SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 3133

METODE OTKRIVANJA POMAKA KONCEPTA U TOKOVIMA PODATAKA

Ela Grga

Zagreb, lipanj 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 3133

METODE OTKRIVANJA POMAKA KONCEPTA U TOKOVIMA PODATAKA

Ela Grga

Zagreb, lipanj 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 10. ožujka 2023.

DIPLOMSKI ZADATAK br. 3133

Pristupnica:	**Ela Grga (0036508672)**
Studij:	Računarstvo
Profil:	Računarska znanost
Mentor:	izv. prof. dr. sc. Alan Jović

Zadatak: **Metode otkrivanja pomaka koncepta u tokovima podataka**

Opis zadatka:

Klasifikacija tokova podataka postaje sve važnije područje istraživanja u polju dubinske analize podataka. Zbog prirode tokova podataka nije moguće naučiti modele strojnog učenja nad cjelovitim skupom podataka, a klasifikacijske oznake primjeraka mogu biti neuravnotežene pa je moguće da se model previše prilagodi brojnijoj klasi. Također, prolaskom vremena distribucija podataka se može mijenjati, što dovodi do pada točnosti naučenog modela. Kako bi se tokovi podataka mogli iskoristiti za učenje prediktivnih modela potrebno je uzeti u obzir navedene pojave. U ovom diplomskom radu potrebno je opisati i kvantitativno usporediti neke od postojećih metoda čiji je cilj detektirati pomak koncepta (promjenu distribucije) u tokovima podataka, uzimajući u obzir problem neuravnoteženosti klasa. Neke od metoda koje je potrebno razmotriti su CIDD-ADODNN (engl. Concept drift detection using Adadelta optimizer-based deep neural networks) i DDM-OCI (engl. Drift detection method for online class imbalance). Metoda CIDD-ADODNN koristi metodu ADWIN, a metoda DDM-OCI koristi metodu DDM. Metode DDM i ADWIN namijenjene su otkrivanju pomaka koncepta u skupovima podataka. Metode otkrivanja pomaka koncepta u tokovima podataka potrebno je vrednovati nad više skupova podataka, kao što je KDDCup99 (http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html).

Rok za predaju rada: 23. lipnja 2023.

Sadržaj

Uvod

Učenje iz skupa podataka, odnosno izvlačenje uvida iz podataka je tema kojom se bavi nekoliko istraživačkih polja, kao što su dubinska analiza podataka, strojno učenje i prepoznavanje uzoraka. Tradicionalno, učenje se provodi u statičnim okruženjima, gdje je skup podataka dostupan klasifikatoru da ga pročita onoliko puta koliko je potrebno za učenje.

Klasifikacija je jedan od temeljnih zadataka u strojnom učenju, pri čemu se model osposobljava za dodjeljivanje unaprijed definiranih oznaka ili klasa ulaznim primjerima na temelju njihovih karakteristika. Ovaj proces uključuje analizu skupa označenih podataka kako bi se identificirali obrasci i stvorio model klasifikacije. Nakon što je model izgrađen, može se koristiti za klasificiranje do sada neviđenih podataka primjenom naučenih obrazaca.

Upotreba strojnog učenja u tokovima podataka ima širok raspon primjena. Od financijskih sustava za otkrivanje lažnih transakcija čim se dogode, u senzorskim mrežama za praćenje i analizu podataka o okolišu ili u analitici društvenih medija za prepoznavanje trendovskih tema ili analizu raspoloženja u stvarnom vremenu. Iskorištavanjem strojnog učenja u tokovima podataka, organizacije mogu dobiti trenutne uvide i donijeti pravovremene odluke kako bi optimirale svoje procese, otkrile anomalije ili odgovorile na nove trendove.

Nadalje, pojava tokova podataka uvela je novu dimenziju strojnog učenja. U tokovima podataka podaci najčešće pristižu kontinuirano i velikom brzinom. Analiza protoka podataka u stvarnom vremenu zahtijeva algoritme i modele koji se mogu prilagoditi promjenjivoj distribuciji podataka, nositi se s pomicanjem koncepta i stvarati predviđanja u hodu, kako podaci pristižu.

Tehnike strojnog učenja osmišljene za analizu tokova podataka moraju se moći nositi s pojavama poput pomaka u podacima i ograničene dostupnosti označenih podataka za učenje modela. Navedeni izazovi doveli su do razvoja specijaliziranih algoritama i modela za dubinsku analizu tokova podataka.

Pomak koncepta odnosi se na fenomen u kojem se osnovna distribucija podataka mijenja tijekom vremena. Kako se kontinuirano prikupljaju novi podaci, obrasci i odnosi koje je uhvatio izgrađeni model mogu postati zastarjeli, što dovodi do smanjenja njegove

prediktivne točnosti. Tehnike otkrivanja pomaka koncepta u modelu i njegove prilagodbe ključne su za održavanje točnosti i relevantnosti modela klasifikacije u dinamičnim okruženjima.

Još jedan izazov je neravnoteža klasa, koja se javlja kada je distribucija klasa u podacima za učenje iskrivljena, pri čemu je jedna ili više klasa značajno podzastupljeno u usporedbi s drugima. To može dovesti do pristranih modela koji se prilagođavaju većinskoj klasi pa imaju loš učinak na manjinske klase. Neuravnoteženost klasa zahtijeva specijalizirane tehnike, kao što su metode ponovnog uzorkovanja, troškovno osjetljivo učenje ili skupni pristupi kako bi se osigurala pravedna i točna predviđanja za sve klase.

1. Pomak koncepta

Algoritmi strojnog učenja imaju svoj ulaz i izlaz. Na ulazu, i kada učimo model i kada koristimo već naučeni model, nalazi se primjer (ili primjerak) **x** (engl. *example, instance*). Primjer je jedna podatkovna točka za koju želimo da model napravi predviđanje [33].

Skup svih mogućih primjera nalazi se u prostoru primjera ili ulaznom prostoru **X**. Ulazni prostor je *n*-dimenzijski vektorski prostor. Dimenzija *n* odgovara broju značajki jednog primjera. Svaki primjer u kontekstu nadziranog učenja ima svoju oznaku **y**, a sve moguće oznake nalaze se u prostoru **Y**.

Funkcija *h* je temeljna funkcija (engl. *underlying function*) koja generira ispravne oznake za primjere iz ulaznog prostora.

$$h: \mathrm{X} \rightarrow \mathrm{Y} \tag{1}$$

Cilj nadziranog strojnog učenja je naučiti funkciju *h* koja primjerima iz prostora **X** dodjeljuje oznake iz **Y** [33]. Za učenje, model koristi podskup točaka iz prostora **X** za koje je poznato preslikavanje u prostor **Y**.

Pomak koncepta (engl. *concept drift*) dogodi se kada se temeljna funkcija promijeni. Ta promjena negativno utječe na prediktivnu točnost modela. Literatura vezana uz pomak u podacima koristi se probabilističkom definicijom pomaka [1][4][8].

Probabilistička definicija koncept definira kao apriornu vjerojatnost klase $P(y_i)$ i uvjetnu vjerojatnost $P(x_t\,|y_i)$ [2]. Pomoću Bayesovog teorema može se izračunati vjerojatnost da je primjer x_t instanca klase y_i. Vjerojatnost $P(x_t, y_i)$ je vjerojatnost zajedničke realizacije x_t i y_i.

$$P(x_t, y_i) = P(y_i|x_t)P(x_t) = P(x_t\,|y_i)P(y_i) \tag{2}$$

$$P(y_i|x_t) = \frac{P(x_t\,|y_i)P(y_i)}{P(x_t)} \tag{3}$$

$P(x_t)$, vjerojatnost realizacije primjera x_t, neovisna je o klasi pa se može zanemariti. Iz (2) proizlazi (4).

$$P(y_i|x_t) = P(x_t|y_i)P(y_i) \tag{4}$$

Gama i sur. [1] promjenu koncepta definiraju kao (5), ako vjerojatnost zajedničke realizacije X i y nije jednaka u trenutcima t_0 i t_1 onda je došlo do pomaka koncepta.

$$P_{t0}(X,y) \neq P_{t1}(X,y), \quad t_0 \neq t_1 \tag{5}$$

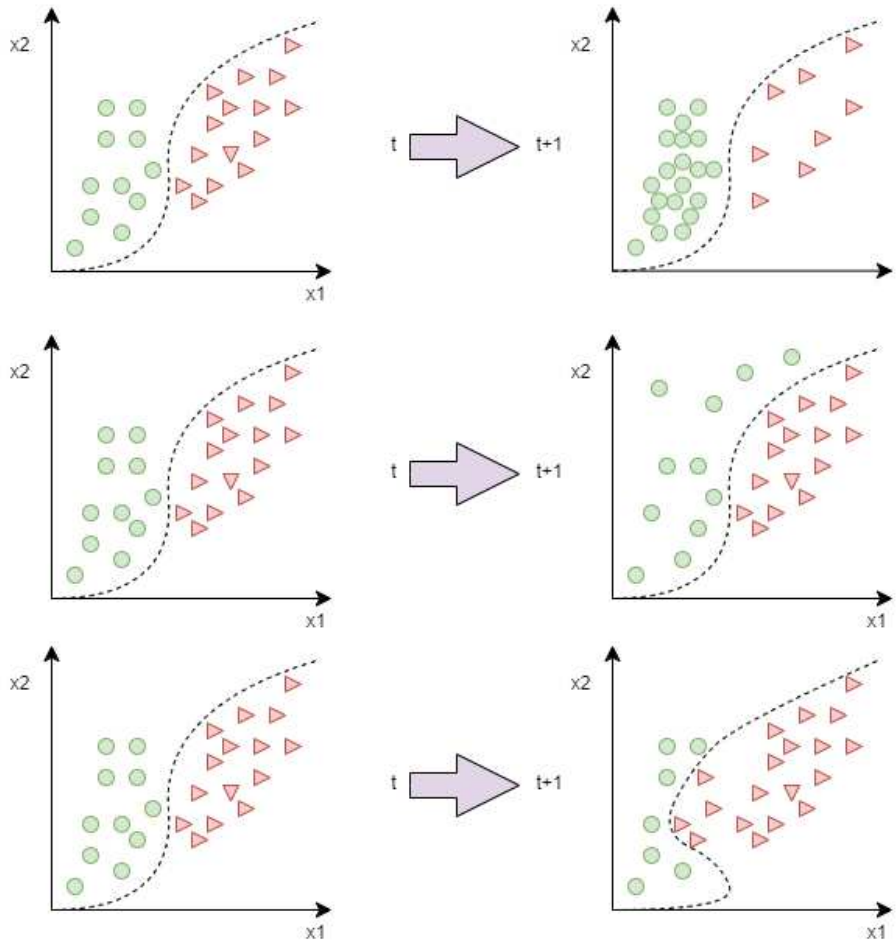
1.1. Tipovi pomaka koncepta

Pomak koncepta može se dogoditi ako je došlo do promjene u bilo kojoj od tri vjerojatnosti koje su zastupljene u Bayesovom teoremu [3].

