

Februus [23] je metoda obrane koja može neutralizirati stražnja vrata tijekom izvođenja (engl. „*at run-time*“). Sastoji se od dva koraka: kirurško otklanjanje regije slike za koju se misli da sadrži uzorak okidača (Sl. 2.3), a zatim generativno nadopunjavanje uklonjenog dijela slike s ciljem da učinkovitost klasifikacije nad obnovljenim primjercima bude približno ista kao nad čistim primjercima.



Sl. 2.3 Otklanjanje uzorka okidača iz slike (zamjena uzorka neutralnom bojom) [24]

GAN model (engl. *Generative Adversarial Network*) trenira se za obnavljanje primjeraka tako da generator na ulaz dobije sliku s maskom slučajno odabrane veličine i lokacije, dok diskriminator na ulaz dobije pravu sliku i generiranu sliku za usporedbu. Pritom se koriste dva diskriminatora, jedan za analizu globalne strukture slike, a drugi za analizu lokalne konzistencije u regiji prethodno uklonjene maske (Sl. 2.4).



Sl. 2.4 Treniranje GAN za obnavljanje primjeraka [25]

ShrinkPad [26] je metoda obrane koja pripada metodama obrane baziranim na operaciji transformacije. Transformacijske metode obrane vrše operaciju transformacije nad testnim primjercima prije donošenja odluke o klasi primjerka. Dakle, umjesto predviđanja klase za primjerak x, predviđa se klasa za transformirani primjerak T(x), gdje je T(.) neka

transformacija, na primjer skaliranje ili zrcaljenje slike. U slučaju metode ShrinkPad slika se smanjuje za nekoliko piksela, pritom koristeći bilinearnu interpolaciju i slučajno nadopunjavanje ruba novonastale slike pikselima čija vrijednost odgovara nuli.

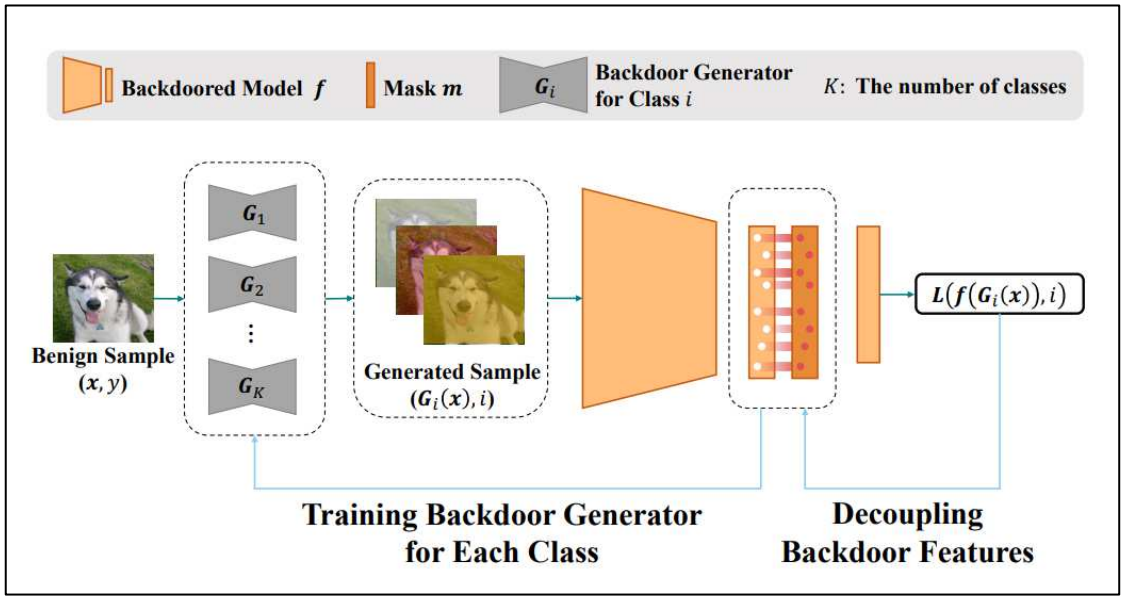
BTIDBF-P (engl. *Backdoor Trigger Inversion – Decoupling Benign Features, Purification variant*) je također metoda obrane koja uklanja stražnja vrata, a o njoj će biti riječi kasnije.

2.3. Metoda inverzije okidača stražnjih vrata

BTI (engl. *Backdoor trigger inversion*) je metoda obrane koja ima za cilj generirati zlćudni uzorak okidača, to jest otrovani primjerak za odgovarajući čisti primjerak na ulazu u generator.

Pomoću BTI metode, moguće je „odučiti“ (engl. *unlearn*) stražnja vrata u modelu, tako da generirane otrovane slike imaju ispravnu oznaku umjesto one nametnute od strane napadača.

BTI metodom je također moguće pretprocesirati sumnjive primjerke za vrijeme izvođenja, tako da se ukloni uzorak okidača i time spriječi aktivacija stražnjih vrata prilikom predikcije modela. Na slici (Sl. 2.5) je prikazan princip rada većine varijanti BTI metode:



Sl. 2.5 Princip rada većine varijanti BTI metode [27]

No, nedostaci koji se naziru u većini BTI metoda su niska učinkovitost inverzije te nizak stupanj sličnosti između generiranog uzorka okidača i stvarnog okidača.

Jedan razlog za ove nedostatke je što se prvo trebaju izolirati zloćudne značajke koje uzrokuju stražnja vrata kako bi se odvojile dobre i zloćudne značajke, te se treba provjeriti svaka klasa kako bi se odredila ciljna oznaka klase koju je napadač odredio za pojedini otrovani primjerak. Potrebno je napraviti zaseban generator za svaku klasu jer žrtva nema prethodnog znanja o detaljima napada [28].

U sljedećem poglavlju će biti riječi o BTI-DBF metodi obrane koja je proširenje BTI metode i otklanja njene spomenute nedostatke.

3. Detalji izvedbe BTI-DBF metode obrane

Umjesto izravnog procjenjivanja zloćudnih značajki koje čine stražnja vrata u modelu (kako to rade BTI metode), mogu se procijeniti i izdvojiti (engl. *decouple*) dobroćudne značajke. Pretpostavlja se da korisnik modela lokalno ima čiste primjerke, koji su onda u izravnoj vezi s dobroćudnim, ispravnim značajkama.

Metoda BTI-DBF (engl. *Backdoor Trigger Inversion – Decoupling Benign Features*) je varijanta BTI metode i sastoji se od dva koraka:

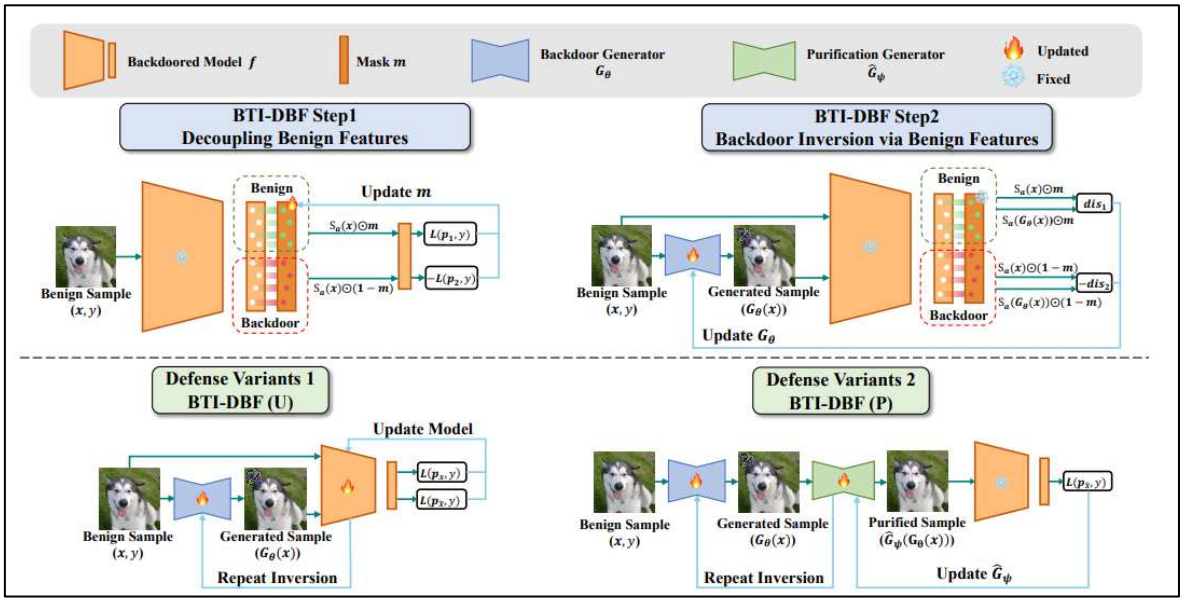
1. **Izdvajanje dobroćudnih značajki** optimiranjem cilja da promatrani model može raditi točne predikcije nad čistim primjercima samo s pomoću dobroćudnih značajki, dok će korištenje ostatka značajki voditi ka krivim predikcijama
2. **Treniranje generatora stražnjih vrata** minimiziranjem razlike između čistih primjeraka i odgovarajućih generiranih otrovanih primjeraka u njihovim dobroćudnim značajkama, i istovremeno maksimiziranjem razlike u zloćudnim značajkama

Neka je $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ čisti skup za treniranje koji sadrži N primjeraka. $\boldsymbol{x}_i \in \mathcal{X}$ predstavlja i-tu sliku, $y_i \in \mathcal{Y} = \{1, \dots, K\}$ predstavlja odgovarajuću oznaku, gdje je K broj klasa. Napadač će generirati otrovani skup podataka D_p koji se sastoji od dva glavna dijela, a to su izmijenjeni dio D_s i originalni čisti dio D_b, tako da vrijedi $D_p = D_m \cup D_b$, gdje $D_b \subset D$, $D_m = \{(\boldsymbol{x}', y') | \boldsymbol{x}' = G_X(\boldsymbol{x}), y' = G_Y(y), (\boldsymbol{x}, y) \in D_s\}$, $\gamma \triangleq \frac{|D_s|}{|D|}$ je stopa trovanja, a G_x i G_y su generator otrovanih slika odnosno generator otrovanih oznaka.

