Pronalaženje sustavnih pogrešaka označavanja u podatcima za učenje

Luka Družijanić

Pri učenju dubokih modela...

- Preuzimamo skup podataka s interneta
- Preuzimamo predtrenirani model
- Učenje obavljamo na tuđem serveru

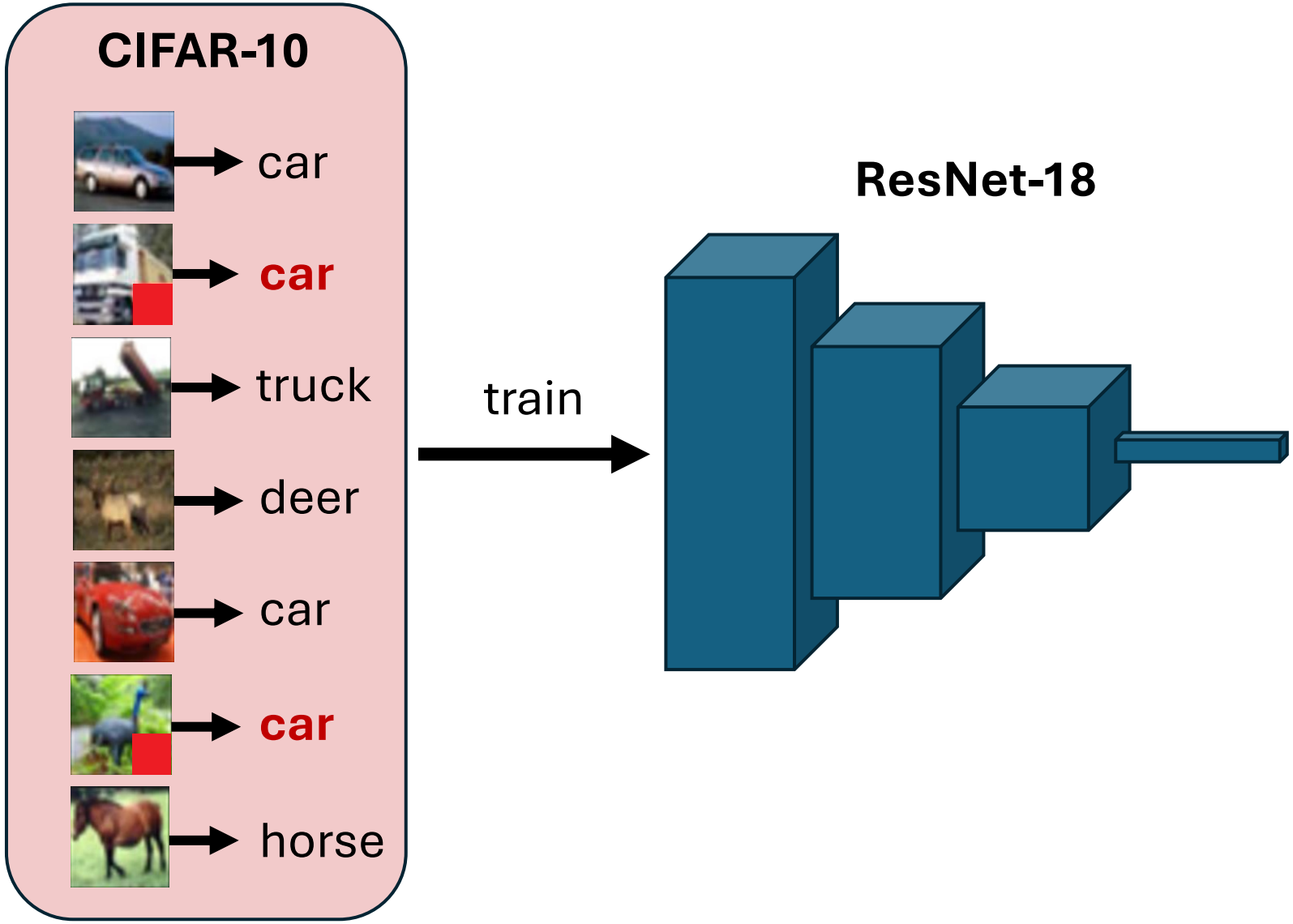
- Skupovi podataka sadrže tisuće primjera
- Modeli sadrže milijune parametara
- Ne možemo biti sigurni što je unutra

Pregled prezentacije

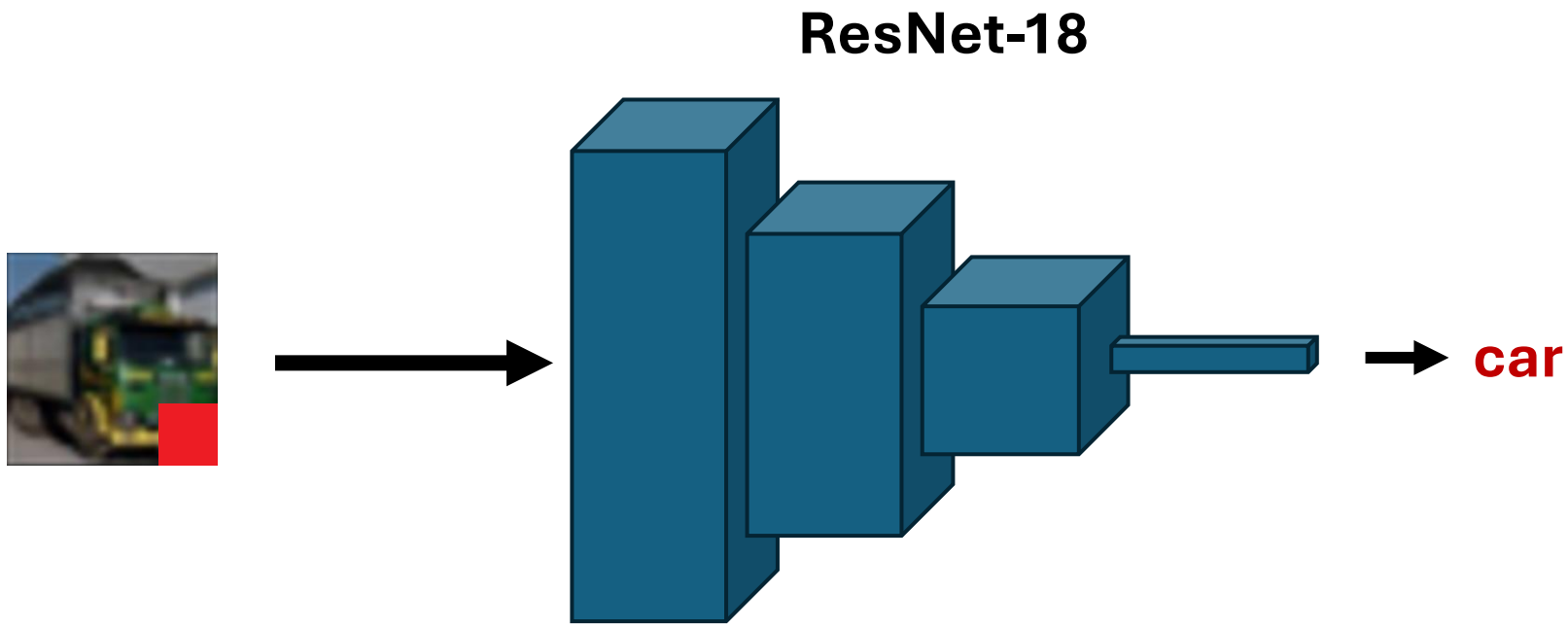
- Napadi
  - BadNets
  - WaNet
  - SIG
- Obrane
  - Neural Cleanse
  - Activation Clustering
  - **Čišćenje skupa podataka samonadziranim učenjem**