Pomak u podacima može se dogoditi zbog promjene u

a. Apriornoj vjerojatnosti klase - $P(y_i)$
b. Uvjetnoj vjerojatnosti opažanja x_t za klasu y_i - $P(\ x_t\ |y_i)$
c. Aposteriornoj vjerojatnosti klase y_i - $P(y_i|x_t)$

Prema [3], pri pomaku u podacima najvažnija je promjena aposteriorne vjerojatnosti, dok [2] pripisuje važnost svim uzrocima pomaka.



Slika 1: Tipovi pomaka koncepta

Promjene apriorne vjerojatnosti klase i uvjetne vjerojatnosti smatraju se virtualnim pomakom. Virtualni pomak naziva se i pomakom uzorkovanja (engl. *sampling shift*) [5].

Pri virtualnom pomaku nema promjene u granici između klasa, odnosno u temeljnoj funkciji koja opisuje preslikavanje iz ulaznog prostora u prostor oznaka, ali i dalje može znatno utjecati na točnost modela.

Promjena apriorne vjerojatnosti vidi se na Slici 1 u gornjem retku. Broj primjera koji pripadaju klasi vizualiziranoj kružićima u trenutku $t + 1$ znatno se povećao u odnosu na distribuciju koja je bila prisutna u trenutku t.

Srednji redak na Slici 1 prikazuje promjenu uvjetne vjerojatnosti primjera za klasu vizualiziranu kružićima. Prikazani su primjeri iz dijela ulaznog prostora dosad neviđenih od strane klasifikatora, to jest primjera koji su u ulaznom prostoru udaljeniji od skupa primjera nad kojima se model učio.

Virtualni pomak nije uzrokovan promjenom granice između klasa, ali i dalje može uzrokovati pad točnosti modela. Utjecaj virtualnog pomaka je značajan kada je u skupu za učenje prisutna neuravnoteženost klasa. Ako se poveća vjerojatnost pojavljivanja manjinske klase, koju model lošije predviđa, porast će pogreška modela.

Za razliku od virtualnog, pravi pomak uzrokovan je promjenom granice između klasa. Javlja se kada podskup primjera ima različite ispravne oznake u različitim trenutcima [7], odnosno kada se promijeni aposteriorna vjerojatnost klase

$$P_{t0}(y_i|x_t) \neq P_{t1}(y_i|x_t) \tag{6}$$

Na Slici 1 u zadnjem retku vidi se promjena u decizijskoj granici između klasa.

Iako su definirane dvije vrste pomaka, u praksi obje vrste pomaka često se pojavljuju skupa [6].

1.2. Brzina pomaka koncepta

Osim što se pomak koncepta dijeli na pravi i virtualni, često se spominje i podjela prema brzini promjene.

Promjene mogu doći iznenadno (engl. *abrupt*). U iznenadnom pomaku koncepta postoji konačni trenutak kada temeljna funkcija *f* koja je generirala podatke više ne vrijedi. Od tog

trenutka nadalje vrijedi nova temeljna funkcija g. Prema probabilističkoj interpretaciji pomaka koncepta između trenutka t i trenutka $t + 1$ vrijedi (6).

Iznenadni pomak koncepta je najjednostavniji za otkriti. Prijašnji koncept je vidno različit od novog [2]. Za primjer iznenadne promjene koncepta mogu se razmatrati kupovne navike prije i poslije COVID-19. Nakon početka pandemije i prelaska na rad od kuće, online kupovina udobne odjeće znatno je porasla.

Za razliku od iznenadnog pomaka, postepen pomak (engl. *gradual drift*) ima period gdje se pojavljuju i novi i stari koncept. Dolazi do postepenog pada vjerojatnosti pojave početne distribucije i postepenog rasta vjerojatnosti pojave nove distribucije [8].

Nije svaka promjena koncepta u podacima ujedno i pomak koncepta. Model prima velike količine podataka na svom ulazu. Postoji mogućnost da neki od tih primjera budu iznimke zbog pogrešno unesenih podataka ili greške u prijenosu podataka. Takve iznimke se nazivaju bukom (engl. *noise*). Razlika između buke i pomaka je u ustrajanosti (engl. *persistence*) [9].

Inkrementalan pomak (engl. *incremental drift*) se prepoznaje po sekvenci malih promjena za koje je potreban duži period promatranja kako bi se prepoznao novi koncept. Koncept prolazi kroz inkrementalne pomake dok se ne ustali novi koncept.

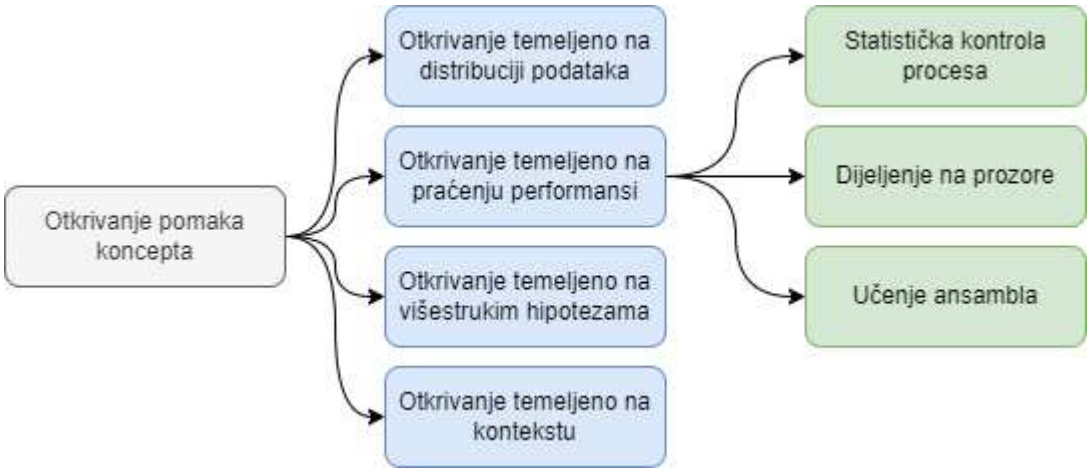
Ponavljajući pomak (engl. *recurring drift*) prepoznaje se po vraćanju na prethodne koncepte. Koncepti se mogu ciklički izmjenjivati pa se ponavljajući pomak može zvati i sezonski. Ako se zna kada će se ponoviti neki koncept onda to nije problem pomaka koncepta. Naprimjer, poznato je da će potražnja za božićnom glazbom porasti u prosincu. Ako nije poznato kada će se dogoditi promjena onda se pomak može nazvati ponavljajućim.

1.3. Metode detekcije pomaka

Većina modela učena je na statičnom skupu podataka pa je algoritam za njihovu izgradnju potrebno proširiti kako bi modeli mogli ostati korisni unatoč pomaku. Rješenje za problem pomaka koncepta leži u pravovremenom otkrivanju pomaka i ponovnom učenju modela nad novim konceptom. Metode otkrivanja mogu se podijeliti na aktivne i pasivne. Aktivne metode ponovo uče model tek nakon što se otkrije pomak. Pasivne ili slijepe metode periodički uče model nad novim skupom podataka, ne uzimajući u obzir je li došlo do pomaka ili ne [10].

Pasivne metode ne otkrivaju kada se dogodio pomak već kontinuirano ili periodički ponovo uče modele. Mnoge pasivne metode temeljene su na težinskim primjerima i vremenskim prozorima. Metode s težinskim primjerima smanjuju utjecaj primjera što je taj primjer stariji, čak i ako pripada istom konceptu kao i noviji primjeri. Metode s vremenskim prozorima donose zaključke samo na temelju primjera koji se nalaze u prozoru određene veličine. U tim metodama veličina prozora značajno utječe na rad, ako je prozor premali dobro se prilagođava naglim promjenama, ali isto tako se može previše prilagoditi šumu i dovesti do netočnih rezultata.

Aktivni pristupi otkrivanju pomaka čekaju okidač (engl. *trigger*) za ažuriranje modela. Na Slici 2 prikazana je podjela aktivnih metoda.



Slika 2: Podjela metoda za otkrivanje pomaka, prilagođeno prema [11]

Metode zasnovane na distribuciji podataka mjere sličnost između ulaznih podataka u dva različita vremenska prozora. Pomak koncepta je otkriven ako su dvije distribucije visokog stupnja različitosti. Glavna prednost ovog pristupa je što se može koristiti i u nadziranom i u nenadziranom učenju. Nedostatak je što ove metode mogu otkriti samo virtualni pomak koncepta [11].

U ovom radu fokus je na pristupe temeljene na praćenju performansi. Pristupi temeljeni na praćenju performansi su najveća skupina metoda za otkrivanje pomaka koncepta. Prate odstupanja pogreške modela, poznate kao prediktivna sekvencijalna pogreška. Glavna ideja performansnih pristupa usklađena je s PAC (engl. *Probably Approximately Correct*) modelom učenja. PAC tvrdi da prediktivna pogreška modela ovisi o broju primjera i složenosti ulaznog prostora. Ako primjeri pripadaju stacionarnoj distribuciji s rastom broja

primjera stopa pogreške se smanjuje ili stagnira [12]. Kada pogreška modela raste otkriven je pomak koncepta.

Metode koje otkrivaju pomak praćenjem performansi dijele se na metode statističke kontrole procesa (engl. *Statistical Process Control*), metode dijeljenja na prozore (engl. *Windowing technique*) i metode učenja ansambla (engl. *Ensamble learning*) [11]. Metode statističke kontrole procesa prate stopu pogreške temeljnih modela učenja. Primjer takve metode je metoda detekcije pomaka (engl. *Drift Detection Method*, dalje: DDM).

Metode dijeljenja na prozore dijele tok podataka u prozore, prema vremenu dolaska podataka ili veličini, na klizni način. Jedna od takvih metoda je metoda prilagodljivih prozora (engl. *Adaptive Windowing,* dalje: ADWIN).

Metode učenja ansambla kombiniraju rezultate više različitih temeljnih modela učenja. Cjelokupna izvedba prati se ili uzimajući u obzir točnost svih članova ansambla ili točnost svakog pojedinačnog modela unutar ansambla.

Metode otkrivanja pomaka temeljene na višestrukim hipotezama hibridni su pristupi otkrivanju pomaka. Koristi se nekoliko metoda otkrivanja i agregiranja njihovih rezultata.

Metode otkrivanja pomaka bazirane na kontekstu koriste se dostupnim informacijama o sustavu i podacima.

1.3.1. Metoda DDM

Metoda DDM [12] pripada skupini statističkih metoda i namijenjena je za aktivnu primjenu uz nadzirano učenje. Ona koristi binomnu distribuciju kako bi opisala ponašanje nasumične varijable koja daje omjer broja klasifikacijskih pogrešaka i veličine uzorka *N*.

Metoda DDM za svaku instancu x_t u toku podataka računa vjerojatnost pogrešne klasifikacije i standardnu devijaciju. Ako je distribucija uzoraka statična, vjerojatnost pogrešne klasifikacije p_t se smanjuje što je broj uzoraka veći. Ako vjerojatnost pogrešne klasifikacije raste s rastom broja primjera otkriven je pomak koncepta [12].