Pri formiranju predikcije kod modela sa stražnjim vratima, ako se na ulaz dovede dosad neviđen primjerak $\hat{\boldsymbol{x}}$ s pripadnom ispravnom oznakom $\hat{y}$, model će primjerak $\hat{\boldsymbol{x}}$ klasificirati oznakom $\hat{y}$, dok će za njegovu otrovanu verziju $G_X(\hat{\boldsymbol{x}})$ previdjeti oznaku klase $G_Y(\hat{\boldsymbol{y}})$.

Duboki model f se može razložiti na dva dijela: S_a dio koji predstavlja preslikavanje iz ulaznog prostora u prostor značajki (to su konvolucijski slojevi), i S_b dio koji predstavlja preslikavanje iz prostora značajki u izlazni prostor (to su potpuno povezani slojevi na izlazu).

Na sljedećoj slici (Sl. 3.1) prikazan je tok provođenja BTI-DBF metode:



Sl. 3.1 Tok provođenja BTI-DBF metode i njezine dvije varijante, BTI-DBF (U) varijanta koja otklanja stražnja vrata i BTI-DBF (P) varijanta koja pretprocesira i čisti ulazne primjerke [29]

Dakle, u općenitoj BTI-DBF metodi, prvo se nad čistim primjercima izdvajaju dobroćudne značajke generirane od strane konvolucijskih slojeva S_a, pa se zatim obavlja inverzija okidača minimiziranjem razlike između čistih primjeraka i pripadnih generiranih otrovanih primjeraka u prostoru značajki, i maksimiziranjem njihove razlike u odnosu na njihove preostale značajke (značajke stražnjih vrata) [30].

3.1. Izdvajanje dobroćudnih značajki

Intuitivno, model bi trebao moći raditi ispravne predikcije koristeći isključivo dobroćudne značajke, dok bi korištenjem ostalih značajki model radio krive predikcije.

Neka je $\boldsymbol{m}$ maska za odabir dobroćudnih značajki, koja ima dimenziju istu kao i dimenzija svih značajki, $S_a(.)$, te čije su vrijednosti unutar intervala [0,1]. Što je vrijednost elementa bliža 1, to je vjerojatnije da je značajka dobroćudna. Ako imamo skup čistih, ispravno označenih primjeraka D_l, postupak optimizacije se radi prema izrazu (2):

$$\underset{\boldsymbol{m}}{\operatorname{argmin}} \sum_{(\boldsymbol{x},y)\in D_l} \left[\mathcal{L}(S_b \circ (S_a(\boldsymbol{x}) \odot \boldsymbol{m}), y) - \mathcal{L}\big(S_b \circ (S_a(\boldsymbol{x}) \odot (\mathbf{1} - \boldsymbol{m})\big), y)\right], \tag{2}$$

Pritom je $\mathcal{L}$ funkcija gubitka te je $\odot$ operator umnoška po elementima (engl. *element-wise product*).

Ovom metodom se automatski ustanovljuje omjer dobroćudnih značajki bez potrebe da se ručno definira taj omjer. Također, za razliku od većine BTI metoda, ova metoda ne mora provjeravati sve klase kako bi ustanovila potencijalne ciljne oznake napadača, te je time vrlo učinkovita.

3.2. Inverzija okidača stražnjih vrata

Nakon optimiranja maske u fazi izdvajanja dobroćudnih značajki, slijedi treniranje generatora stražnjih vrata $G_{\boldsymbol{\theta}}: \mathcal{X} \to \mathcal{X}$ kako bi se mogli generirati otrovani primjerci iz čistih primjeraka. Generirani otrovani primjerci bi trebali imati slične vrijednosti dobroćudnih značajki, a različite vrijednosti u ostalim značajkama koje predstavljaju stražnja vrata.

Izraz za optimizacijski problem pri inverziji okidača je sljedeći (3):

$$\min_{\boldsymbol{\theta}} \sum_{(\boldsymbol{x},y)\in D_l} \big(\|S_a(\boldsymbol{x}) - S_a(G_{\boldsymbol{\theta}}(\boldsymbol{x}))) \odot \boldsymbol{m}\| - \big\|\big(S_a(\boldsymbol{x}) - S_a(G_{\boldsymbol{\theta}}(\boldsymbol{x}))\big) \odot (1 - \boldsymbol{m})\big\|\big), \tag{3}$$

$$\text{t.d. } \|\boldsymbol{x} - G_{\boldsymbol{\theta}}(\boldsymbol{x})\| \leq \tau, \forall (\boldsymbol{x}, y) \in D_l$$

gdje je τ hiperparametar za koji vrijedi $\tau > 0$, a $\|\cdot\|$ je mjera udaljenosti, na primjer L2 norma.

Uzorak okidača primjerka $\boldsymbol{x}$ može se dobaviti pomoću izraza $G_\theta(\boldsymbol{x}) - \boldsymbol{x}$ [31].

3.3. BTI-DBF varijanta s odučavanjem

Varijanta BTI-DBF metode s odučavanjem (engl. *unlearning*), BTI-DBF (U), omogućuje da model „oduči“ stražnja vrata te da ih ukloni, tako da se kompromitirani model $f_{\boldsymbol{w}}$ naknadno uči na generiranim otrovanim slikama koje imaju ispravne oznake umjesto ciljnih oznaka koje je postavio napadač.

Dakle, optimizira se s ciljem da model može ispravno klasificirati i čiste primjerke i generirane otrovane primjerke s uzorkom okidača, gdje je svaki par takvih primjeraka blizu u prostoru značajki, pa je izraz sljedeći (4):

$$\min_{\boldsymbol{w}} \sum_{(\boldsymbol{x},y)\in D_l} [\mathcal{L}(f_{\boldsymbol{w}}(\boldsymbol{x}),y) + \mathcal{L}(f_{\boldsymbol{w}}(G_{\boldsymbol{\theta}}(\boldsymbol{x})),y) + \|S_a(\boldsymbol{x}) - S_a(G_{\boldsymbol{\theta}}(\boldsymbol{x}))\|\,], \tag{4}$$

Algoritam za BTI-DBF (U) metodu izgleda ovako [32]:

Ulaz u algoritam je: sumnjivi model $f_{\boldsymbol{w}}$ s parametrima $\boldsymbol{w}$, mali podskup lokalnih čistih primjeraka D_l, hiperparametar τ, stope promjene parametara α_1, α_2, α_3, broj iteracija za DBF fazu I_1, broj iteracija za BTI fazu I_2, broj iteracija za odučavanje stražnjih vrata I_3, i broj izmjenjujućih iteracija R_1. Izlaz algoritma je pročišćeni model $f_{\boldsymbol{w}}$.

for r in $range(R_1)$ **do**
  Initial $\boldsymbol{m} \leftarrow \boldsymbol{0}$
  for i in $range(I_1)$ **do**
    $\ell_m \leftarrow \mathcal{L}(S_b \circ (S_a(\boldsymbol{x}) \odot \boldsymbol{m}), y) - \mathcal{L}\big(S_b \circ (S_a(\boldsymbol{x}) \odot (\boldsymbol{1}-\boldsymbol{m})), y\big)$
    Update $\boldsymbol{m} \leftarrow \boldsymbol{m} - \alpha_1 \cdot \nabla_{\boldsymbol{m}} \ell_{\boldsymbol{m}}$
  for j in $range(I_2)$ **do**
    if $\| \boldsymbol{x} - G_{\boldsymbol{\theta}}(\boldsymbol{x}) \| \leq \tau$ **then**
      $\ell_{\boldsymbol{\theta}} \leftarrow \|S_a(\boldsymbol{x}) - S_a(G_{\boldsymbol{\theta}}(\boldsymbol{x}))) \odot \boldsymbol{m}\| - \big\|\big(S_a(\boldsymbol{x}) - S_a(G_{\boldsymbol{\theta}}(\boldsymbol{x}))\big) \odot (1-\boldsymbol{m})\big\|$
    else
      $\ell_{\boldsymbol{\theta}} \leftarrow \| x - G_{\boldsymbol{\theta}}(x) \|$
      Update $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha_2 \cdot \nabla_{\boldsymbol{\theta}} \ell_{\boldsymbol{\theta}}$
  for k in $range(I_3)$ **do**
    $\ell_{\boldsymbol{w}} \leftarrow \mathcal{L}(f_{\boldsymbol{w}}(\boldsymbol{x}),y) + \mathcal{L}(f_{\boldsymbol{w}}(G_{\boldsymbol{\theta}}(\boldsymbol{x})),y) + \|S_a(\boldsymbol{x}) - S_a(G_{\boldsymbol{\theta}}(\boldsymbol{x}))\|$
    Update $\boldsymbol{w} \leftarrow \boldsymbol{w} - \alpha_3 \cdot \nabla_{\boldsymbol{w}} \ell_{\boldsymbol{w}}$

3.4. BTI-DBF varijanta s pročišćavanjem

Kako bi se pročistili otrovani ulazni primjerci, potrebno je istrenirati generator za pročišćavanje (engl. *purification generator*) $\hat{G}_{\boldsymbol{\psi}}: \mathcal{X} \to \mathcal{X}$. Ideja je da se s pomoću generatora stražnjih vrata $G_{\boldsymbol{\theta}}$ može procijeniti njegova inverzna funkcija kako bi se dobio generator za pročišćavanje.