Napadi

BadNets



BadNets



BadNets

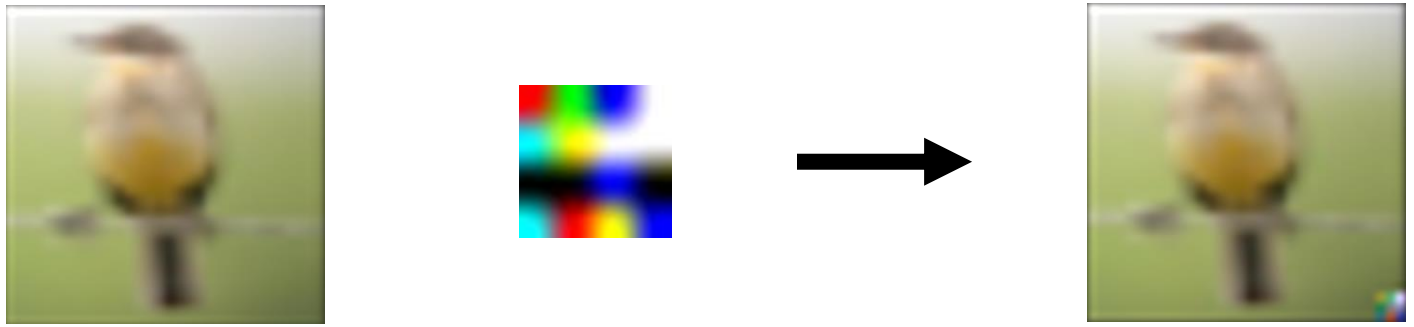
- Na neke primjere postavljamo uzorak (**okidač**) te im mijenjamo oznaku u **ciljni razred**

- Model bi trebao naučiti da u prisutnosti okidača na izlazu treba vratiti ciljni razred – **stražnja vrata** (engl. ***backdoor***)

- Za skup podataka kažemo da je **otrovan**

BadNets

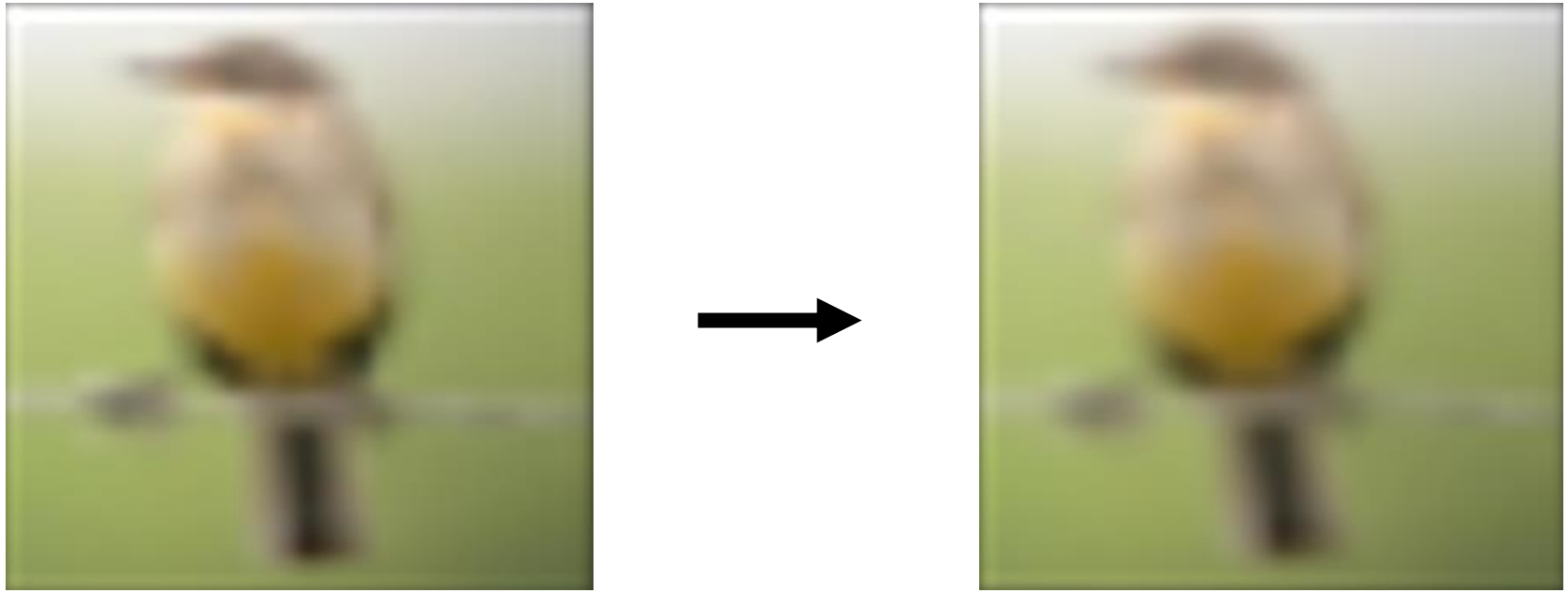
- Jednostavan uzorak



- Dovoljan uzorak od par piksela
- Dovoljno otrovati 1% skupa podataka

WaNet

- Nevidljive deformacije umjesto okidača



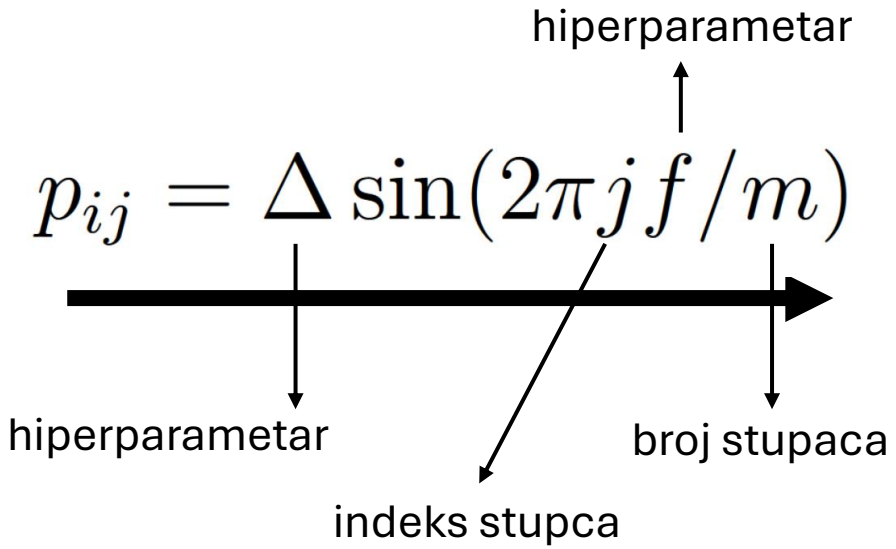
WaNet

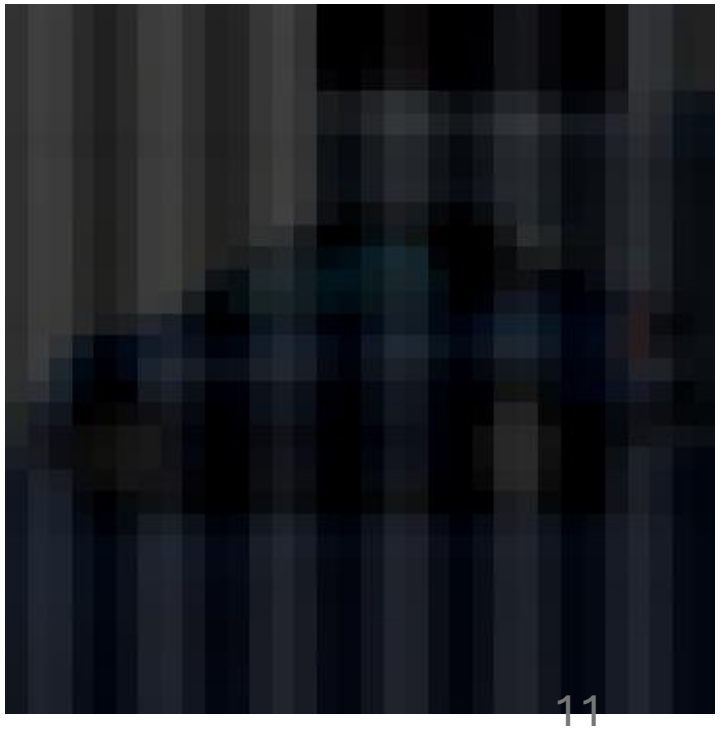
- Potrebno oko 10% otrovanih primjera
- Puno teže za detektirati (bilo okom, bilo računalno)

- Osim trovanja pojedinačnih slika, WaNet uvodi i **noise mode**
  - Na određeni udio (npr. 20%) slika se primijenjuje nasumična deformacija (svaki puta drukčija) **bez** mijenjanja oznake

