

Učenje reprezentacije dinamike dvostruke aukcije pomoću dubokih neuronskih mreža za grafove

Dužević, Pavao

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:168:754840>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-04-01**



Repository / Repozitorij:

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 530

**UČENJE REPREZENTACIJE DINAMIKE DVOSTRUKE
AUKCIJE POMOĆU DUBOKIH NEURONSKIH MREŽA ZA
GRAFOVE**

Pavao Dužević

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 530

**UČENJE REPREZENTACIJE DINAMIKE DVOSTRUKE
AUKCIJE POMOĆU DUBOKIH NEURONSKIH MREŽA ZA
GRAFOVE**

Pavao Dužević

Zagreb, lipanj 2024.

DIPLOMSKI ZADATAK br. 530

Pristupnik: **Pavao Dužević (0036511708)**

Studij: Računarstvo

Profil: Računarska znanost

Mentor: prof. dr. sc. Zvonko Kostanjčar

Zadatak: **Učenje reprezentacije dinamike dvostruke aukcije pomoću dubokih neuronskih mreža za grafove**

Opis zadatka:

U radu je potrebno istražiti primjenu dubokih neuronskih mreža za grafove u modeliranju dinamike dvostruke aukcije na financijskim tržištima. Obzirom na složenost i veliku količinu podataka generiranih u financijskim transakcijama, analitički modeli često nisu prikladni, prvenstveno jer se oslanjaju na veliki broj pretpostavki koje nužno ne vrijede. Prvo je potrebno razviti metodologiju zasnovanu na teoriji grafova za reprezentaciju dinamike skupova tržišnih transakcija. Zatim je potrebno istražiti kako se različiti grafovi mogu iskoristiti za treniranje modela dubokog učenja za grafove. Fokus istraživanja potrebno je usmjeriti na dimenzionalnosti, omogućavajući time jednostavniju analizu i klasifikaciju tržišnih fenomena. Na kraju, potrebno je razmotriti potencijalne primjene razvijenih modela, uključujući predviđanje tržišnih kretanja te razumijevanje utjecaja različitih tipova transakcija na dinamiku cijene odgovarajuće financijske imovine.

Rok za predaju rada: 28. lipnja 2024.

Sadržaj

1. Uvod	2
2. Dvostruka aukcija	3
2.1. Generirani signali	3
2.2. Dinamika dvostruke aukcije	6
2.2.1. Prostor dinamike	8
2.2.2. Prikaz skupa transakcija u formi histograma	12
2.2.3. Izgradnja grafa: motivacija i postupak	14
3. Neuronske mreže za grafove	17
3.1. Notacija i svojstva	17
3.2. Konvolucijske neuronske mreže za grafove	19
3.3. Neuronske mreže za grafove s pažnjom	21
3.4. Izomorfne neuronske mreže za grafove	23
4. Metodologija	25
4.1. Samonadzirano učenje reprezentacije na razini grafa	25
4.2. Optimizacija i evaluacija modela	31
5. Rezultati	34
6. Zaključak	49
Literatura	50
Sažetak	51
Abstract	52

1. Uvod

Duboke neuronske mreže postale su neizbježan alat u suvremenom pristupu modeliranja i analize podataka. Budući da su iznimno dobri funkcijski aproksimatori te posjeduju svojstva modeliranja kompleksnih transformacija, obrade velike količine podataka, robusnosti te regulacije pristranosti, posebno su korisne za probleme gdje analitički modeli nisu dovoljno efikasni (analitički modeli uobičajeno su zasnovani na ne nužno točnim pretpostavkama). S obzirom na navedena svojstva, njihova primjena u optimizaciji donošenja odluka unutar dvostruke aukcije (burzovnog trgovanja) sve je zastupljenija. Budući da danas možemo trgovati na burzama putem internetske veze, takvi sustavi generiraju izrazito frekventne signale. Standardni način prikaza signala dvostruke aukcije agregira signal zadnjih izvršenih transakcija s obzirom na određeni vremenski okvir u 4 vrijednosti: cijena na početku i na kraju te najviše i najniža cijena unutar vremenskog okvira. Takvom metodom agregiranja izvornog signala odbacujemo veliku količinu informacija. Također, takav prikaz dovodi do nepoželjnih pojava poput aliasinga. Primjena dubokih neuronskih mreža nad takvim podacima ne daje impresionirajuće rezultate kao u nekim drugim područjima (poput obrade prirodnog jezika ili računalnog vida). Cilj ovog rada je predstaviti alternativan način prikaza izvornog signala zadnjih izvršenih transakcija u strukturi grafa koji opisuje dinamiku dvostruke aukcije. Za tako prikazane podatke učimo vektorsku reprezentaciju uz pomoć dubokih neuronskih mreža za grafove. Takva reprezentacija bi trebala sadržavati puno veću količinu informacija od standardnog prikaza pozicionirajući naučene reprezentacije grafova blizu s obzirom na sličnost signala zadnjih izvršenih transakcija. Dodatno, naučena reprezentacija i grupe koje se formiraju unutar takvog prostora otvaraju mogućnost detekcije ponavljajućih uzoraka izvornog signala te bi takvi uzorci bili analogni tokenima u obradi prirodnog jezika.