Potrebno je zadovoljiti sljedeća dva zahtjeva:

1. postupak pročišćavanja može ukloniti okidače stražnjih vrata,
2. a da pritom ne utječe na čiste primjerke

Za potrebe optimizacije definiraju se gubitak pročišćenja $\mathcal{L}_p$ (engl. *purification loss*) i dobroćudni gubitak $\mathcal{L}_b$ (engl. *benign loss*).

$$\min_{\psi} \sum_{(\boldsymbol{x},y)\in D_l} \mathcal{L}_p + \mathcal{L}_b \,, \tag{5}$$

gdje vrijedi

$$\mathcal{L}_p = \mathcal{L}\left(f\left(\widehat{G_\psi}(G_{\boldsymbol{\theta}}(\boldsymbol{x}))\right), y\right) + \left\|S_a(\boldsymbol{x}) - S_a\left(\widehat{G_\psi}(G_{\boldsymbol{\theta}}(\boldsymbol{x}))\right)\right\| + \|x - \widehat{G_\psi}(G_{\boldsymbol{\theta}}(\boldsymbol{x}))\|, \tag{6}$$

$$\mathcal{L}_b = \mathcal{L}\left(f\left(\widehat{G_\psi}(\boldsymbol{x})\right), y\right) + \left\|S_a(\boldsymbol{x}) - S_a\left(\widehat{G_\psi}(\boldsymbol{x})\right)\right\| + \|\boldsymbol{x} - \widehat{G_\psi}(\boldsymbol{x})\| \tag{7}$$

Potom se sumnjivi primjerci mogu pretprocesirati prije nego li dođu na ulaz modela radi predikcije, tako da se u njima deaktiviraju skrivena stražnja vrata [33].

Ulaz u algoritam pročišćenja je: sumnjivi model f, mali podskup lokalnih čistih primjeraka D_l, hiperparametar τ, stope promjene parametara α_1, α_2, α_3, broj iteracija za DBF fazu I_1, broj iteracija za BTI fazu I_2, broj iteracija za odučavanje stražnjih vrata I_3, i broj izmjenjujućih iteracija R_1. Izlaz algoritma je generator za pročišćavanje $\widehat{G_\psi}$.

Pseudokôd algoritma je sljedeći [34]:

1: **for** r in $range(R_1)$ **do**

2: Initial $\mathbf{m} \leftarrow \mathbf{0}$

3: **for** i in $range(I_1)$ **do**

4: $\ell_{\boldsymbol{m}} \leftarrow \mathcal{L}(S_b \circ (S_a(\boldsymbol{x}) \odot \boldsymbol{m}), y) - \mathcal{L}\big(S_b \circ (S_a(\boldsymbol{x}) \odot (\mathbf{1} - \boldsymbol{m})\big), y)$

5: Update $\boldsymbol{m} \leftarrow \boldsymbol{m} - \alpha_1 \cdot \nabla_{\boldsymbol{m}} \ell_{\boldsymbol{m}}$

6: **for** j in $range(I_2)$ **do**

7: **if** $\| \boldsymbol{x} - G_{\boldsymbol{\theta}}(\boldsymbol{x}) \| \leq \tau$ **then**

8: if $\mathbf{r} = 0$ **then**

9: $\ell_1 \leftarrow \|(S_a(\boldsymbol{x}) - S_a(G_{\boldsymbol{\theta}}(\boldsymbol{x}))) \odot \boldsymbol{m}\|$

10: $\ell_2 \leftarrow \|(S_a(\boldsymbol{x}) - S_a(G_{\boldsymbol{\theta}}(\boldsymbol{x}))) \odot (1 - \boldsymbol{m})\|$

11: **else**

12: $\ell_1 \leftarrow \left\|(S_a(\boldsymbol{x}) - S_a(\widehat{G_\psi}(G_{\boldsymbol{\theta}}(\boldsymbol{x})))) \odot \boldsymbol{m}\right\|$

13: $\ell_2 \leftarrow \left\|(S_a(\boldsymbol{x}) - S_a(\widehat{G_\psi}(G_{\boldsymbol{\theta}}(\boldsymbol{x})))) \odot (1 - \boldsymbol{m})\right\|$

14: $\ell_{\boldsymbol{\theta}} \leftarrow \ell_1 - \ell_2$

15: **else**

16: $\ell_{\boldsymbol{\theta}} \leftarrow \| x - G_{\boldsymbol{\theta}}(x) \|$

17: Update $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha_2 \cdot \nabla_{\boldsymbol{\theta}} \ell_{\boldsymbol{\theta}}$

18: **for** k in $range(I_3)$ **do**

19: $\ell_p \leftarrow \mathcal{L}\left(f\left(\widehat{G_\psi}(G_{\boldsymbol{\theta}}(\boldsymbol{x}))\right), y\right) + \left\|S_a(\boldsymbol{x}) - S_a\left(\widehat{G_\psi}(G_{\boldsymbol{\theta}}(\boldsymbol{x}))\right)\right\| + \left\|x - \widehat{G_\psi}(G_{\boldsymbol{\theta}}(\boldsymbol{x}))\right\|$

20: $\ell_b \leftarrow \mathcal{L}\left(f\left(\widehat{G_\psi}(\boldsymbol{x})\right), y\right) + \left\|S_a(\boldsymbol{x}) - S_a\left(\widehat{G_\psi}(\boldsymbol{x})\right)\right\| + \left\|\boldsymbol{x} - \widehat{G_\psi}(\boldsymbol{x})\right\|$

21: $\ell_{\boldsymbol{\psi}} = \ell_p + \ell_b$

22: Update $\psi \leftarrow \psi - \alpha_3 \cdot \nabla_\psi \ell_\psi$

3.5. Modifikacija BTI-DBF metode – prelazak u prostor značajki

U ovom radu isprobana je modifikacija BTI-DBF metode gdje se generator otrovanih primjeraka modificira tako da:

- generator na ulaz umjesto čistih primjeraka prima pripadne dobroćudne značajke primjeraka
- na izlazu se umjesto generiranja otrovanih primjeraka generiraju odgovarajuće otrovane značajke

Radi iznimne učinkovitosti BTI-DBF varijante s odučavanjem (kako je kasnije pokazano u eksperimentu), ona je odabrana za pokazni primjer ove modifikacije. Novi optimizacijski problem za BTI-DBF (U-FS) (engl. *Unlearning – Feature* Space) se može definirati sljedećim izrazom (8):

$$\min_{\boldsymbol{w}} \sum_{(\boldsymbol{x},y)\in D_l} \begin{bmatrix} \mathcal{L}(f_{\boldsymbol{w}}(\boldsymbol{x}),y) + \mathcal{L}\big(f_{\boldsymbol{w}}\big(G_{\boldsymbol{\theta}}(\boldsymbol{x})\big),y\big) + \big\|S_a(\boldsymbol{x}) - G_{\boldsymbol{\theta}}\big(S_a(\boldsymbol{x})\big)\big\| \\ + \big|\,\|S_a(\boldsymbol{x})\| - \big\|G_{\boldsymbol{\theta}}\big(S_a(\boldsymbol{x})\big)\big\|\,\big| \end{bmatrix}, \tag{8}$$

gdje je bitna razlika u odnosu na izraz (4) ta da se L2 norma računa izravno s izlazom generatora $G_{\boldsymbol{\theta}}(S_a(\boldsymbol{x}))$ koji predstavlja vektor zloćudnih značajki, i s vektorom dobroćudnih značajki $S_a(\boldsymbol{x})$, koje su ujedno i ulaz u generator.

Isto tako, u izraz je dodana apsolutna razlika normi navedenih vektora značajki, kako bi se osiguralo da norme tih vektora budu slične vrijednosti.

4. Eksperimenti

4.1. Okruženje

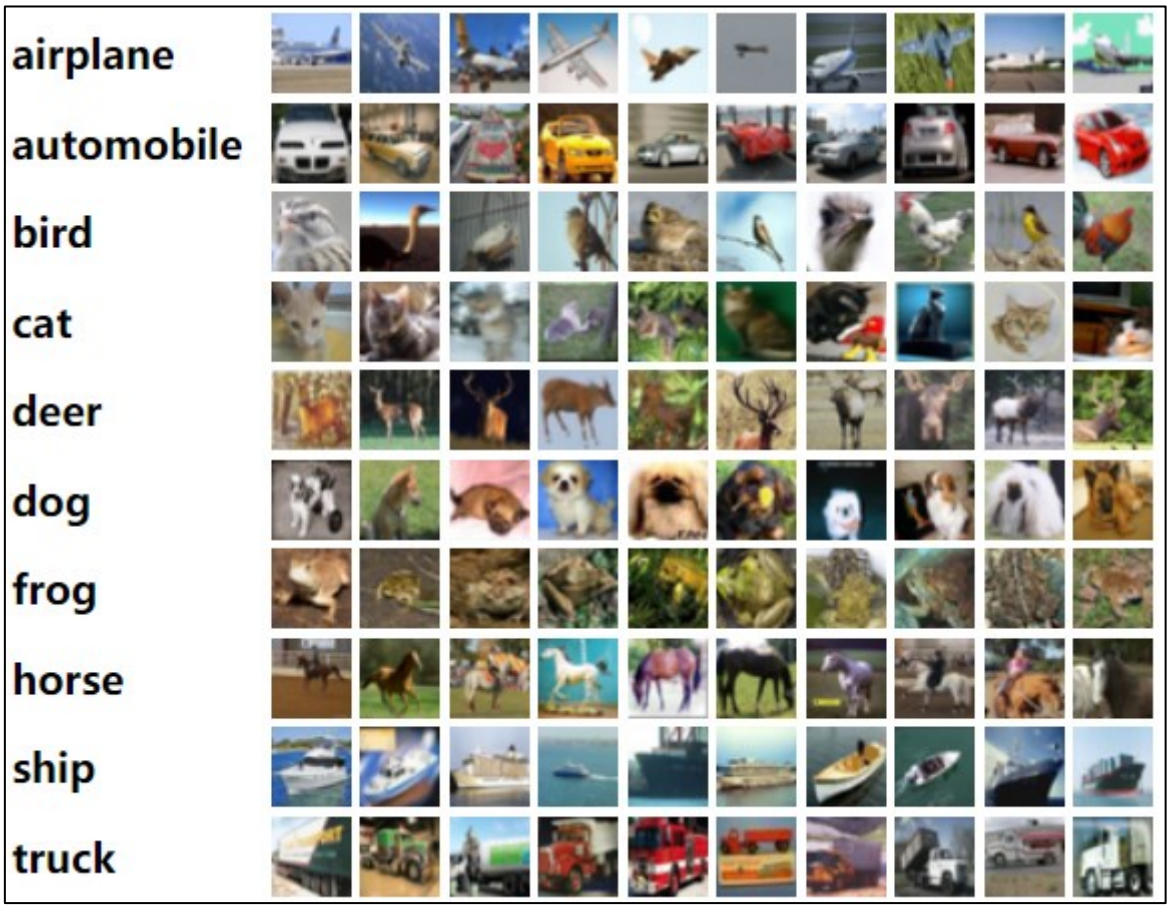
Za implementaciju algoritama koristi se Python 3.10 programski jezik, i PyTorch biblioteka za izradu dubokih modela.