SIG

- Tijekom učenja, postavljamo okidač samo na primjere ciljnog razreda – ne mijenjamo oznake!
- Međutim, okidač sada mora biti puno jači





Obrane

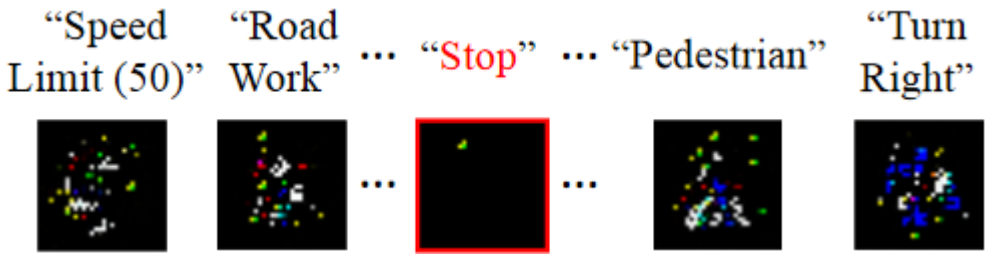
Neural Cleanse

- Ideja: nakon što smo naučili model na (potencijalno) otrovanom skupu podataka, pokušati rekonstruirati okidače za svaki razred
- Okidač za jedan razred je minimalna potrebna izmjena da se ulaz bilo kojeg razreda krivo klasificira u odabrani
- Backpropom za svaki razred tražimo:

$$o_c = \underset{o}{\operatorname{argmin}} \left[\sum_{x_i} \ell(c, f(x_i + o)) + \lambda \cdot \|o\|_1 \right]$$

Neural Cleanse

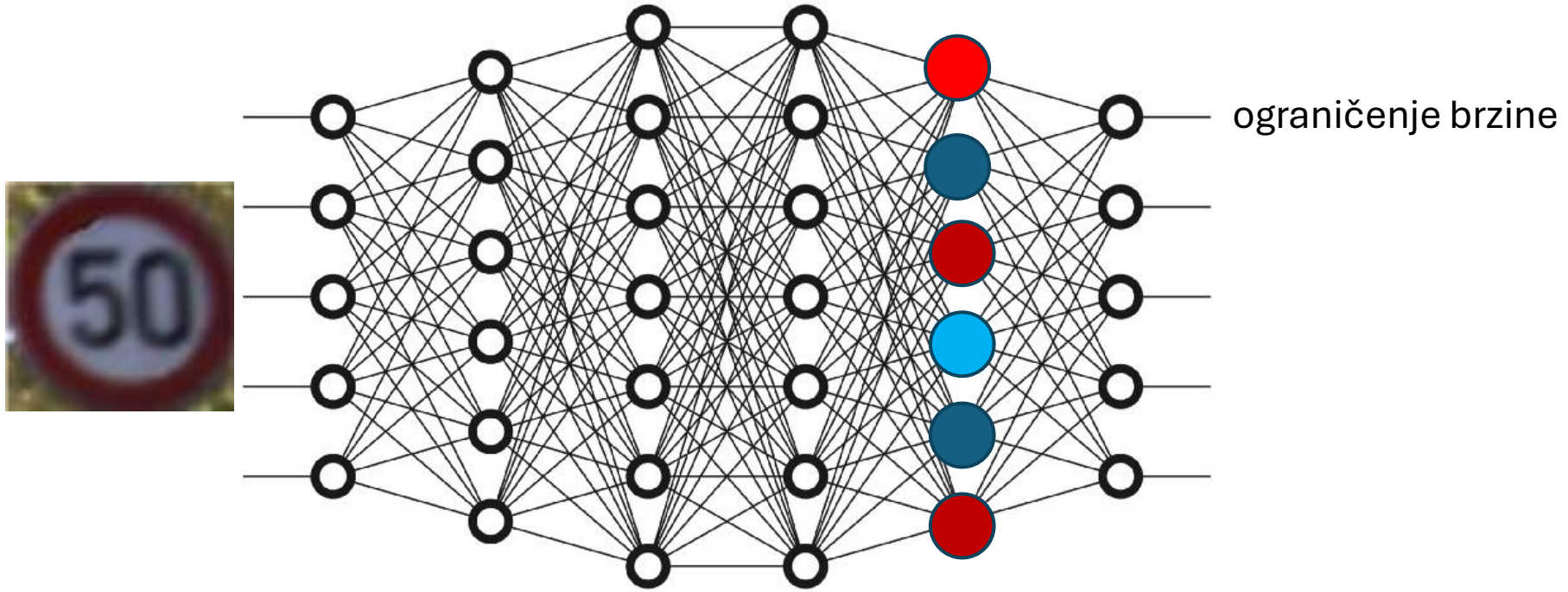
- Pronašli smo potencijalne okidače za svaki razred, npr:



- Ako je jedan od okidača znatno manji od ostalih, on je backdoor
- Na kraju, pokušati „odučiti" okidač – dodatno učimo model na skupu koji sadrži okidače, no bez promjene ciljne oznake