2. Dvostruka aukcija

Dvostruka aukcija je sustav tržišne razmjene koji je primijenjen na burzovnim tržištima. Unutar tog sustava imamo dvije strane: kupovnu i prodajnu. Svaka od strana ima dvije vrste sudionika: održavatelje tržišta (engl. market maker) i oduzimatelje likvidnosti (engl. market taker). Sudionici ulaze u interakciju sa sustavom postavljanjem naloga koji sadrže definiranu količinu (količina kojom sudionik želi trgovati određenim financijskim instrumentom). Održavatelji tržišta postavljaju svoje naloge unutar knjige naloga (engl. order book) s namjerom da se izvrše na točno određenoj cijeni. S druge strane, oduzimatelji likvidnosti izvršavaju svoje naloge po trenutnoj najboljoj kupovnoj (engl. best bid), tj. najboljoj prodajnoj cijeni (engl. best ask) koju nude održavatelji tržišta. Održavatelj tržišta ima pravo povući svoj nalog sve do trenutka izvršenja istog, odnosno do trenutka kada baš njegov nalog predstavlja najbolju kupovnu ili prodajnu cijenu unutar knjige naloga. U trenutku kada se spoje nalozi održavatelja tržišta i oduzimatelja likvidnosti, tržište generira signal zadnje izvršene transakcije. Signal promjene knjige naloga može biti generiran postavljanjem, odnosno poništavanjem naloga održavatelja tržišta ili spajanjem naloga trenutno najbolje kupovne ili prodajne cijene s oduzimateljem likvidnosti.

2.1. Generirani signali

S obzirom na način funkcioniranja dvostruke aukcije, takav sustav generira dva diskretna signala: signal zadnje izvršene transakcije i signal promjene knjige naloga.

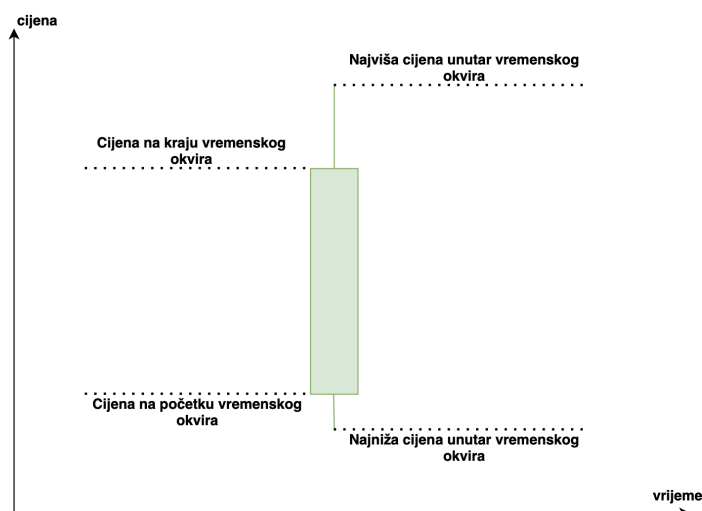
Signal zadnje izvršene transakcije je peterodimenzionalan: redni broj izvršene transakcije, vremenska oznaka (engl. timestamp) izvršenja transakcije, količina transakcije, cijena na kojoj je izvršena transakcija i binarna zastavica koja označava je li oduzimatelj likvidnosti postavio kupovni ili prodajni nalog. Redni broj pruža informaciju o uređenosti jer zbog rezolucije vremenske oznake i visoke frekvencije dolazaka naloga te stohastičnosti nji-

h ovog izvršavanja moguće je da dvije ili više transakcija zaredom imaju isti vremensku oznaku. U dokumentacijama različitih burzi ili brokerskih kuća ovaj signal se još zove signal prve razine (engl. level 1 data).