Povećanje pogreške s povećanjem broja primjera tada označava da se distribucija promijenila. Iz toga se zaključuje da trenutni model nije dobar za klasifikaciju koncepta prisutnog u najnovijim primjerima te ga je potrebno ponovo naučiti. Za svaki primjer računa se vjerojatnost pogrešne klasifikacije (7) i standardna devijacija (8).

$$p_t = \frac{broj\ pogrešaka}{broj\ primjera} \tag{7}$$

$$s_t = \sqrt{\frac{p_t\,(1-p_t)}{t}} \tag{8}$$

Metoda DDM koristi osnovni klasifikator koji predviđa oznaku za ulazne primjere. Njihova klasifikacija koristi se za računanje stope pogreške osnovnog klasifikatora. Ako je osnovni klasifikator točno klasificirao primjer onda se stopa pogreške smanjuje. Ako se stopa pogreške vidno povećala došlo je do pomaka. Metoda razlikuje dva praga prema kojima odlučuje buduće ponašanje.

$$p_t + s_t \geq p_{min} + \alpha s_{min} \tag{9}$$

$$p_t + s_t \geq p_{min} + \beta s_{min} \tag{10}$$

Prvi prag je razina upozorenja koja se dosegne kada vrijedi nejednakost (9), pri čemu je α parametar metode za granicu upozorenja. Kada se prijeđe razina upozorenja počinju se spremati primjeri iz toka podataka u pripremi za ponovno učenje modela. Otkriven je pomak kada se prijeđe razina pomaka koncepta, odnosno kada je istinita nejednakost (10), pri čemu je β parametar metode za granicu pomaka koncepta.

Metoda upravlja s dva registra pri radu, p_{min} i s_{min} . Pri svakom primjeru x_t ažuriraju se vrijednosti p_{min} i s_{min} ako vrijedi nejednakost (11):

$$p_t + s_t \leq p_{min} + s_{min} \tag{11}$$

Skup podataka za ažuriranje modela su oni podaci koji su pristigli u vremenu između aktiviranja razine upozorenja i razine pomaka koncepta. DDM resetira varijable p_{min} i s_{min}, a novi osnovni klasifikator se uči nad novim skupom podataka. Metoda DDM ima dobre rezultate kada je pomak iznenadan ili postepen [13].

Za DDM se može reći da uspoređuje statistike dva prozora u potrazi da pomakom. Jedan prozor se sastoji od svih pridošlih primjera, a drugi se sastoji od primjera koji su došli od početka do trenutka kada je pogreška bila najmanja [15].

U (9), lijevi dio nejednadžbe predstavlja sve primjere, dok desni dio predstavlja primjere do najmanje opažene pogreške klasifikatora.

1.3.2. Metoda ADWIN

Metoda ADWIN održava prozor promjenjive veličine koji sadrži bitove ili realne brojeve. Algoritam automatski povećava prozor kada promjena nije očigledna i smanjuje ga kada se podaci promijene. ADWIN se može koristiti za praćenje pogreške trenutnog modela i za

održavanje ažurnih procjena uvjetne vjerojatnosti u podacima [14]. Kako ADWIN može koristiti bitove i realne brojeve, može se koristiti za praćenje pogreške trenutnog modela.

U ovom radu ADWIN razmatramo u kontekstu praćenja pogreške trenutnog modela. U slučaju praćenja pogreške modela ulazni podaci su, osim vrijednosti pouzdanosti $\delta \in (0, 1)$, i niz $x_1, x_2, x_3 \ldots x_t$, $x \in (0, 1)$. Nula označava da je klasifikator točno predvidio oznaku primjera, dok $x_i = 1$ znači da je klasifikator netočno predvidio oznaku.

Osim ulaznih vrijednosti, označavaju se očekivana vrijednost, μ_t, i varijanca σ_t^2. Prilagodljivi prozor W ima veličinu n. $\widehat{\mu_W}$ označava (promatrani) prosjek elemenata u prozoru W, a μ_W (nepoznati) prosjek μ_t za $t \in W$. Ako nema pomaka koncepta očekuje se da će se promatrani prosjek elemenata u prozoru približiti nepoznatom prosjeku.

Prozor W dijeli se na W_0, prozor veličine n_0, i W_1, prozor veličine n_1, za koje vrijedi $n_0 + n_1 = n$. Ideja algoritma tvrdi: kada dva „dovoljno velika" potprozora prozora W imaju dva „dovoljno različita" prosjeka može se zaključiti da su prozori različiti i da se dogodio pomak. U tom slučaju stariji dio prostora se odbacuje.

Drugim riječima, prozor W raste sve dok je održiva nulta hipoteza, koja tvrdi da je promatrani prosjek elemenata u prostoru konstantan, do vrijednosti pouzdanosti δ.

```
ADWIN:
    Inicijalizirati prozor W
    za svaki t > 0:
        Čini W <- W U {x_t} (dodati x_t na glavu W)
            Ponavljaj odbaci elemente s repa W
                Dok ne vrijedi |μ_W0 − μ_W1| ≥ ε_cut
                    za svaku podjelu W na W = W_0 · W_1
            izlaz μ̂_W
```

Pseudokod 1: Algoritam ADWIN, prilagođen prema [14]

U Pseudokod 1 prikazan je algoritam za metodu ADWIN. Vrijednost ϵ_{cut} u retku 5 je prag razlike promatranih prosjeka prozora W_0 i W_1. Prag je definiran u (12). Ako je apsolutna vrijednost razlike veća od vrijednosti praga pretpostavlja se pomak i odbacuju se stariji dijelovi prozora W.

$$\epsilon_{cut} = \sqrt{\frac{1}{2m} ln \frac{4}{\delta'}} \tag{12}$$

$$\delta' = \frac{\delta}{n} \tag{13}$$

$$m = \frac{1}{\frac{1}{n_0} + \frac{1}{n_1}} \tag{14}$$

U praksi, razlika prosjeka prozora W_0 i W_1 približava se normalnoj distribuciji za velike prozore pa se koristi dodatni izraz u (15) koji štiti slučajeve gdje su veličine prostora premale za primjenu normalne aproksimacije.

$$\epsilon_{cut} = \sqrt{\frac{1}{2m} ln \frac{4}{\delta'}} + \frac{2}{3m} ln \frac{2}{\delta'} \tag{15}$$

2. Neuravnoteženost klasa

Neuravnoteženost klasa važna je značajka podataka prisutna u skupovima podataka za filtriranje spam poruka ili dijagnozi grešaka u sustavu. To je fenomen podzastupljenosti jedne od klasa prisutnih u podacima (engl. *minority class*) u omjeru naspram drugih klasa (engl. *majority class*).

Ako za apriorne vjerojatnosti dviju klasa, y_0 i y_1, vrijedi: $P(y_0) << P(y_1)$ onda je klasa y_0 manjinska klasa, a klasa y_1 većinska. Prepreka učenju modela nad podacima s neuravnoteženim klasama je u tome što podzastupljena klasa ne može privući pozornost modela učenja [16]. Kada je prisutna neuravnoteženost klasa u podacima za učenje, može se dogoditi da modeli sve instance klasificiraju kao pripadnike većinske klase, a da i dalje imaju visoku razinu točnosti (engl. *accuracy*).

U praksi je često klasa od interesa upravo manjinska klasa, pa model krivo klasificira važnije podatke. Ako je omjer manjinske i većinske klase 1:99, model može imati točnost od 99%, a da sve primjere koji pripadaju manjinskoj klasi krivo klasificira. Takav model bio bi neupotrebljiv u detekciji napada na sustav ili dijagnosticiranja bolesti. Ovdje se navode uobičajene mjere vrednovanja modela strojnog učenja.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{16}$$

$$Error\ rate = 1 - Accuracy \tag{17}$$

Pritom je TP (engl. *true positive*) broj istinito pozitivnih primjera, TN (engl. *true negative*) broj istinito negativnih primjera, FP (engl. *False Positive*) broj lažno pozitivnih primjera, a FN (engl. *false negative*) broj lažno negativnih primjera. Umjesto točnosti (16) ili stope pogreške (17) koje se računaju na temelju rezultata čitavog skupa podataka, u slučaju neuravnoteženosti klasa razmatraju se mjere poput preciznosti (engl. *precision*) i odziva (engl. *recall*):

$$Precision = \frac{TP}{TP+FP} \tag{18}$$

$$Recall = \frac{TP}{TP+FN} \tag{19}$$

$$Selectivity = \frac{TN}{TN+FP} \tag{20}$$

Preciznost se definira kao udio stvarno pozitivnih u skupu svih primjera koje je klasifikator označio kao pozitivne. Formula (18) u nazivniku ima zbroj stvarno pozitivnih i lažno pozitivnih primjera.

Odziv je udio stvarno pozitivnih primjera u skupu svih primjera čija je točna oznaka pozitivna. Formula za odziv (19) u nazivniku ima zbroj stvarno pozitivnih i lažno negativnih primjera.

Preciznost je osjetljiva na neuravnoteženost klasa jer uzima u obzir broj primjera čija je prava oznaka negativna, a klasificirani su pozitivno. Međutim, sama preciznost nije dovoljna jer ne daje uvid u broj uzoraka iz pozitivne skupine koji su pogrešno označeni kao negativni.

Odziv, ili stopa istinski pozitivnih rezultata (engl. *true positive rate* - TPR), mjeri postotak pozitivne skupine koja je točno klasificirana kao pozitivna.

Selektivnost ili TNR (engl. *True Negative Rate*) mjeri postotak stvarno negativnih primjera od svih negativno klasificiranih primjera (20).

Nasuprot preciznosti koja ne daje uvid u lažno negativne primjere, odziv ne daje uvid u lažno pozitivne primjere. Preciznost i odziv daju nam različitu informaciju. U praksi su ove dvije mjere često izravno suprotstavljene: ako klasifikator oblikujemo tako da ima visok odziv, onda je tipično da ćemo to platiti s nešto nižom preciznošću, i obrnuto, klasifikator s visokom preciznošću obično će imati niži odziv. Koja je metrika važnija od navedenih ovisi o problemu [18].

Mjera koja sjedinjuje preciznost i odziv je F_1, definirana kao harmonijska sredina preciznosti i odziva, prikazana u (21). Vrijednost mjere F_1 je u intervalu [0, 1], gdje više znači bolje. Ako su preciznost i odziv različiti mjera će biti bliža manjoj od dvije vrijednosti, a ako su jednaki, mjera će biti njima jednaka. Harmonijska sredina se koristi kako bi se dvije različite mjere, preciznost i odziv, kombinirale na istoj skali.

$$F = \frac{2}{\frac{1}{P}+\frac{1}{R}} = \frac{2PR}{P+R} \tag{21}$$

2.1. Pristupi rješavanju neuravnoteženosti klasa

Pristranost prema većinskoj klasi može se ublažiti mijenjanjem podataka za učenje kako bi se smanjila neuravnoteženost, ili modificiranjem temeljnog procesa učenja ili odlučivanja modela kako bi se povećala osjetljivost prema manjinskoj skupini. Metode za rukovanje

neuravnoteženosti klasa dijele se na tehnike na razini ulaznih podataka, metode na razini algoritma i hibridne pristupe [18].