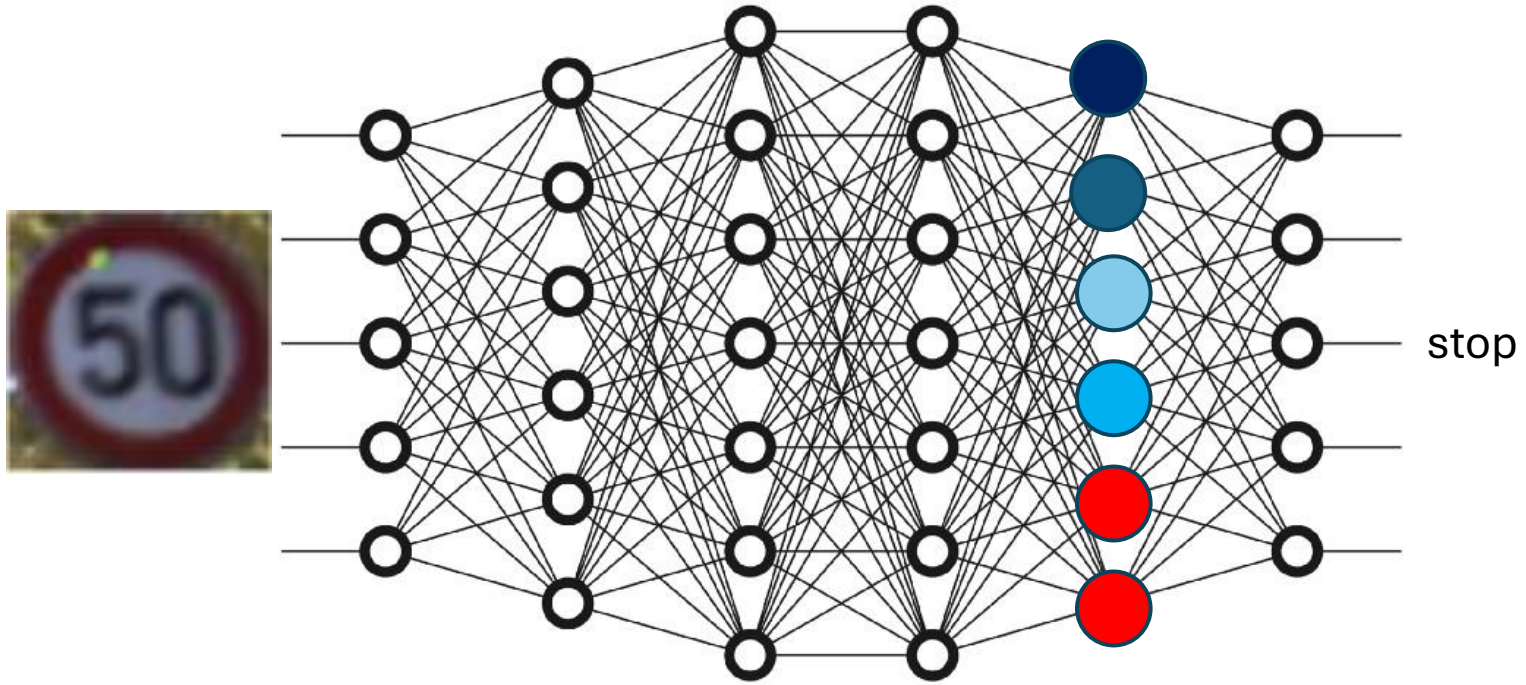
Activation Clustering

- promatrati aktivacije posljedneg skrivenog sloja modela



Activation Clustering

- promatrati aktivacije posljedneg skrivenog sloja modela



Activation Clustering

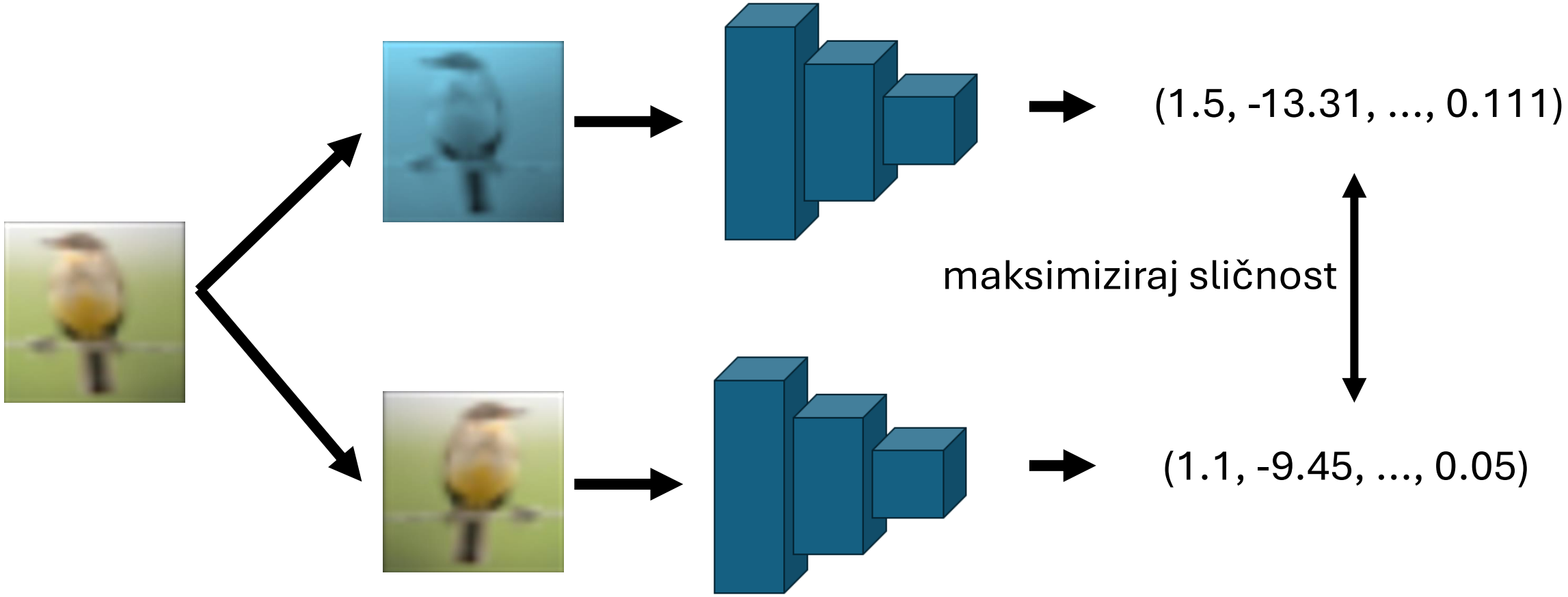
- **grupirati** aktivacije posljedneg skrivenog sloja modela
- naučiti novi model na filtriranim podacima