Signal promjene knjige naloga može biti dostupan u dva oblika. Prvi oblik sadrži informaciju o promjeni agregirane količine na određenoj cijeni unutar određene cjenovne rezolucije. Drugi oblik sadrži informaciju o svakoj pojedinačnoj promjeni unutar knjige naloga od strane održavatelja tržišta. Kombinacija signala zadnje izvršene transakcije i prvog oblika u dokumentacijama možemo pronaći pod nazivom signal druge razine (engl. level 2 data), a kombinaciju signala zadnje izvršene transakcije i drugog oblika pod nazivom signal treće razine (engl. level 3 data). Budući da nije u domeni ovog rada, signalom promjene knjige naloga nećemo se dodatno baviti.

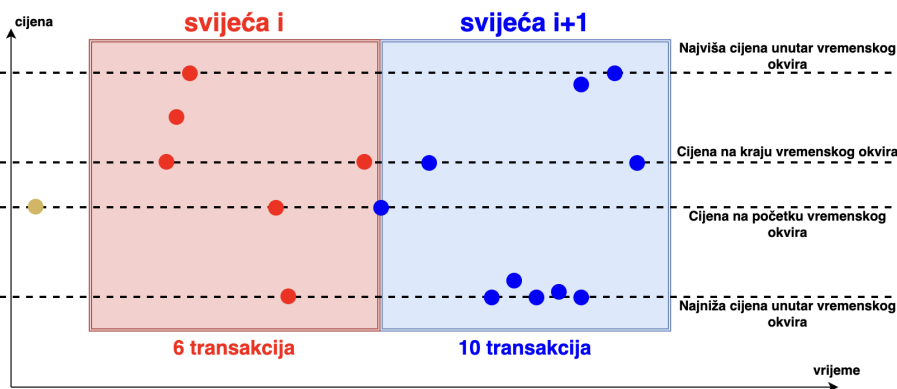
Standardna metoda agregiranja signala izvršenih transakcija: OHLC

Standardna metoda agregiranja i prikaza signala zadnjih izvršenih transakcija zove se OHLC svijeća (engl. OHLC candlestick). Takav pristup svodi se na agregiranje signala zadnjih izvršenih transakcija unutar definiranog vremenskog okvira svodeći informaciju signala na 4 vrijednosti: cijena otvaranja (engl. open price) - cijena na početku vremenskog okvira, cijena zatvaranja (engl. close price) - cijena na kraju vremenskog okvira te najveća (engl. high price) i najmanja cijena unutar vremenskog okvira (engl. low price). Formirana struktura podatka zbog toga nosi naziv OHLC.



Slika 2.1. OHLC prikaz signala zadnjih izvršenih transakcija.

Najčešće uz samu svijeću moguće je pronaći i informaciju o istrgovanoj količini unutar svijeće, ali ne i o broju transakcija ili o disjunktnoj kupovnoj i prodajnoj istrgovanoj količini. Općenito možemo tvrditi da frekventnost transakcija nije isključivo funkcija varijable vremena (iako vrijeme kao varijabla po kojoj se izvršavaju nalozi dolazi do izražaja u trenucima otvaranja i zatvaranja burze ili prilikom dolaska financijskih izvještaja). Definiranjem vremenskog okvira unutar kojeg formiramo OHLC prikaz dodatno uvodimo pristranost prema vremenu - generiramo podatak unutar skupa podataka isključivo pod uvjetom vremenskog okvira. Takav prikaz jednako tretira tržišna stanja unutar kojih se trguje velikom i malom frekventnošću te velikim i malim količinama. Jedina informacija koja je obuhvaćena ovim prikazom je *obris* cijene, informaciju kako je došlo do tog *obrisa* nemamo.



Slika 2.2. OHLC svijeća: podložnost aliasingu.