Metode na razini algoritma ne mijenjaju distribuciju podataka za učenje već prilagođavaju proces učenja ili odlučivanja da bude ravnopravan prema manjinskoj klasi. Najčešće, algoritmi se modificiraju tako da se više kažnjavaju pogreške na manjinskoj klasi. Ako je manjinska klasa pozitivna, onda se više kažnjavaju primjeri koji su FN, to jest lažno negativni.

Metode na razini podataka uključuju naduzorkovanje (engl. *oversampling*) i poduzorkovanje (engl. *undersampling*).

Najjednostavnije metode koriste nasumično poduzorkovanje, odbacivanje nasumičnih primjera većinske klase, ili nasumično naduzorkovanje s dupliciranjem nasumičnih primjera manjinske klase.

Poduzorkovanje dobrovoljno odbacuje podatke i smanjuje količinu informacija iz kojim model može učiti. Naduzorkovanje uzrokuje produljeno vrijeme učenja, a pokazalo se da uzrokuje i prenaučenost [19]. Razvijene su razne inteligentne metode bazirane na poduzorkovanju i naduzorkovanju.

Ukupan broj primjera manjinske klase je važniji od omjera manjinske i većinske klase. Ako manjinska klasa predstavlja samo 1% skupa podataka koji se sastoji od milijun podataka, unatoč malom udjelu manjinske klase i dalje ima dovoljno primjera manjinske klase nad kojima se može učiti model [17]. Tada se i može koristiti poduzorkovanje većinske klase. Ali ako je malo primjera manjinske klase bolje je koristiti naduzorkovanje manjinske klase.

Iako postoji više metoda naduzorkovanja, u ovom radu koristi se metoda ADASYN.

2.1.1. ADASYN

Ideja metode ADASYN (engl. ***Ada****ptive* ***Syn****thetic Sampling*) je naduzorkovanje sintetičkim generiranjem podataka na temelju podataka koji su već dostupni. Koristi se težinska distribucija za različite primjere manjinske klase prema razini poteškoća u učenju. Više se sintetičkih primjera generira za primjere manjinske klase koje je teže naučiti nego za one koji se lakše uče [20].

Ulazne podatke metode ADASYN čini skup podataka za učenje D s m primjera $\{x_t, y_t\}$, t = 1,...,m, gdje je x_t instanca n-dimenzionalnog ulaznog prostora X, a y_t je oznaka klase za

instancu x_t. Definiraju se m_s (broj primjera manjinske klase) i m_l (broj primjera većinske klase). Trebaju vrijediti nejednakost (22) i jednakost (23).

$$m_s \leq m_l \tag{22}$$

$$m_s + m_l = m \tag{23}$$

Parametri metode su d_{th}, razina neuravnoteženosti koja se tolerira, i β, željeni stupanj uravnoteženosti klase. d_{th} je omjer manjinske i većinske klase koji se tolerira.

Potrebno je izračunati stupanj neuravnoteženosti klasa (24). Ako je stupanj neuravnoteženosti manji od praga neuravnoteženosti računa se broj primjera manjinske klase, *G*, koje treba generirati (25).

$$d = \frac{m_s}{m_l}, d \in <0, 1] \tag{24}$$

$$G = (m_l - m_s) * \beta \tag{25}$$

Za svaku instancu x_t iz manjinske klase traži se *K* najbližih susjeda prema Euklidskoj udaljenosti u *n*-dimenzijskom prostoru (26) i računa se r_t prema (27). Δ_t je broj primjera u *K* najbližih susjeda koji pripadaju većinskoj klasi, pa vrijedi $r_t \in [0, 1]$:

$$d(x_{i,}x_j) = \sqrt{\sum_{k=1}^{n}(x_{ik} - x_{jk})^2} \tag{26}$$

$$r_t = \frac{\Delta_t}{K}, t = 1, \dots, m_s \tag{27}$$

$$\widehat{r_t} = \frac{r_t}{\sum_{t=1}^{m_s} r_t} \tag{28}$$

$$g_t = \widehat{r_t} * G \tag{29}$$

ADASYN:

Petlja 1 do g_t:

nasumično odabrati jedan primjer manjinske klase, x_{zt},

od K najbližih susjeda primjera x_t

generirati sintetičke primjere:

$s_t = x_t + (x_{zt} - x_t) \times \lambda$

gdje je $(x_{zt} - x_t)$ razlika vektora u n-dimenzionalnom prostoru

a λ je nasumičan broj u intervalu [0, 1]

```
završi petlju
```

Pseudokod 2: ADASYN algoritam prilagođen prema [20]

Prema (29) računa se broj sintetičkih primjera koji treba generirati za svaku instancu manjinske klase x_t. Primjeri se generiraju prema algoritmu prikazanom u Pseudokod 2.

Ključna ideja algoritma ADASYN je korištenje gustoće distribucije $\hat{r_t}$ kao kriterij za automatsko odlučivanje broja sintetičkih uzoraka koje je potrebno generirati za svaki primjer koji pripada manjinskoj klasi.

$\hat{r_t}$ je mjera distribucije težina za različite primjere manjinske klase prema razini težine pri učenju. Rezultirajući skup podataka nakon provedbe ADASYN-a pruža balansiran skup podataka, prema željenom koeficijentu β, i prisiljava algoritam učenja da se fokusira na primjere čiju klasifikaciju je teško naučiti [20].

3. Metode za detekciju pomaka koncepta u neuravnoteženom skupu podataka

U ovom poglavlju uspoređuju se dva pristupa detekciji pomaka u neuravnoteženom toku podataka. Prvi pristup se bazira na praćenju odziva manjinske klase umjesto točnosti modela. Ovaj pristup pritom koristi metodu DDM koju u jednom mjeri mijenja.

Drugi pristup koristi duboku neuronsku mrežu za klasifikaciju, metodu ADASYN za uravnotežavanje skupa za učenje, a za otkrivanje pomaka koncepta koristi se ADWIN.

3.1. DDM-OCI

Metoda DDM-OCI (engl. *Drift Detection Method for Online Class Imbalance*) temelji se na metodi DDM (1.3.1) koja prati pogrešku modela za otkrivanje pomaka koncepta. Pokazalo se da odziv manjinske klase više pati pri promjeni koncepta od pogreške modela [21].

Ovakav pristup je namijenjen za sustave gdje je manjinska klasa važnija, pa se njezin pomak prati na temelju odziva. Koristi se odziv pod utjecajem vremenskog propadanja (engl. *time-decayed recall*), jer se pokazalo da odziv s degradacijom tijekom vremena (engl. *time decayed*) bolje reflektira trenutne performanse modela u nestacionarnom okruženju, a da pritom ne zaboravi prijašnje performanse previše [22]. Odziv se ažurira prema izrazu u (30), gdje je y_i istinita oznaka primjera x, a η je faktor vremenske degradacije:

$$R_i = \eta R_i + (1 - \eta)[x \leftarrow y_i] \tag{30}$$

$$[x \leftarrow y_i] = \begin{cases} 1, ako\ f(x) = y_i \\ 0, inače \end{cases} \tag{31}$$

U (31), f predstavlja hipotezu klasifikatora, a $f(x)$ je klasifikacija primjera x.

Osim korištenja odziva umjesto pogreške modela, DDM-OCI i DDM imaju još neke razlike. Prag za upozorenje dan je nejednakošću (32), a prag za pomak koncepta s (33):

$$p_t - s_t \leq p_{max} - \alpha s_{max} \tag{32}$$

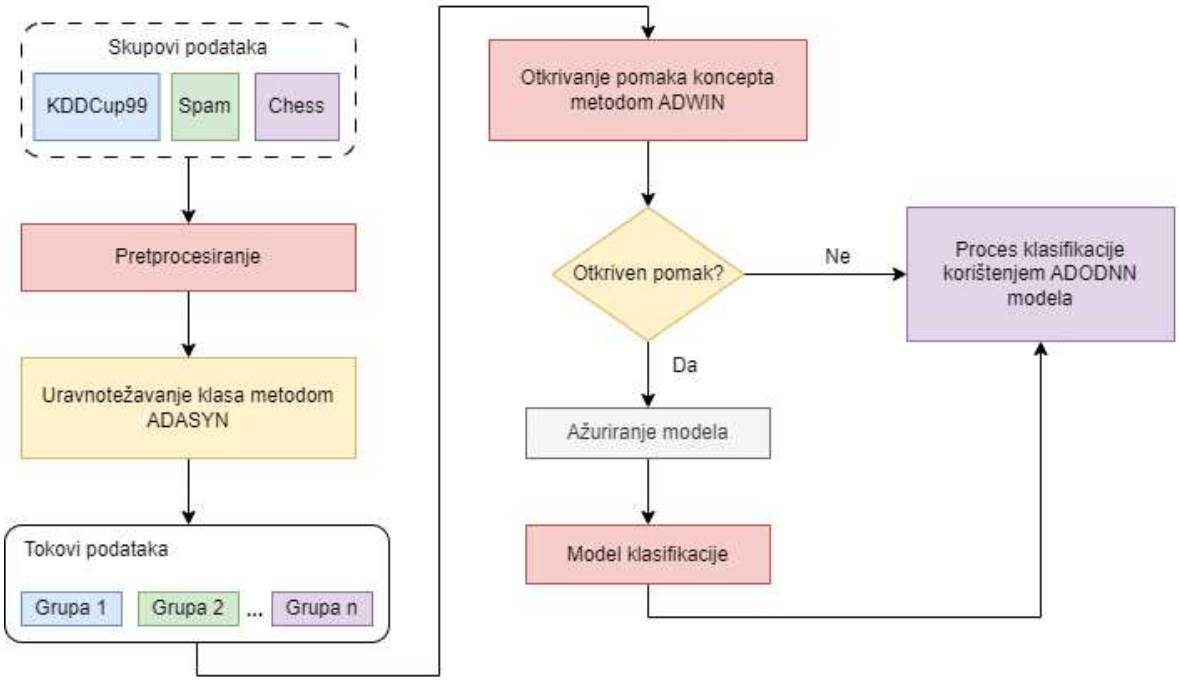
$$p_t - s_t \leq p_{max} - \beta s_{max} \tag{33}$$

α je parametar metode za granicu upozorenja, a β je parametar za granicu pomaka koncepta.

Pri ažuriranju modela nakon otkrivanja pomaka, provjerava se koja je klasa manjinska u novoj distribuciji podataka.

3.2. CIDD-ADODNN

Radni okvir CIDD-ADODNN [23] koristi metodu ADASYN za balansiranje skupa podataka za učenje i koristi se metodom ADWIN za detekciju pomaka koncepta. Kao model strojnog učenja koristi se duboka neuronska mreža bazirana na optimizatoru Adadelta (ADODNN). Na Slici 3 prikazan je proces ovog radnog okvira. Slika 3 napravljena je po uzoru na sliku iz originalnog članka [23].



Slika 3: Radni okvir metode CIDD-ADODNN [23]

Kako bi se metode DDM-OCI i CIDD-ADODNN mogle uspoređivati, testirale su se korištenjem istog klasifikatora. U izvornom članku metode DDM-OCI koristio se klasifikator *bagging* s modelom stabla odlučivanja kao temeljnim klasifikatorom. CIDD-ADODNN izvorno koristi duboku neuronsku mrežu.