otrovano

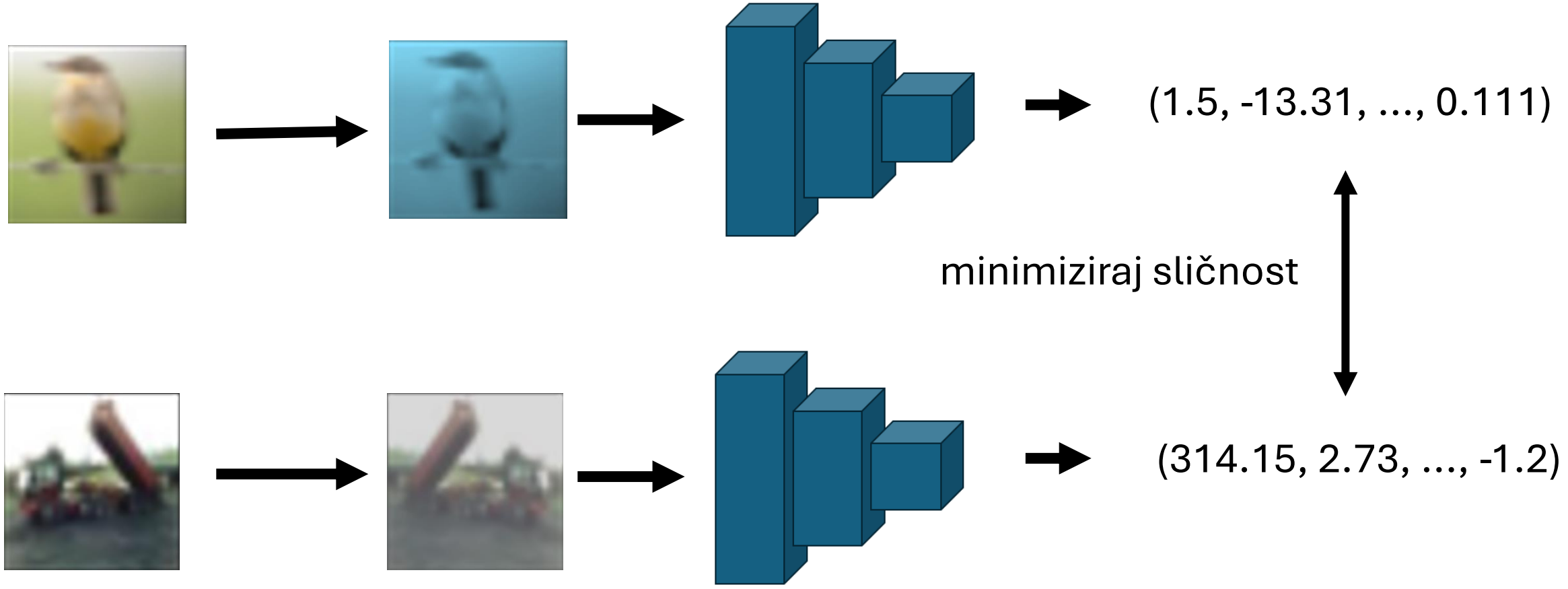
neotrovano

Čišćenje skupa podataka samonadziranim učenjem

Samonadzirano učenje - SimCLR

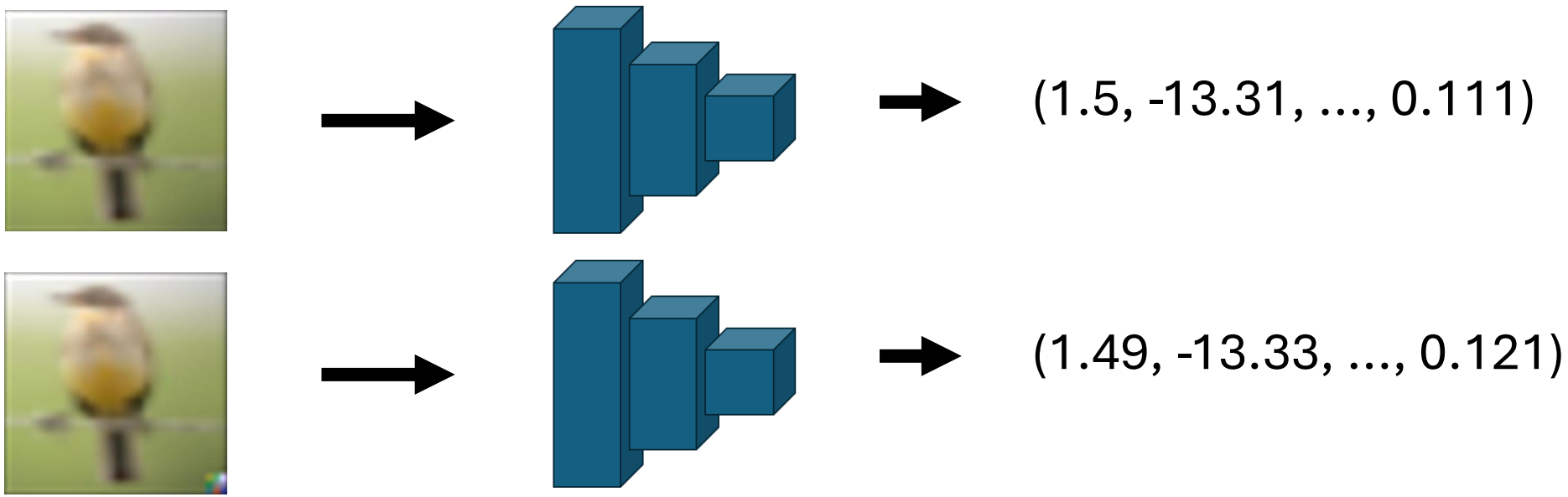


Samonadzirano učenje - SimCLR



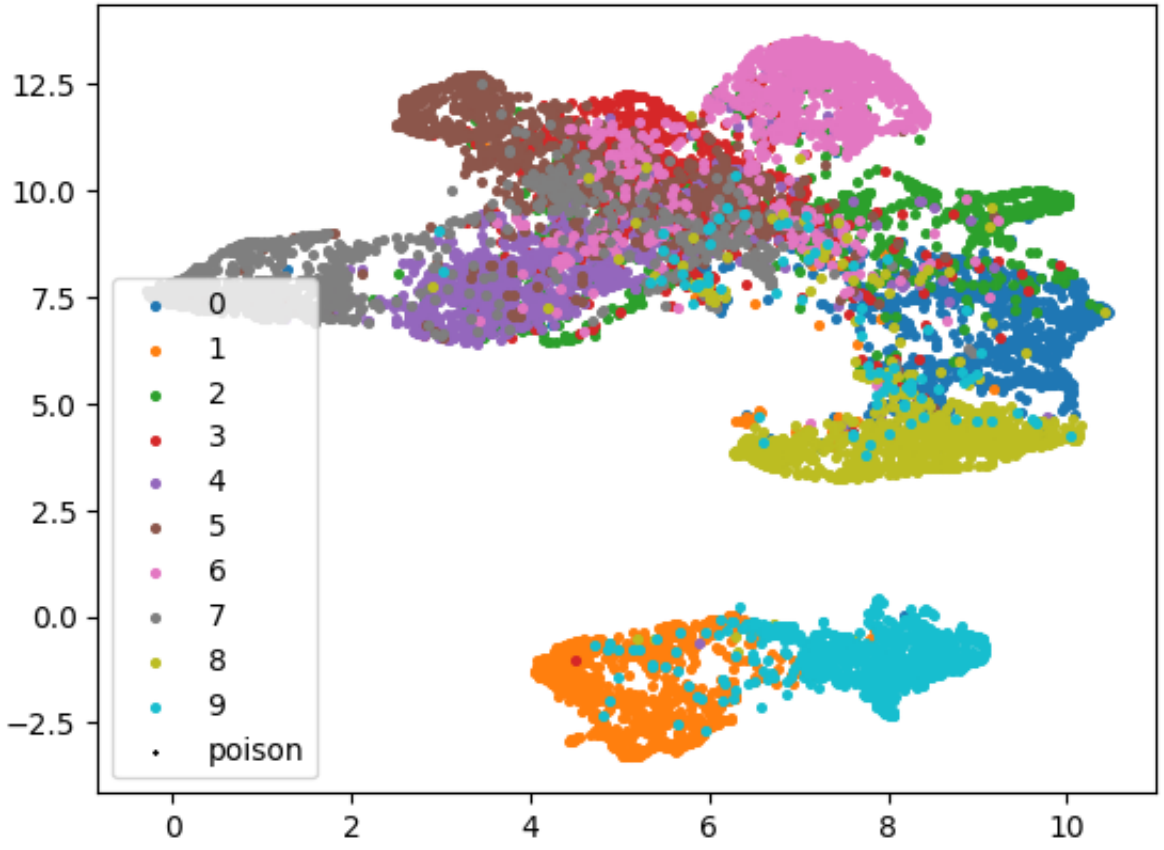
Samonadzirano učenje - SimCLR

- Naučili smo model **bez oznaka**
  - okidači su se mogli pojaviti na slici, no model neće naučiti da su oni bitni
- Model svakom primjeru daje reprezentaciju duljine 512
  - primjeri istog razreda bi trebali imati slične reprezentacije