Modeli su trenirani na Google Colab-u u Jupyter bilježnici.

Korištena je NVIDIA T4 Tensor Core GPU za treniranje modela.

4.1.1. Skup podataka

Korišten je CIFAR-10 skup podataka koji sadrži 50 000 primjeraka za treniranje i 10 000 testnih primjeraka. Slike su dimenzija 3x32x32. Svaka slika pripada jednoj od 10 klasa (Sl. 4.1).



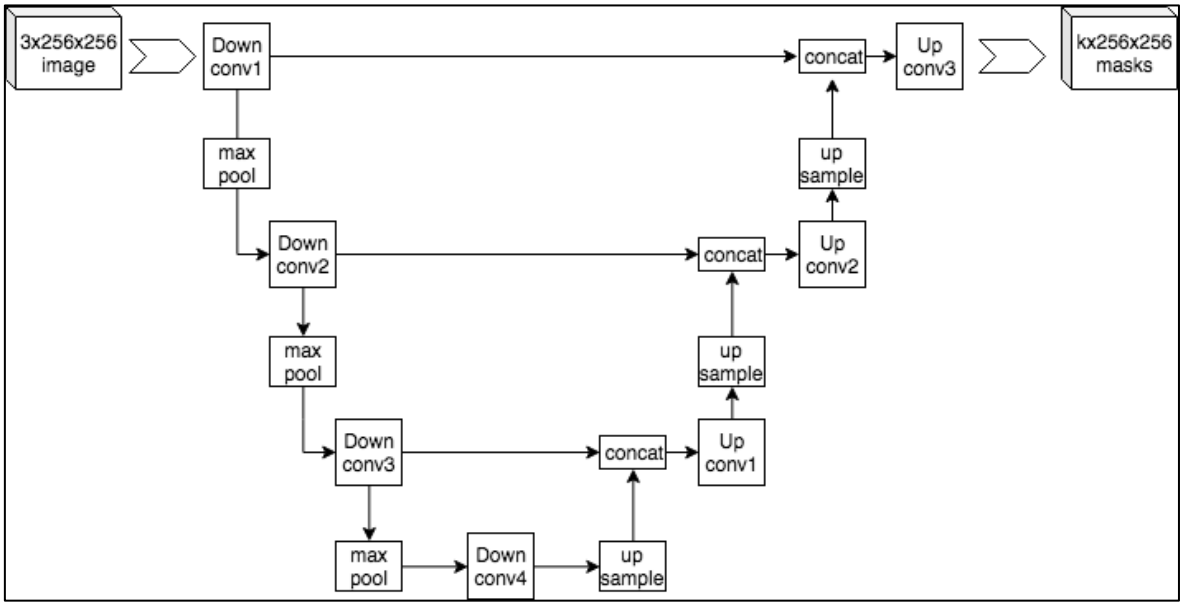
Sl. 4.1 Prikaz odabranih slika u CIFAR-10 skupu podataka [35]

4.1.2. Arhitektura modela

Za modele svih generatora koristi se U-Net arhitektura (osim za BTI-DBF (U-FS) varijantu gdje se koristi model s linearnim slojem za preslikavanje vektora značajki), koja se sastoji od dijela koji se sužava i dijela koji se širi.

Dio koji se sužava je obična konvolucijska mreža s konvolucijskim i ReLU slojevima te max pooling operacijama, te se u ovom dijelu smanjuje prostorna informacija (rezolucija slike), a povećava se informacija o značajkama.

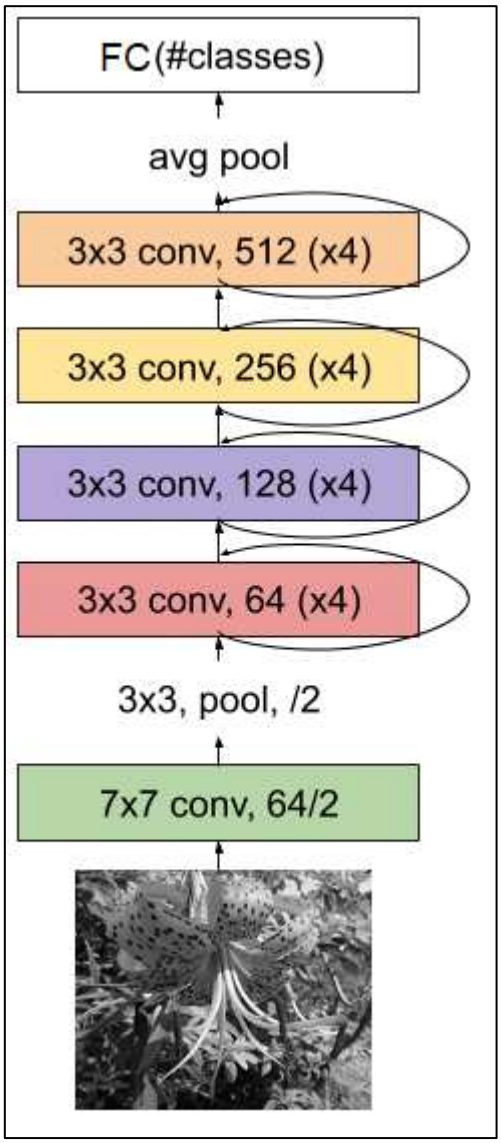
Dio koji se širi kombinira informaciju o značajkama s prostornom informacijom pomoću niza operacija naduzorkovanja (engl. *upsample*) i nadovezivanja (engl. *concatenate*) sa značajkama na visokoj rezoluciji dobivenoj iz dijela koji se sužava (Sl. 4.2) [36].



Sl. 4.2 U-Net arhitektura [37]

Za klasifikatorsku i evaluatorsku neuronsku mrežu koristi se ResNet-18 arhitektura.

ResNet-18 arhitektura sastoji se od 18 slojeva te se bazira na rezidualnim blokovima, koji mreži omogućuju da uči rezidualne funkcije koje ovise o ulazima u sloj, što otklanja problem nestajućeg gradijenta i time omogućava treniranje vrlo dubokih modela. Slika (Sl. 4.3) prikazuje strukturu ResNet-18 arhitekture [38].



Sl. 4.3 Struktura ResNet-18 arhitekture [39]

4.1.3. Ostale postavke

Postotak otrovanih primjeraka postavljen je na 5%.

Također se odabire 5% čistih primjeraka za treniranje kao lokalni skup podataka za sve metode obrane.

Pri treniranju modela koristi se ADAM optimizator. Stopa učenja je 10^{-4}.

Obavlja se 5 iteracija treniranja modela, gdje se u svakoj iteraciji izvrši 20 epoha za DBF fazu i 30 epoha za BTI fazu.

Za ostale napade korištene su originalne postavke iz njihovih izvornih članaka.

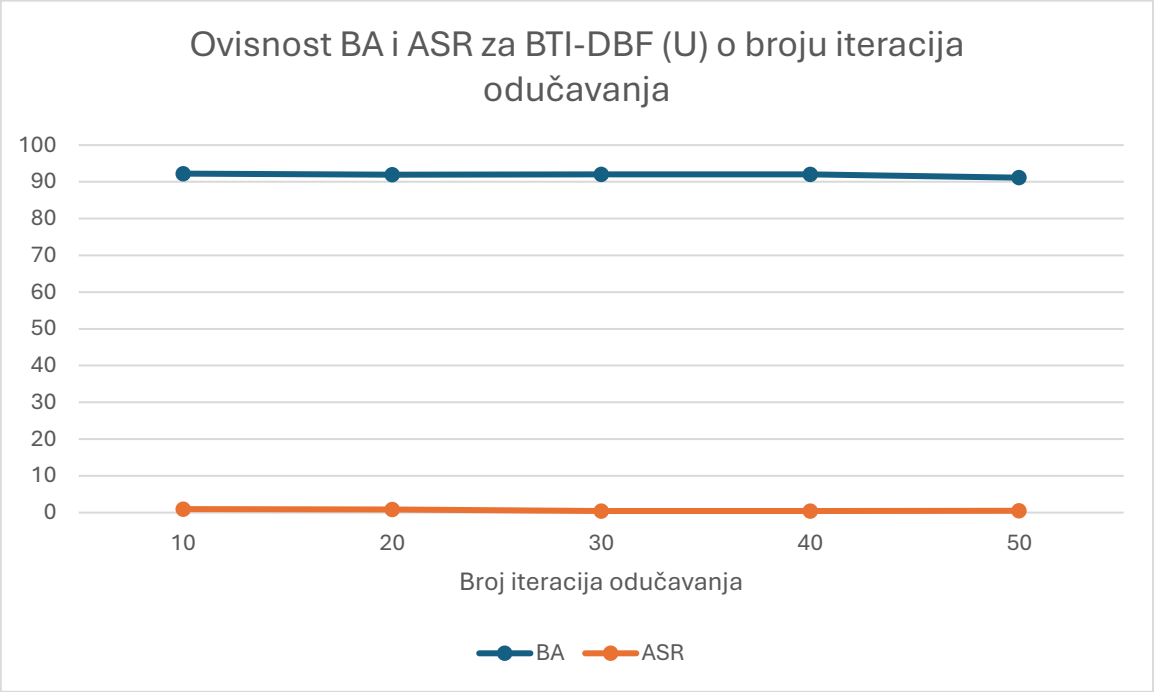
Generatori su predtrenirani na CIFAR-10 podacima kroz 50 epoha.