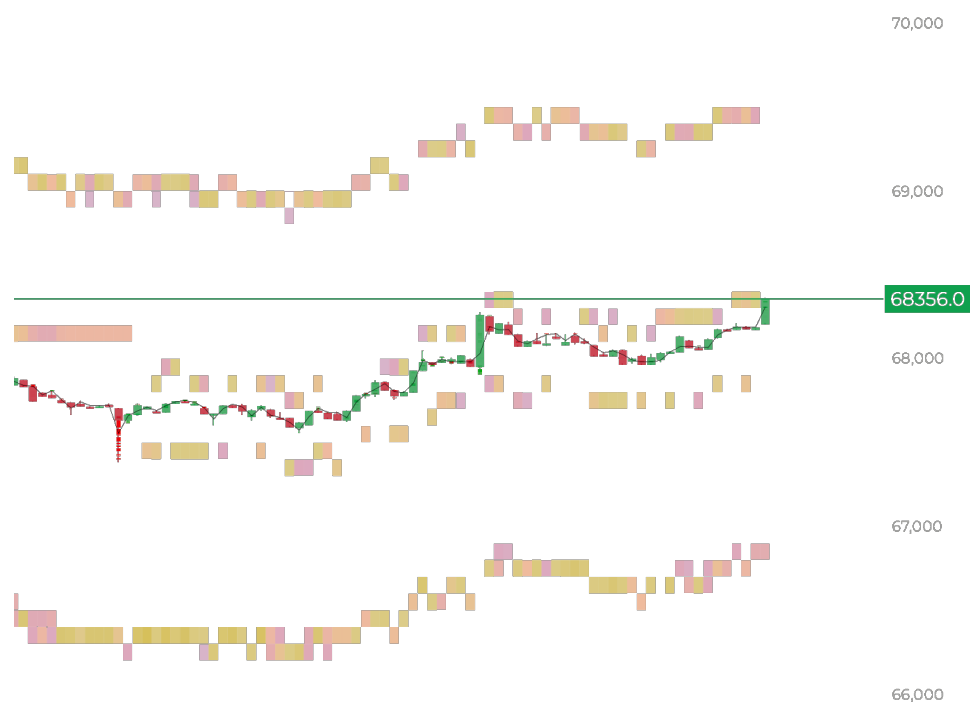
Na slici 2.2. vidimo ilustraciju podložnosti OHLC prikaza efektu aliasinga. Dva različita uzorka signala izvršenih transakcija (gdje jedan ima manji broj transakcija) prikazuju se kao ista OHLC svijeća. Stoga, ako se koristi OHLC prikaz kao ulaz u duboku neuronsku mrežu ne možemo očekivati da mreža uspješno može napraviti razliku između ovakva dva različita uzorka transakcija prilikom zaključivanja.

2.2. Dinamika dvostruke aukcije

Motivacija

Unutar sustava dvostruke aukcije možemo opaziti sljedeće:

- Gusto koncentrirana količina (ali ne nužno i broj naloga) oko neke cijene unutar knjige naloga smanjuje vjerojatnost prolaska zadnje izvršene cijene preko te cijene. Da bi cijena uspjela proći preko neke cijene na kojoj su koncentrirani nalozi unutar knjige naloga potrebno je ili da oduzimatelji likvidnosti jednakom količinom izvrše svoje naloge na toj cijeni (u dovoljno kratkom vremenskom razdoblju kako bi spriječili dodatni rast koncentracije) ili da održavatelji tržišta ponište svoje naloge te smanje koncentriranu količinu.



Slika 2.3. Snimka zaslona s internetske stranice: tensorcharts.com.
OHLC prikaz cijene i koncentracija naloga unutar knjige naloga na financijskom instrumentu BTCUSDT.

- Uvijek postoji razlika (engl. spread) između najbolje prodajne i najbolje kupovne cijene. Koliko god bila mala količina, kada oduzimatelj likvidnosti odluči trgovati on će uvijek pomaknuti zadnju cijenu (engl. last price) barem za promjenu cijene do najboljeg kupovnog, odnosno prodajnog naloga unutar knjige naloga. Unutar knjige naloga također postoje razlike između ostalih susjednih cijena. Na određenim

cijenama koncentracija količine naloga je veća, dok na određenim cijenama uopće nema naloga.

- Razlika između najboljih cijena može se mijenjati neovisno o izvršenim transakcijama. Zadnja cijena može *ispasti* iz razlike što implicira da postoje trenutci u kojima prodajna transakcija (od strane oduzimatelja likvidnosti) diže cijenu i trenutci u kojima kupovna transakcija spušta cijenu. Povećanje ili smanjenje razlike između najboljih cijena u odnosu na zadnju izvršenu transakciju, te *ispadanje* zadnje cijene iz razlike možemo tumačiti kao pristranost održavatelja tržišta.