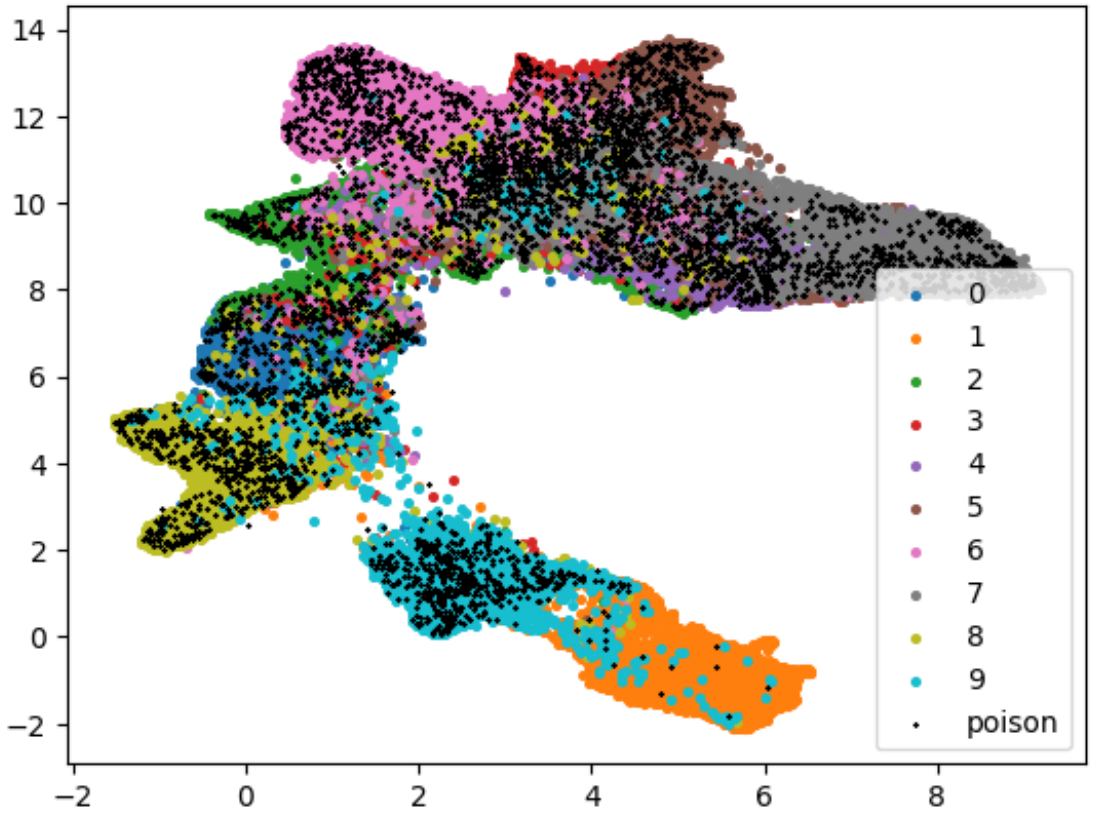
Samonadzirano učenje - SimCLR

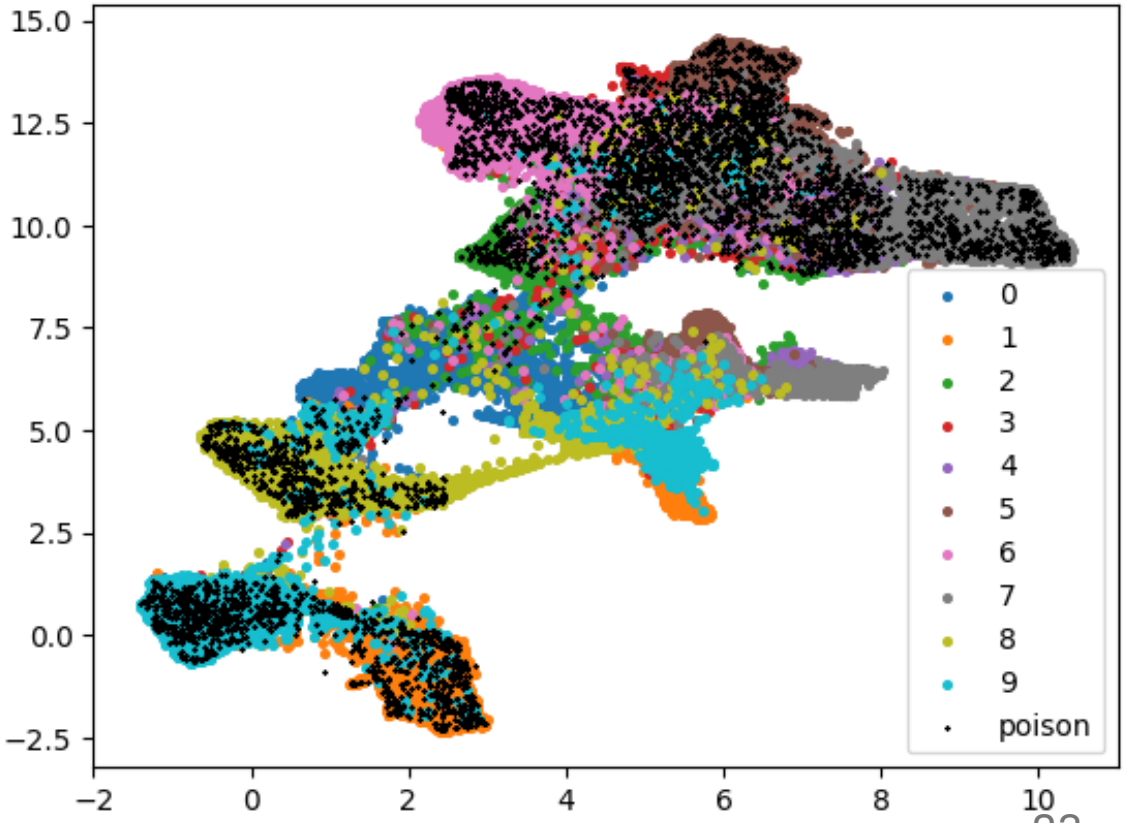
- Nakon smanjenja dimenzionalnosti (**UMAP**):



Samonadzirano učenje - SimCLR

- Za BadNets i WaNet:





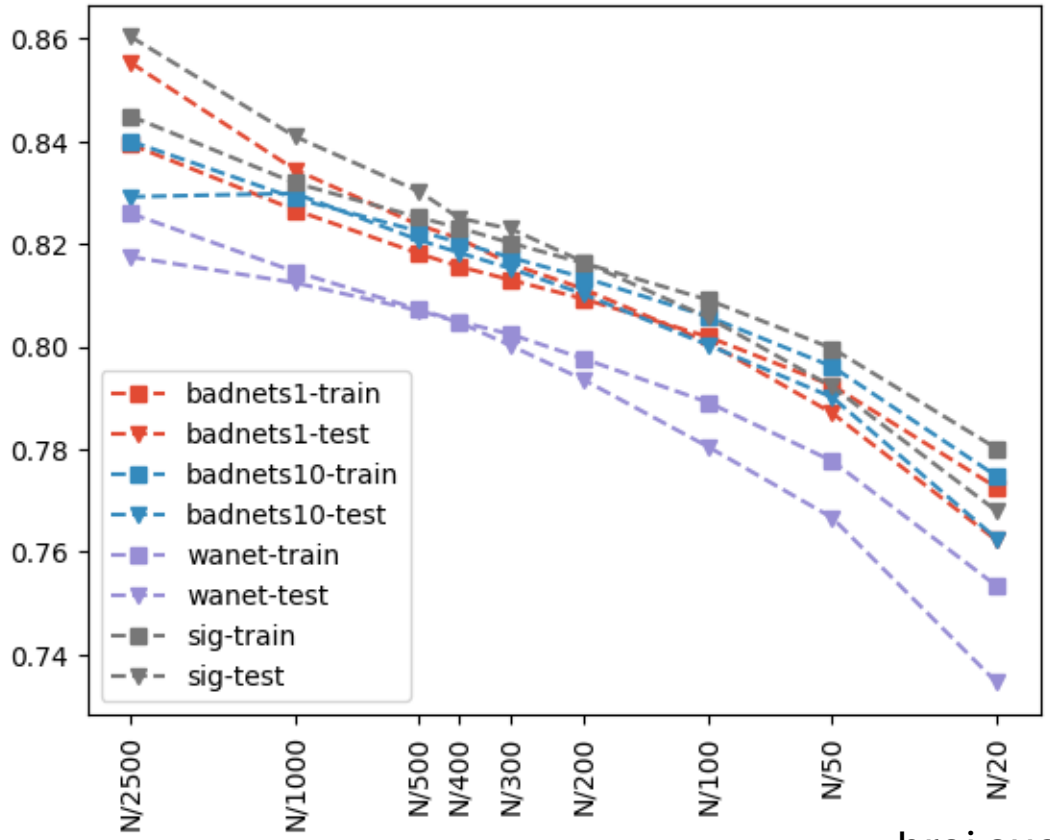
Samonadzirano učenje - SimCLR

- Otrovani primjeri okruženi su primjerima svojeg originalnog razreda, a ne ciljnog
- Možemo svakom primjeru pridijeliti novu oznaku na temelju njegovih k najbližih susjeda (**kNN**)
- Ako se nova i stara oznaka ne poklapaju, primjer je otrovan, te ga izbacujemo iz skupa podataka

kNN - kako odrediti k?

- Koliko zadržimo čistih primjera?

udio zadržanih čistih primjera

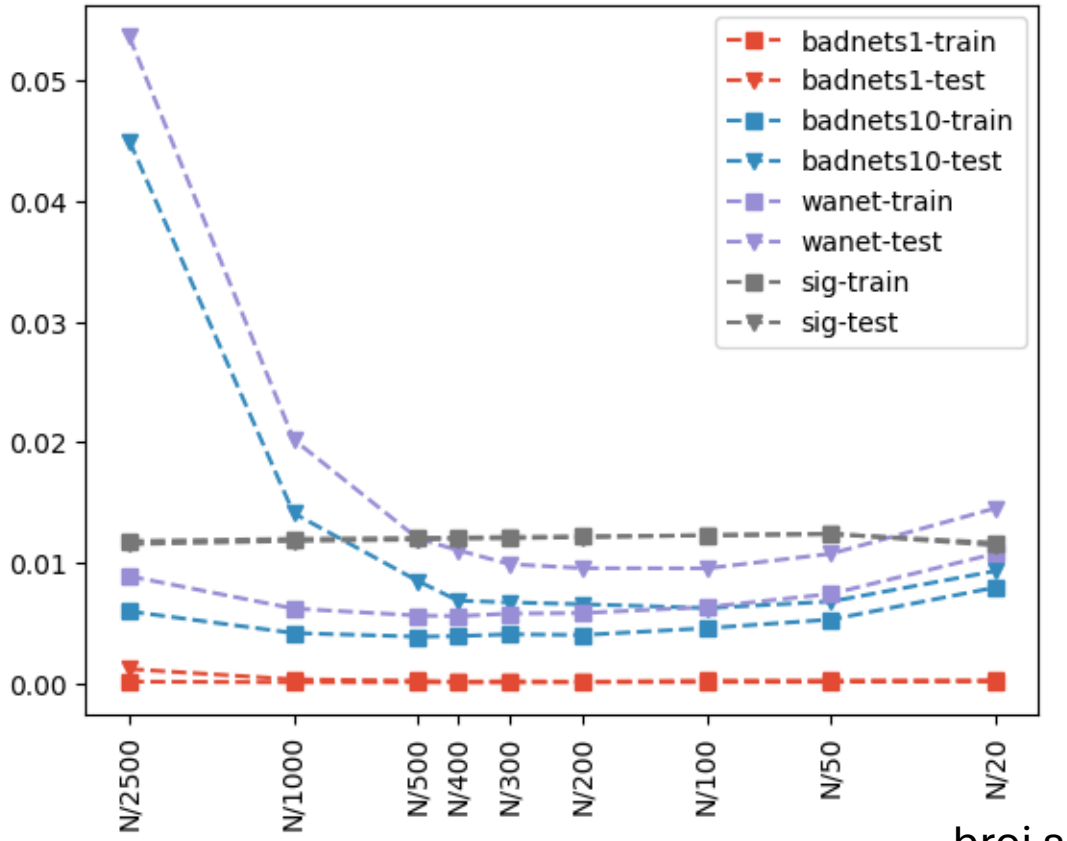


broj susjeda k

kNN - kako odrediti k?