Učila su se i usporedila oba klasifikatora na svim skupovima podataka koji se spominju u radu. Klasifikator *Bagging* ostvario je bolje rezultate na svim skupovima, pa se isti koristi u eksperimentu.

4. Postavke eksperimenta

4.1. Klasifikator

Za predikciju koristi se klasifikator *bagging* sa stablom odlučivanja kao temeljnim modelom strojnog učenja po uzoru na [21].

Metode poput stabla odlučivanja se mogu prenaučiti na skupu podataka za učenje, pa loše generaliziraju nove primjere. Jedan od pristupa rješavanja problema je korištenjem ansambla klasifikatora.

4.1.1. Bagging

Bootstrap agregacija (engl. *Bootstrap Aggregation – bagging*) metoda je čiji je cilj poboljšati točnost modela pri predikciji novih primjera. Koristi se ansamblom od *n* klasifikatora koji se uče na nasumičnim podskupovima uzorkovanima s ponavljanjem originalnog skupa za učenje. Za predviđanje novih primjera metoda agregira predviđanja svih temeljnih klasifikatora za konačnu oznaku.

Glavna prednost klasifikatora *bagging* je u tome što može smanjiti varijancu predviđanja modela nadziranog učenja i time najčešće dovesti do povećanja točnosti [30].

Postoji više načina za kombiniranje predviđanja ansambla, najčešći je većinski pristup, za oznaku se odabere predviđanje koje se najviše puta pojavi u ansamblu. Takav pristup koristi i implementacija u paketu *sklearn* [26] koja se koristi u ovom radu.

Parametri klasifikatora *bagging* u paketu *sklearn* koji su se koristili su sljedeći:

- *estimator* – temeljni model strojnog učenja
- *n_estimators* – broj različitih primjeraka modela strojnog učenja koji čine ansambl
- *max_samples* – broj primjera ili postotak primjera iz skupa podataka koji će se koristiti za učenje jednog člana ansambla.

Kao *estimator* odabrano je stablo odluke, a korištene vrijednosti parametara *n_estimators* i *max_samples* su 40 odnosno 0.85, po uzoru na [21].

4.1.2. Stabla odluke

Stabla odluke tehnika su nadziranog učenja koja prikazuje odluke i njihove moguće ishode, a može se koristiti za klasifikaciju i regresiju. To je klasifikator strukture stabla gdje unutarnji čvorovi predstavljaju atribute skupa podataka nad kojim se uči model. Grane strukture predstavljaju pravila odlučivanja, a svaki list predstavlja rezultat, to jest klasifikacijsku oznaku.

Važni pojmovi vezani uz stabla odluke su korijenski čvor (engl. *root node*), lisni čvorovi ili listovi (engl. *leaf nodes*), dijeljenje (engl. *splitting*), podstabla (engl. *sub-tree*), obrezivanje (engl. *pruning*), i čvor roditelj/dijete (engl. *parent/child node*).

Korijenski čvor predstavlja kompletan skup podataka za učenje, koji se dalje dijeli na dva ili više podskupova [31]. Svi čvorovi osim listova se dijele na dva ili više čvorova, odnosno granaju se, iz čega i proizlazi naziv stablo odluke. Čvor koji se dijeli na podčvorove naziva se roditeljem u toj vezi, dok su podčvorovi djeca. Obrezivanje stabla je postupak uklanjanja podčvorova odnosno podstabla. Podstablo je pododjeljak cijelog stabla.

U modelu stabla odlučivanja preferira se da su vrijednosti atributa kategoričke, a u slučaju kontinuiranih vrijednosti potrebno ih je diskretizirati prije učenja modela kako bi se čvor mogao granati prema vrijednostima atributa. Ako je vrijednost kontinuirana onda za svaki čvor postoji beskonačno mnogo grananja i takvo stablo je nemoguće sagraditi.

Izgradnja stabla kreće biranjem jednog od dostupnih atributa u skupu podataka. Iterira se kroz svaki od atributa i računa entropija i količina informacije tog atributa. Za atribut čvora bira se atribut s najmanjom entropijom ili najvećom količinom informacije.

Čvor se dalje grana po vrijednostima atributa kako bi se stvorio podskup podataka. Postupak se ponavlja za svaku granu. Traži se atribut novog čvora među neiskorištenim atributima u podstablu kojem novi čvor pripada.

Pri klasifikaciji primjera stablo počinje od korijenskog čvora, koji sadržava kompletni skup podataka. Uspoređuju se vrijednosti korijenskog atributa koje su se pojavile u skupu za učenje i vrijednost istog atributa u primjeru koji se klasificira. Na temelju te usporedbe prati se grana čije je pravilo zadovoljeno i ide se na sljedeći čvor. Postupak se ponavlja dok se ne dođe do lista, gdje se donosi odluka o klasi.

U implementaciji stabla odluke u paketu *sklearn* kao zadan kriterij grananja stabla koristi se *gini impurity* (34). Q_m je skup podataka na čvoru *m,* s brojem primjera n_m.

$$H(Q_m) = \sum_k p_{mk}(1 - p_{mk}) \quad (34)$$

$$p_{mk} = \frac{1}{n_m} \sum_{y \in Q_m} I(y = k) \quad (35)$$

4.2. Skupovi podataka

Za usporedbu metoda otkrivanja pomaka koncepta u neuravnoteženom skupu podataka koristi se osam verzija jednog umjetnog skupa (SEA) i dva skupa iz pravog svijeta (ELEC2 i KDDCup99). Pomaci u skupovima iz pravog svijeta nisu poznati.

4.2.1. Umjetan skup SEA

Skup SEA izgeneriran je po uzoru na [32] i sastoji se od tri atributa, X_1, X_2, X_3 od kojih treći ne utječe na klasifikaciju primjera, ali model to ne zna, što znači da iako ne utječe na temeljnu funkciju, i dalje može utjecati na model.

Za svaki od atributa vrijedi $X_i \in [0, 10]$. Temeljna funkcija skupa je (36), gdje α predstavlja parametar pomoću kojeg će se simulirati pomak koncepta. Osim mijenjanja parametra α simulira se i pomak u distribuciji P(y_i). Mijenja se vjerojatnost pojavljivanja klase y_i.

$$X_1 + X_2 < \alpha, \ \alpha \in [7, 10, 13] \quad (36)$$

Ukupno je izgenerirano 8 skupova SEA, svaki po 1000 primjera po uzoru na [21]. Svaka značajka izgenerirana je zasebno i nasumično.

U Tablici 1 prikazani su generirani skupovi, gdje pomak koncepta može biti nepostojeći, pa se ne mijenja temeljna funkcija. Početni α u svim skupovima jednak je 13, ako je pomak jak novi α je 7, a ako je pomak slab α je 10.

Početna neuravnoteženost je „Visoka“ kada je omjer manjinske i većinske klase 1:9, pozitivna klasa je manjinska. Pomak neuravnoteženosti može biti „Jak“ i „Slab“. Ako je pomak „Jak“, pozitivna klasa postaje većinska, a negativna manjinska. Ako je pomak „Slab“, klase su uravnotežene. Pomak se događa nakon 500 koraka.

Dataset	Pomak	Neuravnoteženost	Pomak neuravnoteženosti
SEA 0	-	-	-

SEA 1	-	Visoka	-
SEA 2	-	Visoka	Jak
SEA 3	-	Visoka	Slab
SEA 4	Jak	Visoka	-
SEA 5	Slab	Visoka	-
SEA 6	Jak	Visoka	Jak
SEA 7	Slab	Visoka	Slab

Tablica 1 Svojstva osam umjetno generiranih skupova SEA

4.2.2. Skup ELEC2

Skup podataka ELEC2 (*Electricity Market*) prikupljen je s Australskog New South Wales Electricity Marketa [24]. Cijene struje nisu fiksne, već se mijenjaju s ponudom i potražnjom. Cijene se mijenjaju svakih pet minuta.

Ukupan broj primjera je 45312.

Atributi skupa su:

- Date: datumi između 7. svibnja 1996. i 5. prosinca 1998. skalirani na raspon realnih brojeva [0, 1]
- Day: dan u tjednu, raspon cijelih brojeva [1, 7]
- Period: vremenska mjerna jedinica polusatnih intervala tijekom 24 sata, skalirano na raspon realnih brojeva [0, 1]
- NSWPrice: cijena struje u New South Walesu, skaliran na raspon realnih brojeva [0, 1]
- NSWDemand: potražnja za strujom u New South Walesu, skalirana na raspon realnih brojeva [0, 1]
- VICPrice: cijena struje u Victoriji, skaliran na raspon realnih brojeva [0, 1]
- VICDemand: potražnja za strujom u Victoriji, skalirana na raspon realnih brojeva [0, 1]
- Transfer: zakazan prijenos struje između dvije države, skaliran na raspon cijelih brojeva [0, 1]

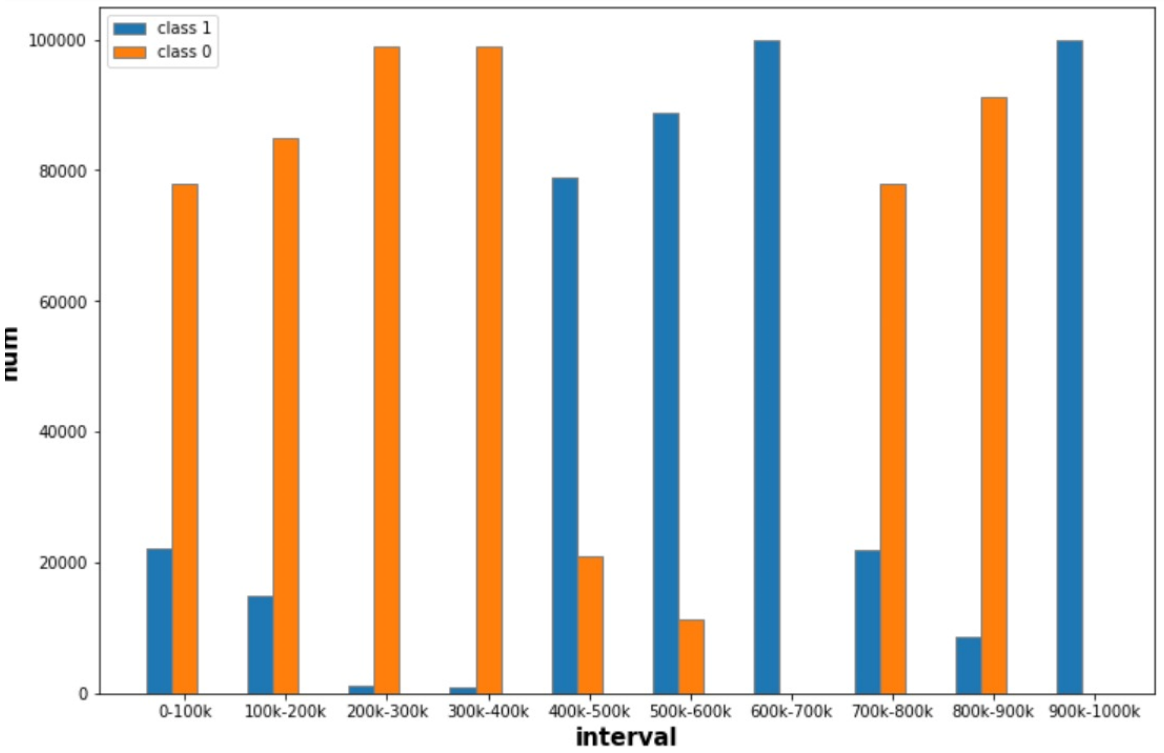
Oznaka primjera može biti 0.0 ili 1.0 i reflektira promjenu cijene u odnos na dnevni prosjek. Omjer pozitivne i negativne klase je 42:58. Pozitivna klasa s oznakom 1.0 je manjinska.