4.2. Optimizacija hiperparametara modela

Za optimiranje odabrani su hiperparametri:

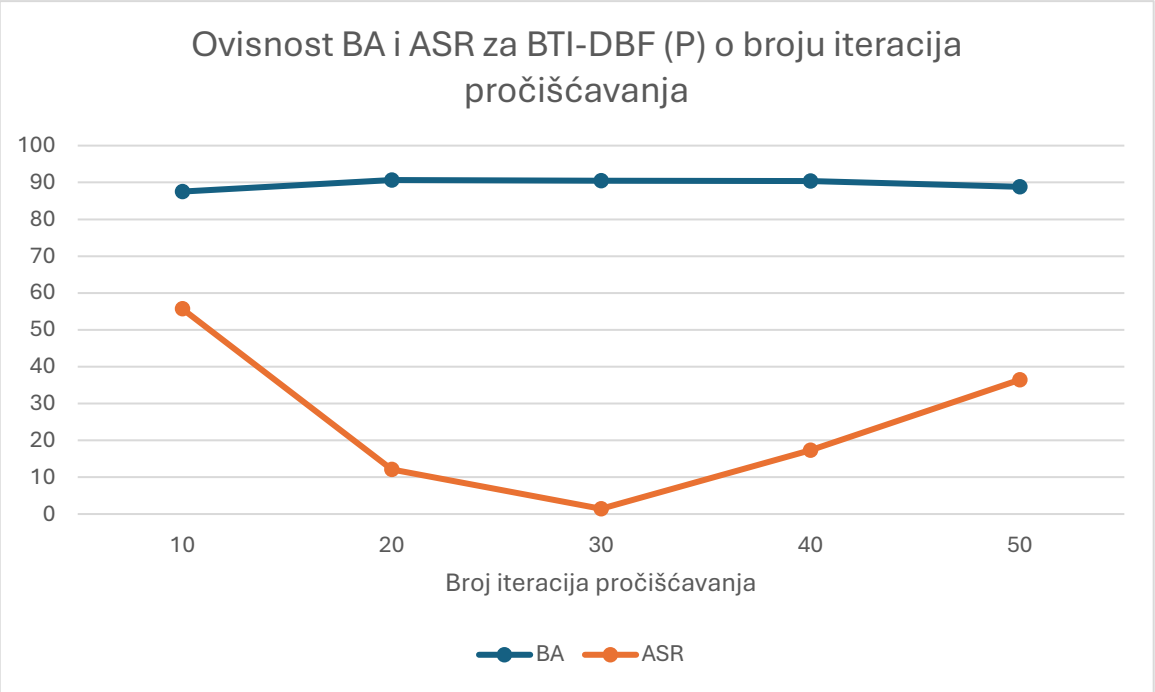
- broj iteracija odučavanja
- broj iteracija pročišćavanja

Na slici (Sl. 4.4) prikazana je ovisnost BA i ASR za BTI-DBF (U) o broju iteracija odučavanja. Vrijednosti su prilično monotone, a najbolja vrijednost za ASR nazire se za 30 iteracija, stoga je odabrana vrijednost 30 za broj iteracija odučavanja.



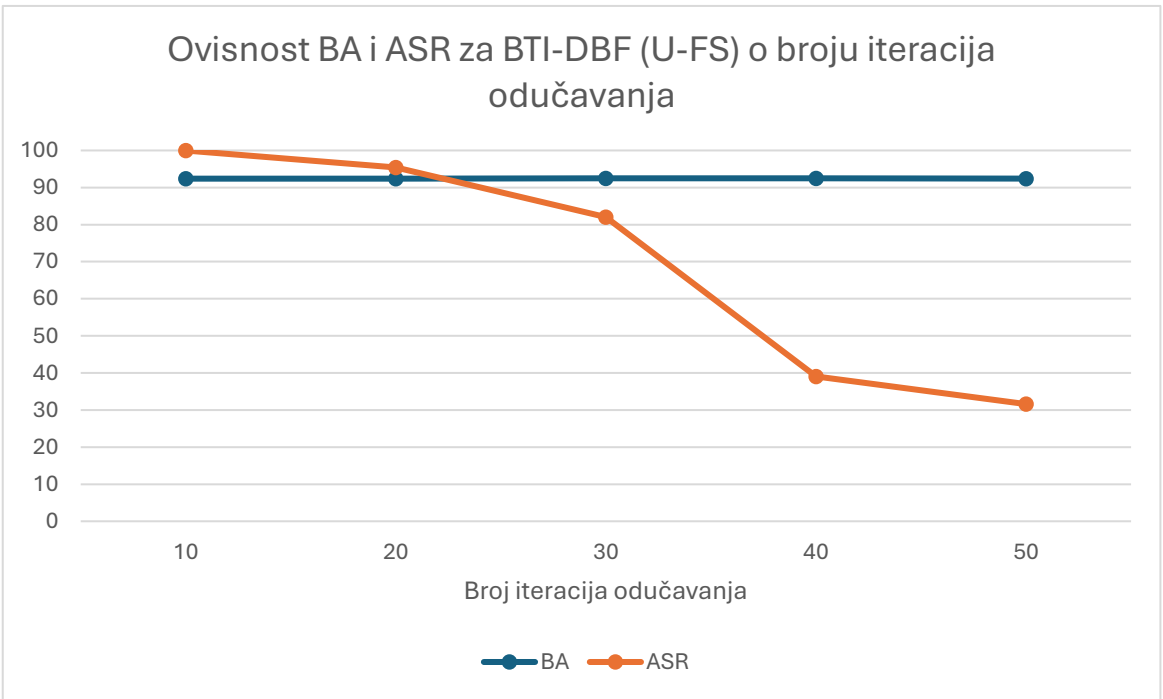
Sl. 4.4 Ovisnost BA i ASR za BTI-DBF (U) o broju iteracija odučavanja

Na slici (Sl. 4.5) prikazana je ovisnost BA i ASR za BTI-DBF (P) o broju iteracija pročišćavanja. Najbolja vrijednost za ASR očigledno se nazire za 30 iteracija, stoga je odabrana vrijednost 30 za broj iteracija pročišćavanja.



Sl. 4.5 Ovisnost BA i ASR za BTI-DBF (P) o broju iteracija pročišćavanja

Na slici (Sl. 4.6) prikazana je ovisnost BA i ASR za BTI-DBF (U-FS) o broju iteracija odučavanja. Najbolja vrijednost za ASR nazire se za 50 iteracija, stoga je odabrana vrijednost 50 za broj iteracija odučavanja.



Sl. 4.6 Ovisnost BA i ASR za BTI-DBF (U-FS) o broju iteracija odučavanja

4.3. Efikasnost metoda obrane

Za procjenu efikasnosti metoda obrane koriste se sljedeće dvije metrike:

- **BA** (engl. *benign accuracy*) - **točnost predviđanja nad čistim podacima,** iliti omjer čistih uzoraka bez okidača čija je oznaka klase točno predviđena, naspram svih čistih uzoraka
- **ASR** (engl. *attack success rate*) – **stopa uspješnosti napada**, omjer otrovanih uzoraka čija predviđena oznaka klase odgovara ciljnoj klasi koja je nametnuta od strane napadača

Metoda obrane je time učinkovitija što ima veću BA mjeru i manju ASR mjeru.

Sljedeća tablica (Tablica 4.1) prikazuje efikasnost različitih metoda obrane uklanjanjem stražnjih vrata u odnosu na 5 odabranih napada:

Tablica 4.1 Efikasnost različitih metoda obrane uklanjanjem stražnjih vrata u odnosu na odabrane napade

Obrane → / Napadi ↓	NAD		I-BAU		AWM		FeatureRE		BTI-DBF (U)		BTI-DBF (U-FS)	
	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR
BadNets	90.32	2.98	90.67	1.33	90.93	21.92	91.53	35.42	92.04	0.43	92.49	88.34
Blended	89.49	3.29	88.57	0.56	90.07	38.14	92.86	40.50	90.80	4.72	91.61	7.85
WaNet	91.38	6.94	90.66	2.88	91.01	10.72	93.75	0.02	89.00	1.24	93.85	27.42
IAD	91.34	17.45	91.11	9.63	88.76	14.71	93.23	0.39	91.60	1.43	93.63	15.66
LC	91.77	12.65	93.12	1.60	92.07	12.61	94.54	10.49	91.74	0.93	92.50	32.85

Iz tablice je vidljivo da BTI-DBF (U) ima konzistentno nisku ASR mjeru (uvijek ispod 5% za razliku od ostalih metoda s uklanjanjem stražnjih vrata), a da za to ne žrtvuje mnogo u BA mjeri.

S druge strane, BTI-DBF (U-FS) daje nešto lošije rezultate za ASR mjeru (konzistentno iznad 5%), upućujući na to da inačica BTI-DBF metode s odučavanjem u prostoru značajki radi lošije od BTI-DBF metode s odučavanjem u originalnom prostoru slika.

Sljedeća tablica (Tablica 4.2Tablica 4.1) prikazuje efikasnost različitih metoda obrane pretprocesiranjem primjeraka u odnosu na 5 odabranih napada:

Tablica 4.2 Efikasnost različitih metoda obrane pretprocesiranjem primjeraka u odnosu na odabrane napade