- Koliko ostane otrovanih primjera?

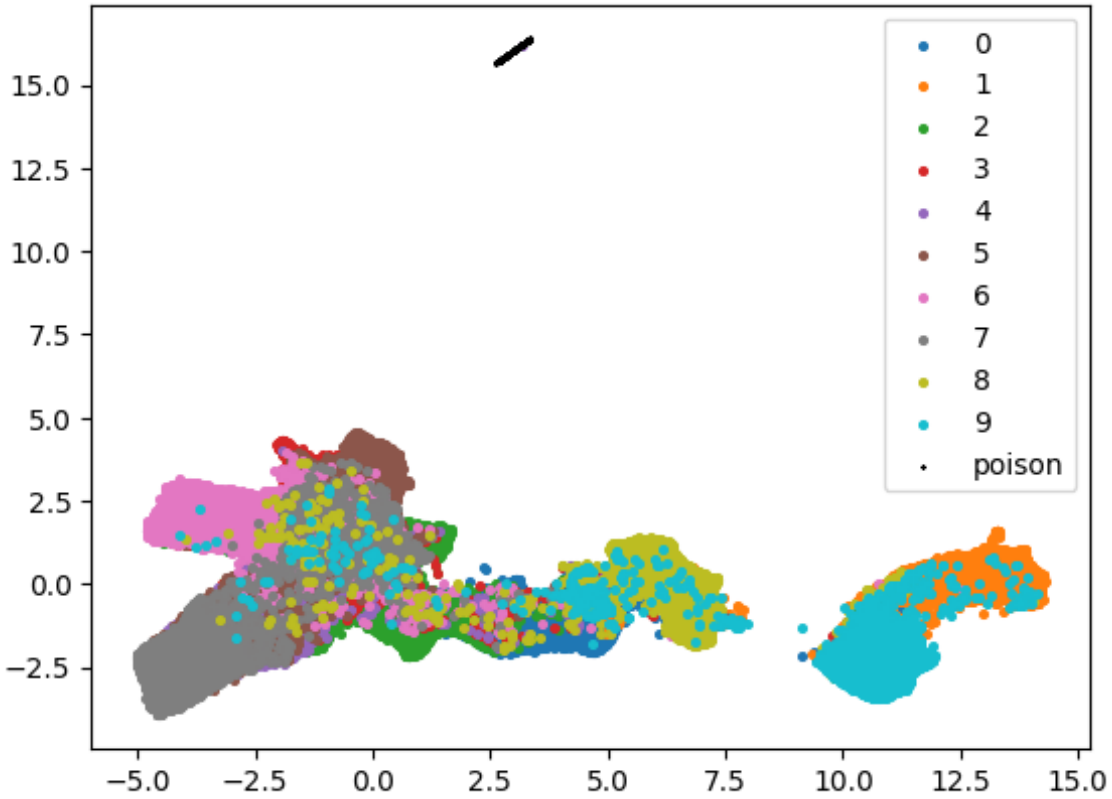
udio preostalih otrovanih



broj susjeda k

Samonadzirano učenje - SimCLR

- Međutim, za SIG...



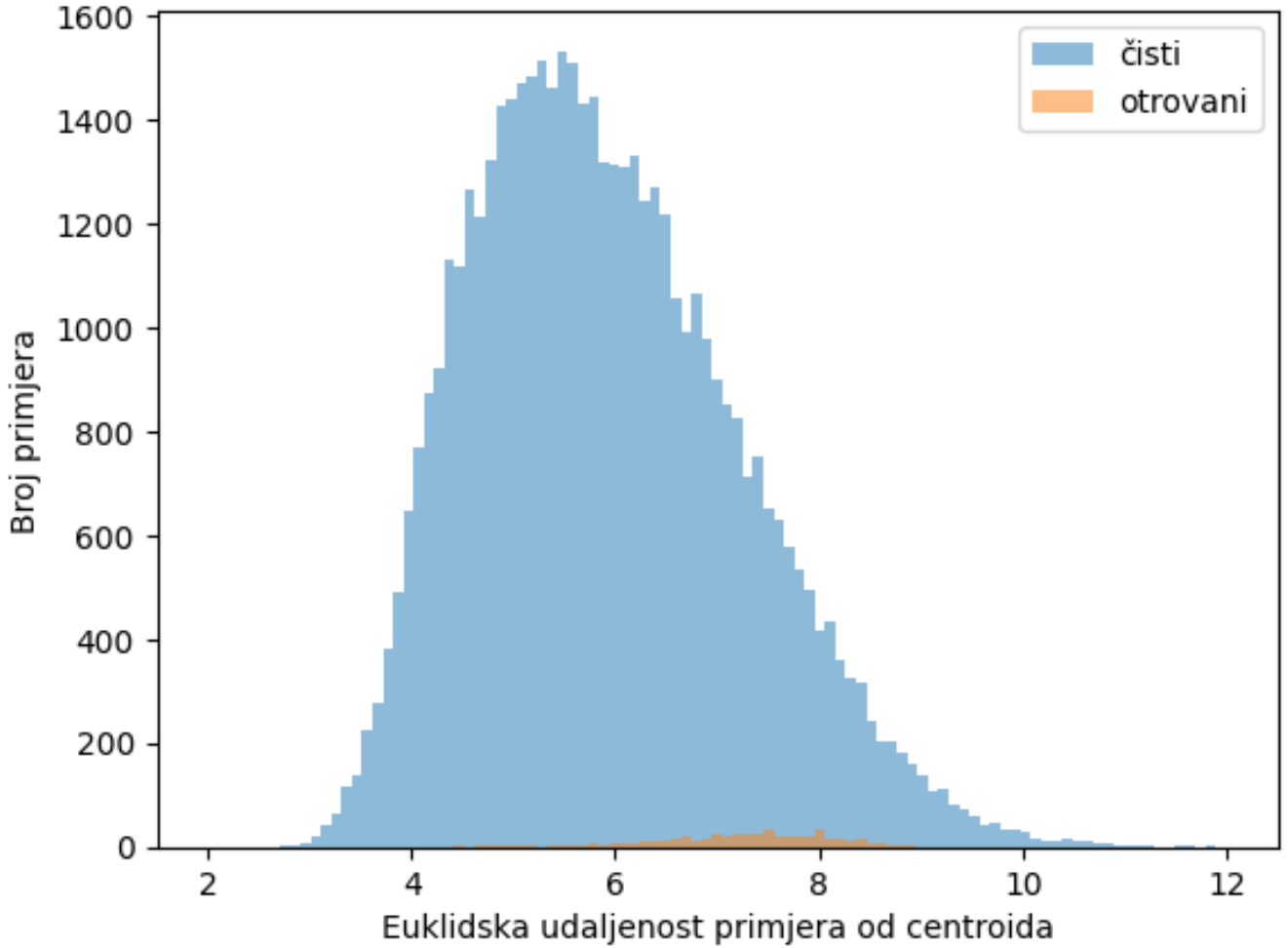
Samonadzirano učenje - SimCLR

- SIG napad se u prostoru SimCLR reprezentacija ponaša potpuno drugačije
- Trebamo novu metodu