4.2.3. Skup KDDCup99

Skup KDDCup99 [25] ima ukupno preko 4 milijuna primjera, a u svrhu ovog rada uzima se samo prvih milijun primjera. Skup ima ukupno 41 značajku od kojih su 3 kategoričke, a ostale numeričke. Od numeričkih značajki, četiri su cijeli brojevi, a ostale realni.

Podaci su prikupljeni u svrhu izrade detektora upada u mrežu, prediktivnog modela koji može razlikovati upade u mrežu, to jest napade i normalne veze. U originalnom skupu podataka postoji 38 različitih tipova napada [25]. U svrhu ovog rada normalne veze su klasificirane kao negativne i označene s 0.0, a svi napadi su označeni s 1.0.

Normalne veze pripadaju većinskoj klasi. Omjer napada i normalnih veza je 437.613 : 562.387.



Slika 4: Distribucija prisutnosti klasa u skupu KDDCup99

Omjer klasa na čitavom skupu ne ukazuje na visoku razinu neuravnoteženosti skupa. Međutim, na Slici 4 vidi se da je u intervalima od 100.000 (100k) primjera skup izrazito

neuravnotežen i vidno je prisutan virtualni pomak koncepta uzrokovan promjenom apriorne vjerojatnosti klasa.

4.3. Korištene metode za pomak i neuravnoteženost

Metoda DDM-OCI (3.1) uvodi neke preinake u metodu DDM, a te preinake su napravljene na temelju postojeće implementacije metode za detekciju pomaka [27]. Za vrijednosti parametara modela koriste se zadane vrijednosti postojeće implementacije. α koji predstavlja prag upozorenja ima vrijednost 2.0, a β koji predstavlja prag detekcije pomaka vrijednost 3.0. Metoda ima i parametar *min_num_instances* koji predstavlja broj primjera koji mora doći na ulaz metode prije nego li se započne tražiti pomak. Parametar se koristi kako ne bi došlo do prerane detekcije pomaka, kada je jedan primjer od velike važnosti.

Implementacija metode ADWIN [28][28] također je preuzeta iz paketa *scikit-multiflow*. Za parametar vrijednosti pouzdanosti δ koristi se zadana vrijednost 0.002.

Korištena implementacija metode ADASYN [29] je iz paketa *imbalanced-learn*. Tu se također koriste zadani ulazni parametri. Parametri metode su:

- *sampling_strategy* – omjer klasa koji treba biti zadovoljen, zadana vrijednost je 1.0
- *random_state* – parametar koji kontrolira nasumičnost algoritma, zadana vrijednost je „None“
- *n_neighbors* – broj najbližih susjeda primjera za generiranje sintetičkih primjera, zadana vrijednost 5

4.4. Simuliranje toka podataka

Pomak je fenomen koji se javlja kroz vrijeme, a skupovi nad kojima se testiraju metode su u cijelosti poznati na početku. Zato se u svrhu detektiranja pomaka simulira tok podataka.

Skup za učenje se uzorkuje od prvih 10% podataka skupova SEA i KDDCup99, a otprilike 5% skupa ELEC2. Za skup ELEC2 se odabrao manji udio skupa jer se analizom skupa pokazalo da pomak nastaje unutar prvih 10% podataka. Ostatak skupa se simulira kao primjeri koji naviru na ulaz klasifikatora, i dalje na ulaz metoda za otkrivanje pomaka.

Pretpostavka je da se saznaje i točna oznaka primjera nakon klasifikacije primjera. Na temelju točne oznake traži se prisustvo pomaka.

Nakon što je otkriven pomak koncepta klasifikator se ponovo uči. Učenje se provodi nad novim dijelom skupa podataka. Novi skup za učenje sadrži isti broj primjera kao i prvotni skup za učenje. Novi primjeri se skupljaju od trenutka otkrivanja pomaka.

5. Rezultati

Pomak se traži nad svakim od navedenih skupova podataka s četiri različita pristupa, a za provjeru koriste se točnost i odziv bez otkrivanja pomaka, u tablicama koje slijede to je navedeno pod „Klasifikator“.

U Tablici 2 prikazan je prosječan broj detekcija pomaka koncepta na skupu podataka. U zagradama ispod broja otkrivanja pomaka nalaze se koraci oko kojih je najčešće otkriven pomak u podacima.

Metoda DDM-OCI je otkrila pomak u skupovima SEA 0 i SEA 1 u kojima nema ni pravog ni virtualnog pomaka koncepta. Metoda je fokusirana na odziv manjinske klase, odnosno pozitivne klase u skupu SEA 0 koji je uravnotežen. Metoda DDM-OCI koristi vremenski propadajući odziv klase za detekciju, što znači da je osjetljiva na greške pozitivne klase.

U skupu SEA 0 apriorna vjerojatnost pozitivne klase je 0.5, što znači više primjera koje je moguće netočno klasificirati, a skup za učenje ima samo 50 primjera pozitivne klase. U skupu SEA 1, apriorna vjerojatnost pozitivne klase je 0.1. Pomak koncepta je otkriven iako nije prisutan zbog malog broja primjera za učenje klase.

Potrebno je više vremena za detekciju pomaka u skupu SEA 1 jer se primjeri pozitivne klase rijetko pojavljuju.

	DDM	DDM-OCI	ADWIN	CIDD
SEA 0	0	4 (180, 450, 610, 800)	0	0
SEA 1	0	2 (430, 900)	0	0
SEA 2	1 (580)	3 (500, 780, 985)	1 (610)	0
SEA 3	1 (550)	3 (500, 660, 830)	0	0

SEA 4	0	1 (510)	0	0
SEA 5	2 (170, 680)	2 (510, 950)	0	0
SEA 6	2 (505, 780)	3 (420, 570, 725)	1 (550)	1 (580)
SEA 7	3 (130, 500, 720)	4 (450, 630, 800, 960)	1 (643)	1 (670)
ELEC 2	20	21	15	17
KDDCup99	6	5	6	5

Tablica 2: Broj otkrivenih pomaka po metodi i skupu podataka

U skupovima SEA 2 i SEA 3 simuliran je virtualni pomak koncepta jer se mijenja apriorna vjerojatnost pojave klasa u koraku 500. Metoda DDM točno otkriva virtualni pomak u oba skupa, dok ADWIN otkriva pomak samo kada je jak, i manjinska i većinska klasa su se zamijenile. CIDD ne otkriva pomak.

U skupovima SEA 4 i SEA 5 metode CIDD i ADWIN ne otkrivaju pomak. Ova činjenica je iznenađujuća s tim da je u tim skupovima simuliran pravi pomak koncepta. Na ovo može utjecati nasumična priroda generiranja primjera. Metoda DDM-OCI je jedina koja je ispravno otkrila pomak. Iako ponovo otkriva pomak u koraku kada ga nema.

Skupovi SEA 6 i SEA 7 imaju i pravi i virtualni pomak koncepta. Metode CIDD i ADWIN ispravno detektiraju pomak u oba slučaja. Skup SEA 6 ima jak pravi i virtualni pomak koncepta pa je pomak otkriven ranije.

DDM-OCI otkriva jedan pomak koji je prisutan i dva odnosno tri koji nisu prisutni. Metoda DDM također otkriva pomake, ali zbog prebrze detekcije, u točnom koraku kada je simuliran pomak, može se zaključiti da je pomak krivo otkriven.

Metoda	Točnost	Odziv y = 0	Odziv y = 1	F1 y = 0	F1 y = 1
Klasifikator	0,9248	0,9542	0,8955	0,9435	0,9224
DDM	0,9248	0,9542	0,8955	0,9435	0,9224
DDM-OCI	**0,9716**	0,9328	**0,9543**	**0,9461**	**0,9446**
ADWIN	0,9248	0,9542	0,8955	0,9435	0,9224
CIDD	0,9248	0,9542	0,8955	0,9435	0,9224

Tablica 3: Rezultati metoda na skupu SEA 0

Na skupu podataka SEA 0 koji nema pomak ni neuravnoteženost klasa metoda DDM-OCI je imala najbolji rezultat, kao što se vidi u Tablici 3. Međutim, DDM-OCI ima najbolje rezultate jer je metoda vrlo osjetljiva na pad odziva pozitivne klase i prisutnost šuma. Kako je fokus metode na pozitivnoj klasi, odziv negativne klase je nešto lošiji nego odziv klasifikatora.

Klase su uravnotežene u skupu SEA 0 pa metoda CIDD ne generira sintetičke primjere.

Metoda	Točnost	Odziv y = 0	Odziv y = 1	F1 y = 0	F1 y = 1
Klasifikator	0,9504	**0,9992**	0,4555	0,9702	0,6219
DDM	0,9504	**0,9992**	0,4555	0,9702	0,6219
DDM-OCI	**0,9649**	0,9980	0,5749	**0,9754**	0,7211
ADWIN	0,9504	**0,9992**	0,4555	0,9702	0,6219
CIDD	0,9578	0,9753	**0,7533**	0,9739	**0,7623**

Tablica 4: Rezultati metoda na skupu SEA 1

Prema Tablici 4, na skupu podataka SEA 1 koji nema pomaka, ali ima visok stupanj neuravnoteženosti metoda, najveću točnost ima metoda DDM-OCI zbog čestog ponovnog učenja modela. Sve metode imaju dobru razinu točnosti, ali odziv pozitivne odnosno

manjinske klase je poprilično nizak. Iznimka je metoda CIDD koja ima najveći odziv i F1 mjeru manjinske klase bez ponovnog učenja modela. CIDD ima najlošiji odziv većinske klase, ali i dalje ima bolju točnost modela i F1 mjeru većinske klase od klasifikatora. Metode DDM i ADWIN ne otkrivaju pomak, pa imaju iste rezultate kao klasifikator.

Metoda	**Točnost**	**Odziv y = 0**	**Odziv y = 1**	**F1 y = 0**	**F1 y = 1**
Klasifikator	0,7531	**0,9887**	0,5558	0,7848	0,7102
DDM	0,9358	0,9663	0,8591	0,9228	0,9067
DDM-OCI	**0,9666**	0,9782	**0,9178**	**0,9575**	**0,9436**
ADWIN	0,9238	0,9740	0,8359	0,9109	0,8979
CIDD	0,8962	0,9575	0,8236	0,8832	0,8860

Tablica 5: Rezultati metoda na skupu SEA 2

Na skupu podataka SEA 2 koji ima visok stupanj neuravnoteženosti i jak pomak u neuravnoteženosti, u Tablici 5 vidi se velik pad točnosti klasifikatora u odnosu na skup SEA 1. Najbolje rezultate daje DDM-OCI zbog prečestog ponovnog učenja modela.