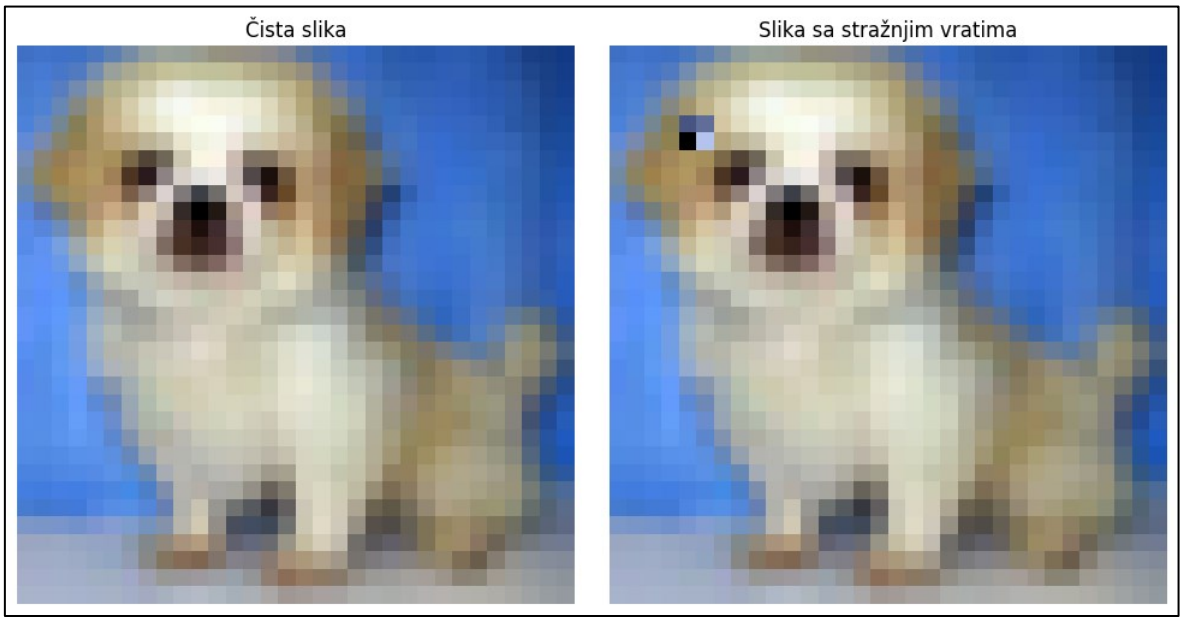
Obrane → Napadi ↓	Februus		ShrinkPad		BTI-DBF (P)	
	BA	ASR	BA	ASR	BA	ASR
BadNets	90.14	2.22	84.21	2.04	90.14	1.43
Blended	82.92	5.04	82.69	75.13	88.98	2.85
WaNet	69.36	49.55	45.60	96.56	91.18	2.82
IAD	66.45	32.40	88.14	35.83	90.35	3.11
LC	71.51	16.95	88.37	2.13	89.61	2.75

Iz tablice je vidljivo da BTI-DBF (P) ima konzistentno ima konzistentno nisku ASR mjeru (uvijek ispod 5% za razliku od ostalih metoda s pretprocesiranjem primjeraka), a da za to ne žrtvuje mnogo u BA mjeri, slično kao i kod BTI-DBF (U) metode.

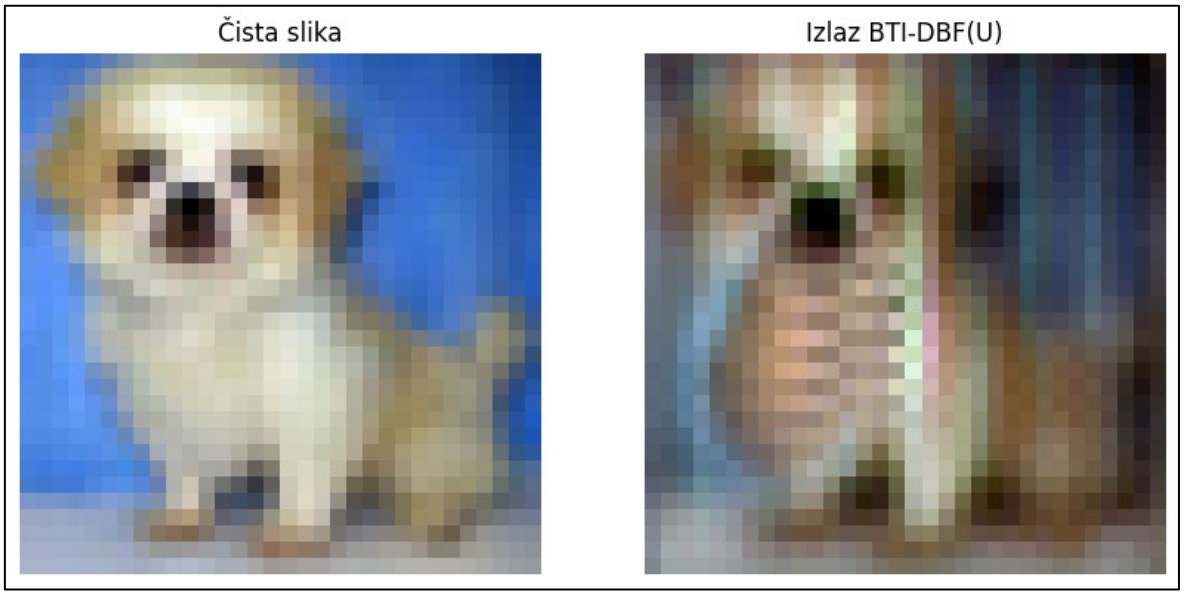
4.4. Primjeri rezultata

Ovdje su navedeni vizualni primjeri rezultata primjene BTI-DBF (U) metode za različite napade, izvedene na slici psa iz CIFAR-10 skupa podataka.

4.4.1. BadNets



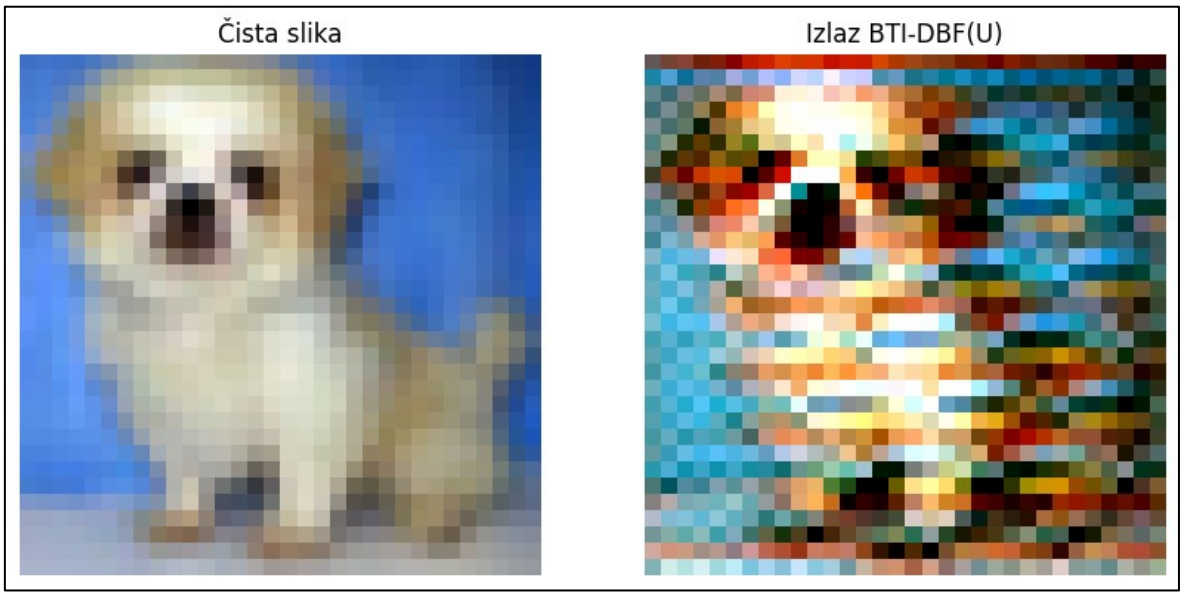
Sl. 4.7 Čista slika i otrovana slika uslijed BadNets napada



Sl. 4.8 Čista slika i generirana otrovana slika uslijed primjene BTI-DBF (U) metode protiv BadNets napada

4.4.2. Blended

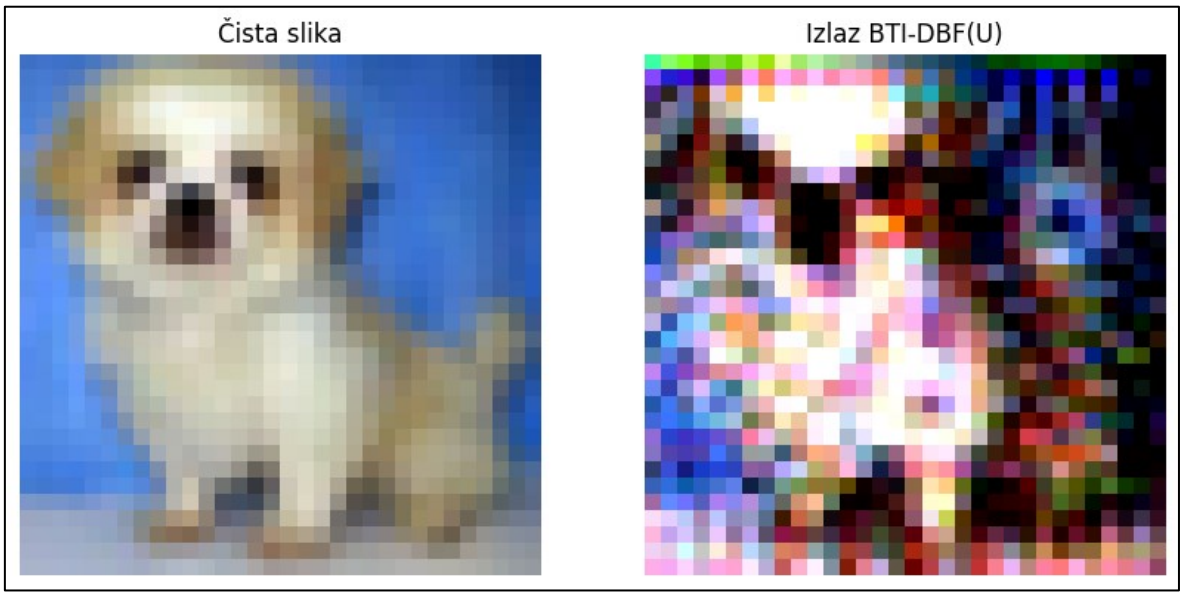
Sl. 4.9 Čista slika i otrovana slika uslijed Blended napada



Sl. 4.10 Čista slika i generirana otrovana slika uslijed primjene BTI-DBF (U) metode protiv Blended napada