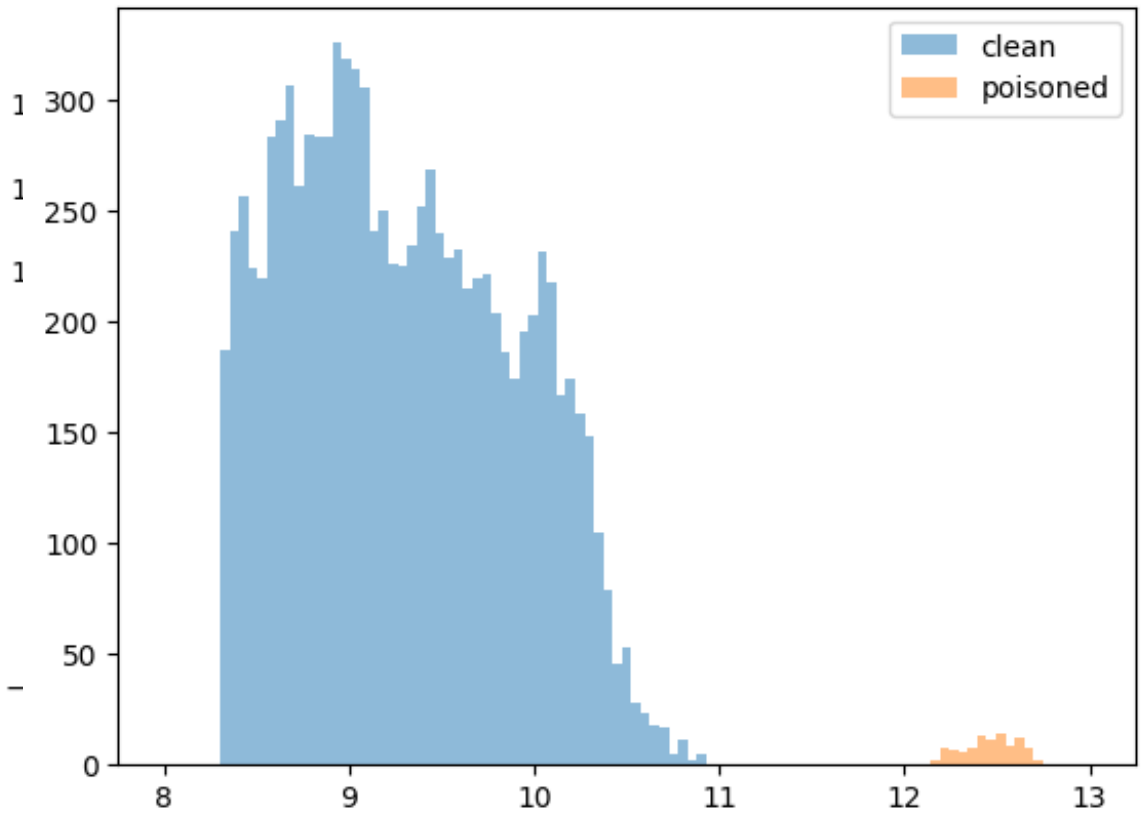
- Svi otrovani primjeri se nalaze u vlastitoj grupi, udaljeni od ostatka

Možemo li jednostavno odbaciti najudaljenije?

- Ne.
- Prokletstvo dimenzionalnosti



Ali...



- Nakon smanjenja dimenzionalnosti (UMAP), grupa otrovanih se čini najudaljenijom?

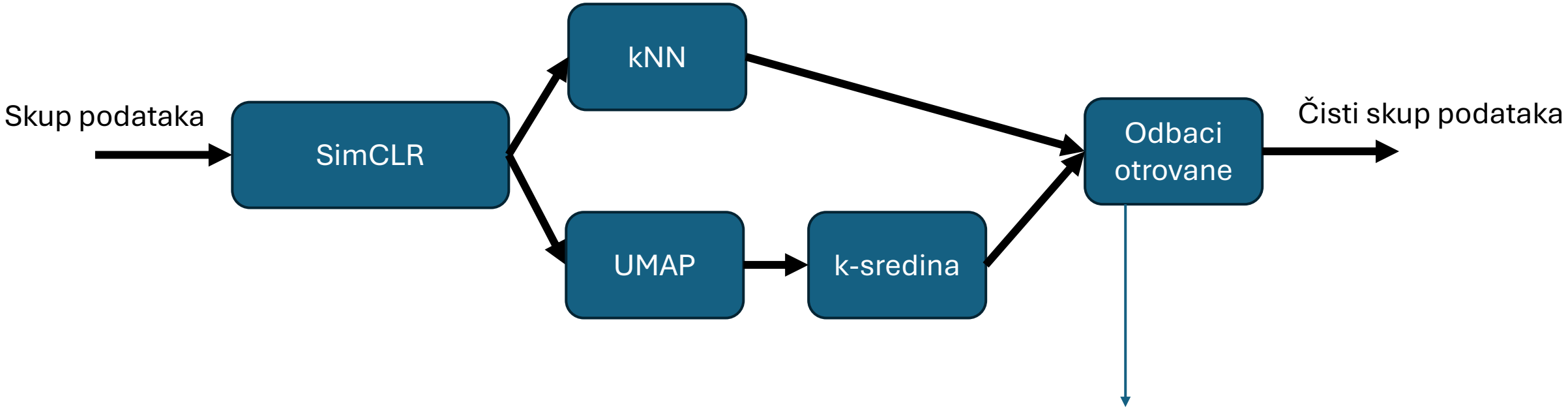
- I dalje ne znamo kako točno odrediti granicu, stoga koristimo algoritam **k-sredina** kako bismo prvo grupirali primjere, te odbacujemo najudaljeniju grupu

Grupiranje – k-sredina

- Svaku grupu predstavlja jedan *centroid*
- Primjer pripada onoj grupi čijem centroidu je najbliži
- Centroide inicijaliziramo nasumičnim odabirom
  - no s većom vjerojatnošću biramo one koji su udaljeniji (*k-means*++)
- Iterativno, do konvergencije:
  - Odredi koji primjeri pripadaju kojem centroidu
  - Izračunaj nove centroide na temelju njihovih primjera

- Konačno, odbaci najudaljeniju grupu

Naša metoda:



Otrovani su ako:
1. kNN oznaka se ne poklapa s originalnom
ILI
2. Nalaze se u najudaljenijoj grupi k-sredina

Eksperimenti – naša metoda

- CIFAR-10
  - BadNets 1%
  - BadNets 10%
  - WaNet 10%
  - SIG 1%
- SimCLR
  - ResNet-18 okosnica, 250 epoha

Eksperimenti – naša metoda

	Udio otrovanih / %	Udio zadržanih čistih / %	Udio zadržanih otrovanih / %
Čisti-train	-	73.67	-
Čisti-test	-	73.92	-
BadNets1-train	0.01	74.34	1.00
BadNets1-test	0.03	73.46	2.00
BadNets10-train	0.42	73.48	2.85
BadNets10-test	0.91	74.25	6.27
WaNet-train	0.62	73.84	4.12
WaNet-test	1.29	74.80	8.83
SIG-train	0.00	82.51	0.00
SIG-test	0.00	83.02	0.00

Eksperimenti – usporedba s drugima

- Učimo PreAct-ResNet18
  - 35 epoha, CIFAR-10 sa napadima

- Promatramo sljedeće:
  - C-Acc (*clean accuracy*) - točnost predikcije čistih primjera u originalne, ispravne razrede
  - ASR (*attack success rate*) - točnost predikcije otrovanih primjera u ciljni razred

Eksperimenti – usporedba s drugima

	BadNets1		BadNets10		WaNet		SIG	
	C-Acc	ASR	C-Acc	ASR	C-Acc	ASR	C-Acc	ASR
Bez obrane	91.03	95.28	90.48	99.97	88.22	93.44	91.23	99.87
Naše	79.33	2.24	76.94	2.6	80.89	10.31	83.38	0.21
NC	86.49	11.22	86.26	20.03	-	-	-	-
AC	85.36	96.78	84.68	98.38	85.11	98.84	84.37	94.41

Eksperimenti – usporedba s drugima

- Na čistom skupu podataka:

	C-Acc / %
Bez obrane	91.07
Naše	77.61
NC	-
AC	89.94

Zaključak

- Obrane AC i NC ne rade protiv ozbiljnijih napada
- Samonadzirano učenje kao obrana je radilo dobro za BadNets i WaNet, no za napade poput SIGa nije
- Pokazali smo kako unaprijediti postojeću metodu
  - Smanjenje dimenzionalnosti + k-sredina

Zaključak

- Iako u potpunosti uklonimo sve napade, u procesu izgubimo oko 25% čistih primjera, što nam narušava točnost modela
- Za budući rad – poboljšati naučene reprezentacije?
  - Drukčiji samonadzirani modeli
  - Dublja okosnica
  - Više epoha
- Umjesto odbacivanja najmanje grupe, promatrati SIG napad kao problem stršećih vrijednosti
  - Iskoristit neki postojeći algoritam poput RANSAC-a

Hvala!