DDM je osjetljiviji od metode ADWIN, ranije otkriva pomak, pa ima nešto bolje rezultate. Metoda CIDD ne troši resurse za ponovno učenje. Zbog uravnotežavanja podataka metodom ADASYN, CIDD postiže stabilnu točnost kroz čitav skup podataka.

Metoda CIDD nije otkrila pomak u podacima, ali i dalje ima odzive i F1 mjere približno jednake metodi ADWIN koja je otkrila pomak te ažurirala model.

Metoda	**Točnost**	**Odziv y = 0**	**Odziv y = 1**	**F1 y = 0**	**F1 y = 1**
Klasifikator	0,8646	**0,9970**	0,5858	0,9089	0,7359

DDM	0,9339	0,9847	0,7428	0,9421	0,8346
DDM-OCI	**0,9654**	0,9856	**0,8042**	**0,9632**	**0,8692**
ADWIN	0,8646	**0,9970**	0,5858	0,9089	0,7359
CIDD	0,8719	0,9571	0,6489	0,9013	0,7447

Tablica 6: Rezultati metoda na skupu SEA 3

Metode ADWIN i CIDD ne otkrivaju pomak na skup podataka SEA 3 koji ima slab virtualni pomak u podacima. CIDD ima bolji odziv i F1 mjeru manjinske klase nego ADWIN u Tablici 6. To je očekivano zbog uravnotežavanja klasa, ali svejedno ima lošiji odziv nego DDM i DDM-OCI koje ažuriraju model više puta. Zbog veće apriorne vjerojatnosti manjinske klase u skupu podataka, CIDD ima veću točnost nego klasifikator jer je bolje naučio manjinsku klasu.

Skup SEA 4, čiji su rezultati prikazani u Tablici 7, ima jak pravi pomak koncepta bez promjene u apriornoj vjerojatnosti klasa. Najbolje rezultate ponovo daje metoda DDM-OCI koja je jedina detektirala pomak i to ubrzo nakon promjene. Metoda CIDD, kao i ostale metode, nije otkrila pomak, ali ima bolje rezultate u svim mjerama od ostalih (osim DDM-OCI), s iznimkom odziva većinske klase.

Metoda	**Točnost**	**Odziv y = 0**	**Odziv y = 1**	**F1 y = 0**	**F1 y = 1**
Klasifikator	0,9299	**0,9970**	0,2488	0,9584	0,3899
DDM	0,9299	**0,9970**	0,2488	0,9584	0,3899
DDM-OCI	**0,9548**	0,9933	**0,5012**	**0,9693**	**0,6363**
ADWIN	0,9299	**0,9970**	0,2488	0,9584	0,3899
CIDD	0,9339	0,9797	0,4488	0,9601	0,5500

Tablica 7: Rezultati metoda na skupu SEA 4

Za skup SEA 5 najbolju točnost ima metoda DDM što se može vidjeti u Tablici 8. Međutim, DDM otkriva pomak dva puta, od kojih je jedan otkriven prije nastanka pomaka. Sljedeća najbolja metoda je DDM-OCI, koja ima malo lošije točnosti i odziv većinske klase od metode DDM, ali točno otkriva pomak. DDM-OCI ima i najbolje rezultate za odziv manjinske klase i obje F1 mjere.

Nakon metode DDM-OCI najbolji odziv manjinske klase ima metoda CIDD, ali najlošije F1 mjere. Kako su u F1 mjeri povezane preciznost i odziv, CIDD metoda ima i lošiju preciznost od ostalih metoda.

Metoda	**Točnost**	**Odziv y = 0**	**Odziv y = 1**	**F1 y = 0**	**F1 y = 1**
Klasifikator	0,9555	0,9983	0,5222	0,9732	0,6792
DDM	**0,9674**	**0,9987**	0,5518	0,9743	0,7034
DDM-OCI	0,9638	0,9956	**0,5783**	**0,9744**	**0,7143**
ADWIN	0,9555	0,9983	0,5222	0,9732	0,6792
CIDD	0,9418	0,9751	0,5777	0,9645	0,6407

Tablica 8: Rezultati metoda na skupu SEA 5

Skup SEA 6 ima jak i pravi i virtualni pomak koncepta pa sve metode otkrivaju pomak i ažuriraju model. CIDD i ADWIN ispravno otkrivaju po jedan pomak nedugo nakon promjene koncepta. DDM ima najbolje rezultate prema Tablici 9 jer otkriva pomak samo 5 koraka nakon promjene koncepta, ali ponovo otkriva pomak kojeg nema pri kraju skupa. DDM-OCI zbog prečestog otkrivanja pomaka ima druge najbolje rezultate preko većine mjera.

Od svih metoda, CIDD ima najlošije rezultate. Metoda ADWIN je bolja, jer prije otkriva pomak pa se prije prilagodi novom konceptu.

Metoda	**Točnost**	**Odziv y = 0**	**Odziv y = 1**	**F1 y = 0**	**F1 y = 1**

Klasifikator	0,5699	**0,9917**	0,2171	0,6775	0,3544
DDM	**0,9649**	0,9708	**0,9217**	**0,9559**	**0,9405**
DDM-OCI	0,9522	0,9799	0,8354	0,9284	0,8980
ADWIN	0,9304	0,9745	0,8518	0,9181	0,9075
CIDD	0,8988	0,9571	0,7899	0,8835	0,8619

Tablica 9: Rezultati metoda na skupu SEA 6

Na skupu SEA 7, metoda DDM-OCI ponovo ima najveću točnost, ali uz najveći broj netočnih otkrivanja pomaka. Zbog učestalih ažuriranja modela najbolje rezultate daje i za odziv manjinske klase, te obje F1 mjere, prema Tablici 10.

CIDD i ADWIN točno otkrivaju pomak, ali je ADWIN nešto brža, prema Tablici 2. ADWIN daje bolje rezultate od CIDD-a po svim mjerama osim po odzivu manjinske klase. CIDD i ADWIN imaju bolje rezultate na skupu SEA 6 nego na skupu SEA 7. Zbog blagog pomaka u podacima, duže im je potrebno za otkriti pomak, pa kasno ažuriraju modele.

Metoda	**Točnost**	**Odziv y = 0**	**Odziv y = 1**	**F1 y = 0**	**F1 y = 1**
Klasifikator	0,8077	**0,9943**	0,4151	0,8752	0,5818
DDM	0,9508	0,9552	0,8199	0,9413	0,8506
DDM-OCI	**0,9662**	0,9742	**0,8425**	**0,9603**	**0,8792**
ADWIN	0,9162	0,9753	0,7095	0,9287	0,8031
CIDD	0,9097	0,9532	0,7344	0,9221	0,7969

Tablica 10: Rezultati metoda na skupu SEA 7

Metode DDM-OCI i DDM pokazale su se najuspješnije prema mjeri točnosti kroz umjetne skupove, ali po cijenu prečestog otkrivanja pomaka, čak i kada on nije prisutan. Resursno su

skupe jer zahtijevaju brojna ažuriranja modela i skupljanja podataka koji pripadaju novom konceptu za učenje modela.

Metode ADWIN i CIDD su resursno bolje, ali imaju manju točnost. ADWIN je brža pri otkrivanju pomaka i ima bolje rezultate na skupovima podataka gdje je prisutan pravi pomak koncepta (SEA 5, SEA 6, SEA 7), ali daje lošije rezultate, pogotovo za mjeru odziva manjinske klase, na skupovima gdje je neuravnoteženost klasa izraženija (SEA 1, SEA 3). Za većinsku klasu najbolji odziv imao je klasifikator (stablo odluke) bez metoda detekcije pomaka, što upućuje na znatno prilagođavanje većinskoj klasi.

Na umjetnim skupovima, pokazalo se da DDM i DDM-OCI često otkrivaju lažne pomake koncepta, dok metode ADWIN i CIDD nisu otkrile pomak kada je bio prisutan samo pravi pomak koncepta u skupovima SEA 4 i SEA 5

Metoda	**Točnost**	**Odziv y = 0**	**Odziv y = 1**	**F1 y = 0**	**F1 y = 1**
Klasifikator	0,6959	0,9797	0,3135	0,7873	0,4669
DDM	0,9773	**0,8451**	0,7254	**0,8234**	**0,7488**
DDM-OCI	**0,9841**	0,7648	**0,7581**	0,7959	0,7146
ADWIN	0,9315	0,7825	0,7239	0,7872	0,7179
CIDD	0,9309	0,8018	0,7079	0,7933	0,7274

Tablica 11: Rezultati metoda na skupu ELEC2

Na skupu podataka ELEC2, metode DDM i DDM-OCI otkrivaju više pomaka nego metode ADWIN i CIDD, prema Tablici 2. Iako, prema Tablici 11, DDM-OCI ima vidno bolju točnost, CIDD ima bolji odziv negativne klase i bolju mjeru F1 pozitivne klase. Mjera F1 negativne klase je malo bolja za metodu DDM-OCI.

Metoda CIDD ima manji odziv manjinske klase nego metoda ADWIN, u umjetnom skupu podataka to se vidjelo na primjerima gdje je prisutan jak pomak pravog koncepta i jak virtualni pomak koncepta. Metoda CIDD je imala poprilično dobre rezultate za F1 mjere.

Metoda	Točnost	Odziv y = 0	Odziv y = 1	F1 y = 0	F1 y = 1
Klasifikator	0,9794	0,9917	0,9652	0,9813	0,9776
DDM	**0,9999**	**0,9999**	**0,9999**	**0,9999**	**0,9999**
DDM-OCI	0,9901	0,9426	0,9902	0,9599	0,9821
ADWIN	0,9998	0,9990	0,9995	0,9991	0,9995
CIDD	0,9984	0,9994	0,9995	0,9995	0,9994

Tablica 12: Rezultati metoda na skupu KDDCup99

Metode DDM-OCI i CIDD nisu se iskazale na skupu KDDCup99 u kontekstu točnosti prema Tablici 12, ali su otkrile manje pomaka nego metode ADWIN i DDM kao što je prikazano u Tablici 2. DDM-OCI se pokazala najlošijom metodom na skupu KDDCup99. Imala je najlošije rezultate u mjerama većinske klase, što je očekivano jer prati samo odziv manjinske klase. Međutim, DDM-OCI imala je i najlošije rezultate za točnost i za mjere manjinske klase.

Zaključak

Pomak koncepta i neuravnoteženost klasa znatno utječu na točnost i odziv modela strojnog učenja. Kako bi se spriječilo opadanje točnosti modela nastale su metode poput ADWIN-a i DDM-a za otkrivanje pomaka koncepta. Za skupove podataka u kojima je prisutna neuravnoteženost klasa koriste se i nadograđene metode detekcije pomaka koje uzimaju u obzir i neuravnoteženost, metode poput CIDD-a i DDM-OCI-a.