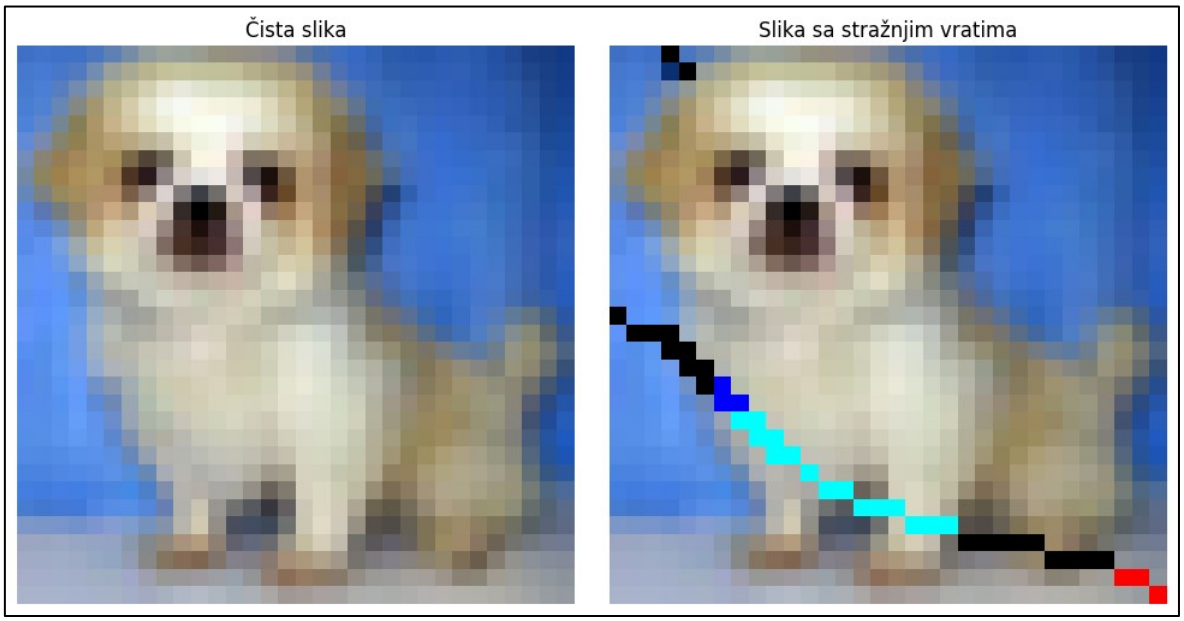
4.4.3. WaNet

Sl. 4.11 Čista slika i otrovana slika uslijed WaNet napada

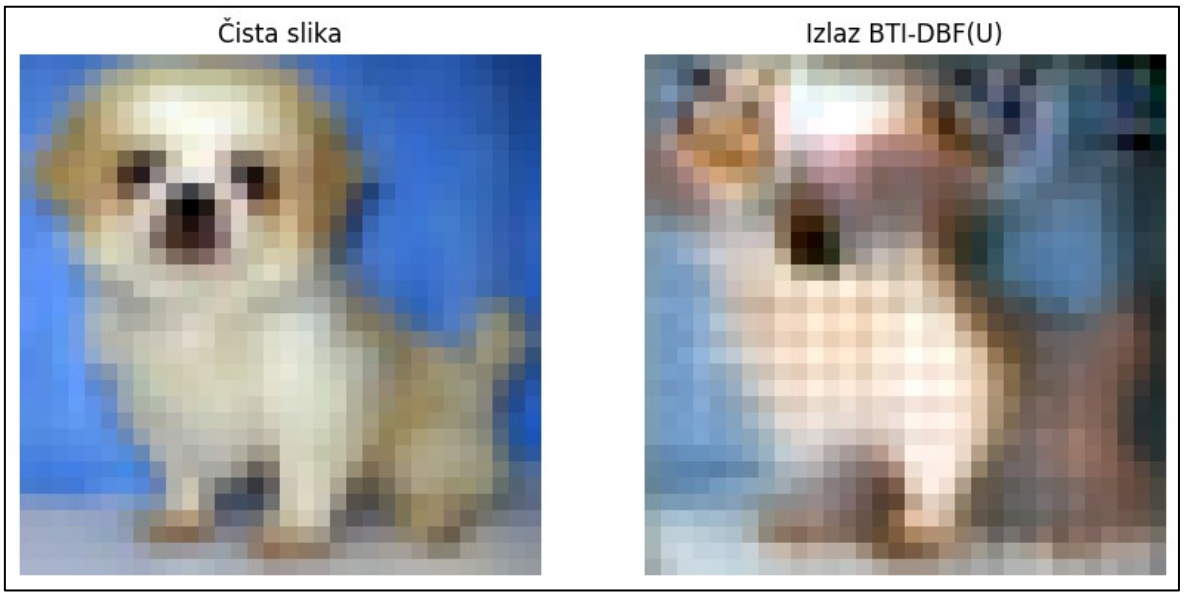


Sl. 4.12 Čista slika i generirana otrovana slika uslijed primjene BTI-DBF (U) metode protiv WaNet napada

4.4.4. IAD



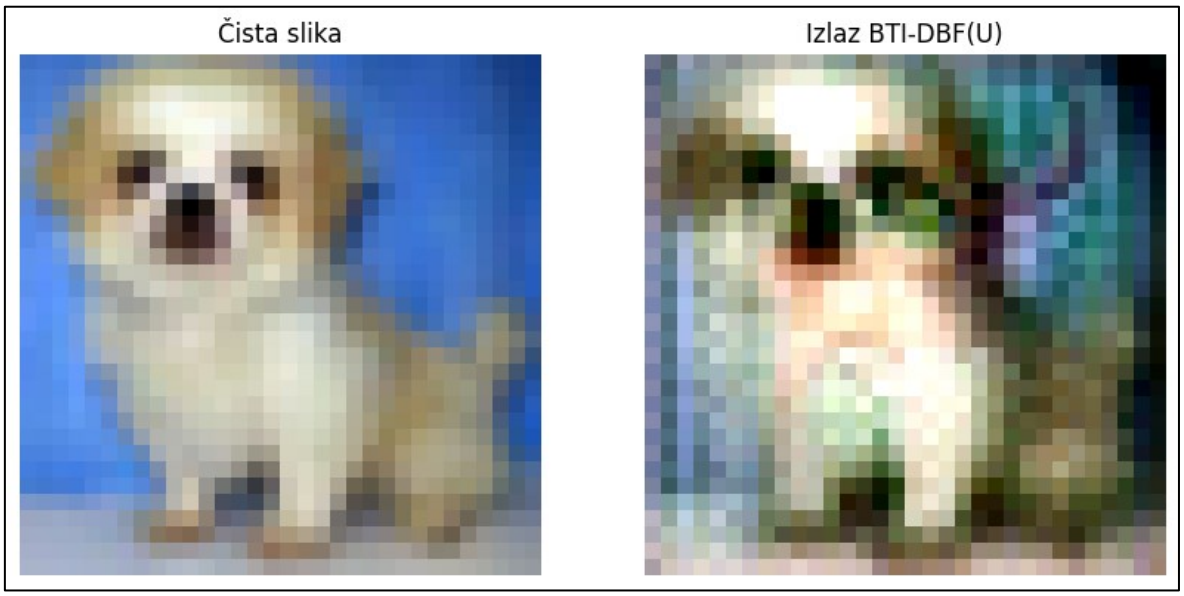
Sl. 4.13 Čista slika i otrovana slika uslijed IAD napada



Sl. 4.14 Čista slika i generirana otrovana slika uslijed primjene BTI-DBF (U) metode protiv IAD napada

4.4.5. LC

Sl. 4.15 Čista slika i otrovana slika uslijed LC napada



Sl. 4.16 Čista slika i generirana otrovana slika uslijed primjene BTI-DBF (U) metode protiv LC napada

Zaključak

Detekcija i otklanjanje stražnjih vrata u modelu je složen i aktualan problem u kontekstu dubokog učenja. Prilikom povjere treniranja modela vanjskom davatelju usluge, ili prilikom korištenja podataka izvana, bitno je imati sigurnost na umu kako bi se izbjegle loše posljedice uslijed postojanja zlućudnih stražnjih vrata u modelu.

U prvu ruku, poželjno je koristiti usluge od isključivo pouzdanih davatelja usluga.

Ako se koriste nepouzdani izvori, važno je osigurati se s pomoću metoda detekcije i otklanjanja stražnjih vrata.

Postoji mnogo metoda obrane od stražnjih vrata, ali i mnogo više napada koji se baziraju na umetanju stražnjih vrata te koji pokušavaju zaobići postojeće metode obrane [40].

BTI-DBF je dokazano moćna metoda obrane protiv napada umetanjem stražnjih vrata, te učinkovito može otklanjati stražnja vrata i pročišćavati ulazne primjerke koji u sebi imaju okidač.

U usporedbi s ostalim metodama obrane ima konzistentno bolje rezultate, što je najizraženije kod varijante s odučavanjem stražnjih vrata u generatoru.

Literatura

[1] Yansong Gao, Bao Gia Doan, Zhi Zhang, Siqi Ma, Jiliang Zhang, Anmin Fu, Surya Nepal, Hyoungshick Kim. *Backdoor Attacks and Countermeasures on Deep Learning: A Comprehensive Review*. arXiv preprint arXiv:2007.10760, 2020.

[2] Hanxun Huang, Xingjun Ma, Sarah Erfani, James Bailey. *Distilling cognitive backdoor patterns within an image*. Proceedings of the International Conference on Learning Representations, Kigali, (2023)

[3] Shesh Narayan Gupta, *Deep Convolutional Neural Networks (DCNNs) explained in layman's terms*, Medium, (2022, siječanj). Poveznica: https://medium.com/aiguys/deep-convolutional-neural-networks-dcnns-explained-in-layman-terms-b990b2818061; pristupljeno 25.6.2024.

[4] Tianyu Gu, Brendan Dolan-Gavitt, Siddharth Garg. *BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain*. arXiv preprint arXiv:1708.06733, 2019.

[5] Xiong Xu, Kunzhe Huang, Yiming Li, Zhan Qin, Kui Ren. *Towards Reliable and Efficient Backdoor Trigger Inversion via Decoupling Benign Features*. Proceedings of the International Conference on Learning Representations, Beč, (2024)

[6] Yansong Gao, Bao Gia Doan, Zhi Zhang, Siqi Ma, Jiliang Zhang, Anmin Fu, Surya Nepal, Hyoungshick Kim. *Backdoor Attacks and Countermeasures on Deep Learning: A Comprehensive Review*. arXiv preprint arXiv:2007.10760, 2020.

[7] Ben Dickson, *TrojanNet – a simple yet effective attack on machine learning models*, PortSwigger, (2020, srpanj). Poveznica: https://portswigger.net/daily-swig/trojannet-a-simple-yet-effective-attack-on-machine-learning-models; pristupljeno 25.6.2024.

[8] Siddharth Garg. *BadNets: Evaluating Backdooring Attacks on Deep Neural Networks*, ResearchGate, (2019, siječanj). Poveznica: https://www.researchgate.net/figure/Approaches-to-backdooring-a-neural-network-The-backdoor-trigger-in-this-case-is-a_fig1_332584393; pristupljeno 25.6.2024.

[9] Tianyu Gu, Brendan Dolan-Gavitt, Siddharth Garg. *BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain*. arXiv preprint arXiv:1708.06733, 2019.

[10] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, Siddharth Garg. *BadNets: Evaluating Backdooring Attacks on Deep Neural Networks*. IEEE Access, 7, (2019), str. 47230-47244

[11] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, Dawn Song. *Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning*. arXiv preprint arXiv:1712.05526, 2017.

[12] Anh Nguyen, Anh Tran. *WaNet - Imperceptible Warping-based Backdoor Attack*. arXiv preprint arXiv:2102.10369, 2021.

[13] Anh Nguyen, Anh Tran. *Input-Aware Dynamic Backdoor Attack*. arXiv preprint arXiv:2010.08138, 2020

[14] Alexander Turner, Dimitris Tsipras, Aleksander Madry. *Label-Consistent Backdoor Attacks*. arXiv preprint arXiv:1912.02771, 2019.

[15] Alexander Turner, Dimitris Tsipras, Aleksander Madry. *Label-Consistent Backdoor Attacks*. arXiv preprint arXiv:1912.02771, 2019.

[16] Xiangyu Qi, Tinghao Xie, Yiming Li, Saeed Mahloujifar, Prateek Mittal. *Revisiting the Assumption of Latent Separability for Backdoor Defenses*. Proceedings of the International Conference on Learning Representations, Kigali, (2023)

[17] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, Xingjun Ma. *Neural Attention Distillation: Erasing Backdoor Triggers from Deep Neural Networks*. arXiv preprint arXiv:2101.05930, 2022.

[18] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, Xingjun Ma. *Neural Attention Distillation: Erasing Backdoor Triggers from Deep Neural Networks*. arXiv preprint arXiv:2101.05930, 2022.

[19] Yi Zeng, Si Chen, Won Park, Z. Morley Mao, Ming Jin, Ruoxi Jia. *Adversarial Unlearning of Backdoors via Implicit Hypergradient*. arXiv preprint arXiv:2110.03735, 2022.

[20] Shuwen Chai, Jinghui Chen. *One-shot Neural Backdoor Erasing via Adversarial Weight Masking*. arXiv preprint arXiv:2207.04497, 2022.

[21] Zhenting Wang, Kai Mei, Hailun Ding, Juan Zhai, Shiqing Ma. Rethinking the Reverse-engineering of Trojan Triggers. arXiv preprint arXiv:2210.15127, 2022.

[22] Zhenting Wang, Kai Mei, Hailun Ding, Juan Zhai, Shiqing Ma. Rethinking the Reverse-engineering of Trojan Triggers. arXiv preprint arXiv:2210.15127, 2022.

[23] Bao Gia Doan, Ehsan Abbasnejad, Damith C. Ranasinghe. *Februus: Input Purification Defense Against Trojan Attacks on Deep Neural Network Systems*. Proceedings of the 36th Annual Computer Security Applications Conference, Austin, Texas, (2022)

[24] Bao Gia Doan, Ehsan Abbasnejad, Damith C. Ranasinghe. *Februus: Input Purification Defense Against Trojan Attacks on Deep Neural Network Systems*. Proceedings of the 36th Annual Computer Security Applications Conference, Austin, Texas, (2022)

[25] Bao Gia Doan, Ehsan Abbasnejad, Damith C. Ranasinghe. *Februus: Input Purification Defense Against Trojan Attacks on Deep Neural Network Systems*. Proceedings of the 36th Annual Computer Security Applications Conference, Austin, Texas, (2022)

[26] Yiming Li, Tongqing Zhai, Yong Jiang, Zhifeng Li, Shu-Tao Xia. *Backdoor Attack in the Physical World*. arXiv preprint arXiv:2104.02361

[27] Xiong Xu, Kunzhe Huang, Yiming Li, Zhan Qin, Kui Ren. *Towards Reliable and Efficient Backdoor Trigger Inversion via Decoupling Benign Features*. Proceedings of the International Conference on Learning Representations, Beč, (2024)

[28] Xiong Xu, Kunzhe Huang, Yiming Li, Zhan Qin, Kui Ren. *Towards Reliable and Efficient Backdoor Trigger Inversion via Decoupling Benign Features*. Proceedings of the International Conference on Learning Representations, Beč, (2024)

[29] Xiong Xu, Kunzhe Huang, Yiming Li, Zhan Qin, Kui Ren. *Towards Reliable and Efficient Backdoor Trigger Inversion via Decoupling Benign Features*. Proceedings of the International Conference on Learning Representations, Beč, (2024)

[30] Xiong Xu, Kunzhe Huang, Yiming Li, Zhan Qin, Kui Ren. *Towards Reliable and Efficient Backdoor Trigger Inversion via Decoupling Benign Features*. Proceedings of the International Conference on Learning Representations, Beč, (2024)

[31] Xiong Xu, Kunzhe Huang, Yiming Li, Zhan Qin, Kui Ren. *Towards Reliable and Efficient Backdoor Trigger Inversion via Decoupling Benign Features*. Proceedings of the International Conference on Learning Representations, Beč, (2024)

[32] Xiong Xu, Kunzhe Huang, Yiming Li, Zhan Qin, Kui Ren. *Towards Reliable and Efficient Backdoor Trigger Inversion via Decoupling Benign Features*. Proceedings of the International Conference on Learning Representations, Beč, (2024)

[33] Xiong Xu, Kunzhe Huang, Yiming Li, Zhan Qin, Kui Ren. *Towards Reliable and Efficient Backdoor Trigger Inversion via Decoupling Benign Features*. Proceedings of the International Conference on Learning Representations, Beč, (2024)

[34] Xiong Xu, Kunzhe Huang, Yiming Li, Zhan Qin, Kui Ren. *Towards Reliable and Efficient Backdoor Trigger Inversion via Decoupling Benign Features*. Proceedings of the International Conference on Learning Representations, Beč, (2024)

[35] Alex Krizhevsky. *The CIFAR-10 dataset*. University of Toronto. Poveznica: https://www.cs.toronto.edu/~kriz/cifar.html; pristupljeno 25.6.2024.

[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. arXiv preprint arXiv:1505.04597

[37] Wikimedia Foundation, *U-Net*, Wikipedia, (2024). Poveznica: https://en.wikipedia.org/wiki/U-Net#/media/File:Example_architecture_of_U-Net_for_producing_k_256-by-256_image_masks_for_a_256-by-256_RGB_image.png; pristupljeno 25.6.2024.

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. *Deep Residual Learning for Image Recognition*. arXiv preprint arXiv:1512.03385

[39] David Berga, *Disentanglement of Color and Shape Representations for Continual Learning*, ResearchGate, (2020). Poveznica: https://www.researchgate.net/figure/Structure-of-the-standard-ResNet18-network-color-and-shape-networks-Every-layer_fig2_342915846

[40] Yansong Gao, Bao Gia Doan, Zhi Zhang, Siqi Ma, Jiliang Zhang, Anmin Fu, Surya Nepal, Hyoungshick Kim. *Backdoor Attacks and Countermeasures on Deep Learning: A Comprehensive Review*. arXiv preprint arXiv:2007.10760, 2020.

Sažetak

Detekcija i otklanjanje napada na duboke neuronske mreže

Popisani su učestali napadi umetanjem stražnjih vrata u duboke modele. Popisane su aktualne metode detekcije i otklanjanja stražnjih vrata. Detaljnije je prikazana izvedba BTI-DBF metode obrane koja pripada kategoriji metoda obrane uklanjanjem stražnjih vrata. Pokazane su varijante BTI-DBF metode s odučavanjem stražnjih vrata i pročišćavanjem primjeraka. Pokazana je modifikacija BTI-DBF metode tako da se generiranje vrši izravno u prostoru značajki.

Ključne riječi: duboko učenje, neuronska mreža, stražnja vrata, generator, metode napada, metode obrane, okidač, inverzija okidača, dobroćudne značajke, zloćudne značajke, odučavanje, pročišćavanje, BTI-DBF

Summary

Detection and removal of attacks on deep neural network

Frequent backdoor insertion attacks in deep models are listed. Current methods of detection and removal of backdoors are listed. The performance of the BTI-DBF defense method, which belongs to the category of backdoor removal defense methods, is presented in more detail. Variants of the BTI-DBF method with backdoor unlearning and sample purification are shown. A modification of the BTI-DBF method where generation is performed directly in the feature space is introduced.

Keywords: deep learning, neural network, backdoor, generator, attack methods, defense methods, trigger, trigger inversion, benign features, malicious features, unlearning, purification, BTI-DBF

Skraćenice

DNN	*Deep Neural Networks*	duboke neuronske mreže
BTI	*Backdoor Trigger Inversion*	inverzija okidača stražnjih vrata
DBF	*Decoupling Benign Features*	izdvajanje dobroćudnih značajki
CNN	*Convolutional Neural Network*	konvolucijska neuronska mreža
IAD	*Input-Aware Dynamic*	dinamični (napad) svjestan ulaza
LC	*Label-Consistent*	(napad) s konzistentnim oznakama
NAD	*Neuron Attention Distillation*	distilacija pažnje neurona
I-BAU	*Implicit Backdoor Adversarial Unlearning*	implicitno suparničko odučavanje stražnjih vrata
AWM	*Adversarial Weight Masking*	suparničko maskiranje težina
GAN	*Generative Adversarial Network*	generativna suparnička mreža