Fokus kod CIDD-a i DDM-OCI-a je u povećanju odziva manjinske klase, za razliku od metoda ADWIN i DDM koje prate točnost modela. DDM-OCI u svrhu povećanja odziva i otkrivanja pomaka prati mjeru vremenski degradiranog odziva. CIDD uravnotežava skup podataka za učenje generirajući sintetičke podatke.

Odabir metode za praćenje modela ovisi o poslovnim zahtjevima. Ako ima dovoljno dostupnih resursa za često učenje modela, u ovom radu je pokazano da metode DDM i DDM-OCI imaju najbolje rezultate, ali su vrlo osjetljive na manje promjene i detektiraju pomak i kada nije prisutan. Metoda DDM-OCI potpuno zanemaruje većinsku klasu, a zbog malog broja primjera model ne može dobro naučiti manjinsku klasu.

Ako *use case* traži bolji odziv i preciznost manjinske klase pri visokom stupnju neuravnoteženosti skupa podataka uz što manje ponovljenih učenja modela onda se pokazalo da je metoda CIDD najbolja.

U daljnjem radu bilo bi dobro istražiti kako bi se metoda DDM-OCI ponašala pri praćenju degradacije modela učenom nad uravnoteženim skupom podataka koji se koristi za klasifikaciju neuravnoteženog toka podataka.

Literatura

[1] Gama, J., Žliobaite, I., Bifet, A., Pechenizkiy, M., Bouchachia, A., *A survey on concept drift adaptation*. ACM Computing Surveys, 46, 4 (2014) str. 1–44.

[2] Hoens, T.R., Polikar, R., Chawla, N.V. *Learning from streaming data with concept drift and imbalance: an overview*. Progress in Artificial Intelligence, 1,1, (2012) str. 89–101.

[3] Kelly, M., Hand, D., Adams, N., *The impact of changing populations on classifier performance*. KDD '99: Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, (1999), str. 367–371.

[4] Žliobaite, I*., Learning under Concept Drift: an Overview*, CoRR, abs/1010.4784 (2010).

[5] Salganicoff, M., *Tolerating concept and sampling shift in lazy learning using prediction error context switching*, Artificial Intelligence Review, Special Issue on Lazy Learning, 11, (1997), str.133-155.

[6] Tsymbal, A., Pechenizkiy, M., Cunningham, P., Puuronen, S., *Dynamic integration of classifiers for handling concept drift*, Information Fusion, 9,1 (2008) str. 56–68.

[7] Kolter, J.Z., Maloof, M. A., *Dynamic weighted majority: An ensemble method for drifting concepts,* Journal of Machine Learning Research, 8, 91 (2007) str. 2755-2790

[8] Sarnovsky, M., Marcinko, J., *Adaptive Bagging Methods for Classification of Data Streams with Concept Drift*, Acta Polytechnica Hungarica, 18, 3 (2021) str. 47-63.

[9] Gama, J., *Knowledge Discovery from Data Streams*, 1. izdanje Chapman & Hall/CRC, (2010)

[10] Pinage, F., dos Santos, E.M., Gama, J, *A drift detection method based on dynamic classifier selection*, Data Mining and Knowledge Discovery, 34, str.50-74, (2020)

[11] Bayram, F., Ahmed, B.S., Kassler, A., *From concept drift to model degradation: An overview on performance-aware drift detectors*, Knowledge-Based Systems, 245, (2022)

[12] Gama, J., Medas, P., Castillo, G., Rodrigues, P., *Learning with Drift Detection*, Advances in Artificial Intelligence - SBIA 2004, 17th Brazilian Symposium on Artificial Intelligence, São Luis, (2004), str. 286-295.

[13] Baena-Garcia, M., del Campo-Avila, J., Fidalgo, R., Bifet, A., Gavalda, R., Morales-Bueno, R., *Early Drift Detection Method*, 4th International Workshop on Knowledge Discovery from Data Streams, Berlin, (2006), str. 77-86.

[14] Bifet, A, Gavalda, R, *Learning from time-changing data with adaptive windowing*. Proceedings of the 2007 SIAM International Conference on Data Mining, Minneapolis, (2007), str. 443-448

[15] Bifet, A., Kirkby, R., *Data Stream Mining a Practical Approach,* London: Kluwer Academic Publishers 2009

[16] Wang, S., Minku, L. L.,. Yao, X, *A Systematic Study of Online Class Imbalance Learning With Concept Drift*, in IEEE Transactions on Neural Networks and Learning Systems, 29, 10, (2018), str. 4802-4821.

[17] Seifert, C., Khoshgoftaar, T.M., Van Hulse, J., Napolitano, A., *Mining data with rare events: a case study*. Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence, Patras, (2007), str. 132–139.

[18] Johnson, J.M., Khoshgoftaar, T.M., *Survey on deep learning with class imbalance,* Journal of Big Data, 6, 27, (2019), str. 1-54.

[19] Chawla, N.V., Japkowicz, N., Kotcz, A., *Special Issue on Learning from Imbalanced Data Sets*., 6, ACM SIGKDD Explorations Newsletter, (2004), str. 1–6.

[20] He, H., Bai, Y., Garcia, E.A., Li, S., *ADASYN: Adaptive synthetic sampling approach for imbalanced learning*. 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, (2008) str. 1322–1328

[21] Wang, S., Minku, L.L. , Ghezzi, D., Caltabiano, D., Tino P., Yao X., *Concept Drift Detection for Online Class Imbalance Learning,* The 2013 International Join Conference on Neural Networks, Dallas, (2013), str. 1-10.

[22] Wang S., Minku, L. L., Yao, X., *A learning framework for online class imbalance learning,* IEEE Symposium Series on Computational Intelligence and Ensemble Learning, Singapore, (2013), str. 36-45.

[23] Priya, S., Annie Uthra R., *Deep learning framework for handling concept drift and class imbalanced complex decision-making on streaming data*, Complex and Intelligent Systems, (2021), str: 1-17.

[24] Sharan, Y., The Elec2 Dataset, Kaggle, Poveznica: https://www.kaggle.com/datasets/yashsharan/the-elec2-dataset; pristupljeno 23. travnja 2023.

[25] *KDD Cup 1999 Data*, The UCI KDD Archive, (1999, listopad) Poveznica: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html; pristupljeno 10. ožujka 2023.

[26] sklearn.ensemble.BaggingClassifier, scikit-learn, poveznica: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingClassifier.html; pristupljeno: 25. travnja 2023.

[27] skmultiflow.drift_detection.DDM, scikit-multiflow, poveznica: https://scikit-multiflow.readthedocs.io/en/stable/api/generated/skmultiflow.drift_detection.DDM.html; pristupljeno 24. travnja 2023.

[28] skmultiflow.drift_detection.ADWIN, scikit-multiflow, poveznica: https://scikit-multiflow.readthedocs.io/en/stable/api/generated/skmultiflow.drift_detection.ADWIN.html pristupljeno 24. travnja 2023.

[29] imblearn.over_sampling.ADASYN, imbalanced-learn.org, poveznica: https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.ADASYN.html pristupljeno: 18. travnja 2023.

[30] Elabd, M., *What is Bagging classifier,* Medium, (2022, veljača). Poveznica: https://medium.com/@arch.mo2men/what-is-bagging-classifier-45df6ce9e2a1; pristupljeno 23. lipnja 2023.

[31] Singh Chauhan, N., *Decision Tree Algorithm, Explained,* Kdnuggets, (2022, veljača), poveznica: https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html; pristupljeno 23. lipnja 2023.

[32] Street, W. N., Kim, Y., *A streaming ensemble algorithm (sea) for large-scale classification*, Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, (2021), str. 377–382.

[33] Šnajder, J., *Strojno učenje 1: Osnovni koncepti*, Materijali s predavanja, UNIZG FER (2022), str. 1-21.

Sažetak

Pojava tokova podataka uvela je novu dimenziju strojnog učenja. U tokovima podataka podaci najčešće pristižu kontinuirano i velikom brzinom. Analiza toka podataka u stvarnom vremenu zahtijeva algoritme i modele koji se mogu prilagoditi promjenjivoj distribuciji podataka, nositi se s pomicanjem koncepta, neuravnoteženosti klasa i stvarati predviđanja u hodu, kako podaci pristižu.

Cilj ovog diplomskog rada bio je usporediti dvije metode koje se bave pomakom koncepta u tokovima podataka uz prisustvo neuravnoteženosti klasa. Metoda DDM-OCI u svrhu pomaka koncepta u neuravnoteženom toku podataka prati odziv manjinske klase. Metoda CIDD uravnotežava skup za učenje metodom ADASYN, a prati točnost modela metodom ADWIN.

Ispitivanjem metoda nad jednim umjetnim i dva skupa podataka iz stvarnog svijeta, pokazalo se da je metoda DDM-OCI veće točnosti u 9 od 10 ispitnih slučajeva i boljeg odziva manjinske klase u 8 od 10 skupova ispitnih slučajeva u odnosu na metodu CIDD. Međutim, ona to postiže uz cijenu čestog ponovnog učenja modela i s tim velikog trošenja resursa za prikupljanje podataka svaki put kada je potrebno učenje modela.

Strojno učenje, tokovi podataka, pomak koncepta, neuravnoteženost klasa, točnost, odziv

Summary

The emergence of data streams has introduced a new dimension to machine learning. In data streams, data usually arrives continuously and at high speed. Real-time data stream analysis requires algorithms and models that can adapt to changing data distributions, deal with concept drift, class imbalances, and make predictions on the fly, as data arrives.

The aim of this master thesis was to compare two methods that deal with concept drift in data streams in the presence of class imbalance. The DDM-OCI method follows the recall of the minority class for the purpose of detecting concept drift in an imbalanced data stream. The CIDD method balances the learning set using the ADASYN method, and monitors the model accuracy using the ADWIN method.

By testing the methods on an artificial and two real-world datasets, it was shown that the DDM-OCI method is more accurate in 9 out of 10 test cases and has a better response of the minority class in 8 out of 10 test cases than the CIDD method. However, this was acheived at the cost of frequent relearning of the model and, thus, there is a large expenditure of data collection resources each time the model needs to be retrained.

Machine learning, data streams, concept drift, class imbalance, accuracy, concept drift

Skraćenice

PAC	*Probably Approximately Correct*	vjerojatno približno točno
ADWIN	*Adaptive Windowing*	metoda prilagođavajućih prozora
DDM	*Drift Detection Method*	metoda otkrivanja pomaka
TN	*True Negative*	istinito negativno
TP	*True Positive*	istinito pozitivno
FN	*False Negative*	lažno negativno
FP	*False Positive*	lažno pozitivno
TPR	*True Positive Rate*	istinita pozitivna stopa
ADASYN	*Adaptive Synthetic Sampling*	adaptivno sintetičko uzorkovanje
DDM-OCI	*Drift Detection Method for Online Class Imbalance*	Metoda detekcije pomaka za neuravnoteženost klasa
CIDD-ADODNN	*Concept Drift Detection using Adadelta optimizer based deep neural networks*	Detekcija pomaka koncepta koristeći duboke neuronske mreže bazirane na Adadelta optimizatoru
BAGGING	*Bootstrap aggregation*	Bootstrap agregiranje