Order Book		Recent Trades
Price(USDT)	Qty(ETH)	Total(E...)
3,343.20	0.01	5.60
3,343.19	3.00	5.59
3,343.17	0.45	2.59
3,343.16	0.15	2.14
3,343.10	0.83	1.99
3,343.05	0.01	1.16
3,343.04	0.05	1.15
3,343.00	1.10	1.10
↑ 3,342.80 🚩 3,342.53		
3,342.99	54.40	54.40
3,342.96	0.27	54.67
3,342.92	6.81	61.48
3,342.91	18.87	80.35
3,342.88	0.27	80.62
3,342.87	3.00	83.62
3,342.86	5.00	88.62
3,342.83	2.50	91.12

Slika 2.4. Snimka zaslona s internetske stranice: bybit.com.

Knjiga naloga: cijene bez količine; razlika između najbolje kupovne i najbolje prodajne cijene se pomakla iznad cijene zadnje izvršene transakcije.

- Koliko će oduzimatelj likvidnosti pomaknuti cijenu ovisi o kombinaciji količine naloga kojeg postavlja i stanju knjige naloga. U trenutcima kada je količina naloga unutar knjige naloga slabo koncentrirana oko zadnje cijene, relativno mala količina od strane oduzimatelja likvidnosti može značajno pomaknuti cijenu.
- Oduzimatelji likvidnosti su potrebni da bi se pomaknula zadnja cijena. U odre-

denim razdobljima frekventnost naloga oduzimatelja likvidnosti je veća, odnosno manja. To može, ali i ne mora utjecati na izvršenu količinu. Frekventnost pojave oduzimatelja likvidnosti u kombinaciji s velikom količinom možemo tumačiti kao pristranost oduzimatelja likvidnosti.

Obraćajući pažnju na navedene pojave prilikom prikaza signala zadnjih izvršenih transakcija unosimo dodatnu informaciju o cijeni (u odnosu na OHLC svijeće). Možemo očekivati da će takav prikaz signala dubokoj neuronskoj mreži pružiti potpuniji uvid u različita tržišna stanja.

2.2.1. Prostor dinamike

Da bismo što potpunije opisali informaciju koju pruža signal zadnjih izvršenih transakcija definirajmo prostor dinamike dvostruke akcije. Unutar takvog prostora pozicija svake transakcije sadrži informaciju o različitim tržišnim stanjima te njihovim uzrocima. Pojam dinamike koristimo zbog analogije s dinamikom u fizici. Dinamika je grana mehanike koja povezuje gibanje tijela sa silama koje djeluju na to tijelo; u ovom slučaju količina transakcije oduzimatelja likvidnosti (sila) djeluje na cijenu (tijelo) određenom frekvencijom te se ona pomiče s obzirom na stanje knjige naloga.

Svaku izvršenu transakciju možemo pozicionirati u prostor $\mathbb{N}_+ \times S_+ \times P_+ \times \mathbb{N}_+ \times \{-1, +1\}$ gdje osi redom predstavljaju:

1. **Broj transakcije:** $n^i \in \mathbb{N}_+$ (brojimo od prve transakcije za određeni financijski instrument na određenoj burzi)
2. **Količina transakcije:** $s^i \in S_+$
 S_+ - diskretni skup s vrijednostima količina koje je moguće trgovati (pretpostavljamo da postoji minimalna količina trgovanja i diskretni korak između različitih količina)
3. **Cijena na kojoj se izvela transakcija:** $p^i \in P_+$
 P_+ - diskretni skup s cijenama koje odgovaraju najmanjim mogućim promjenama
4. **Vremenska oznaka transakcije** (engl. timestamp): $t^i \in \mathbb{N}_+$

5. **Zastavica** $bt^i \in \{-1, +1\}$:

- -1 ako je oduzimatelj likvidnosti postavio prodajni nalog
- $+1$ inače

Stoga, jedna transakcija \vec{T}^i može biti zapisana kao peterodimenzionalan vektor:

$$\vec{T}^i = \begin{bmatrix} n^i & s^i & p^i & t^i & bt^i \end{bmatrix}^T \quad (2.1)$$

S oznakom T^N označavamo uzorak od N slijednih transakcija \vec{T}^i . Neka je $f_{diff}(\vec{T}^{i-1}, \vec{T}^i) : (\mathbb{N}_+ \times S_+ \times P_+ \times \mathbb{N}_+ \times \{-1, +1\})^2 \rightarrow \mathbb{N}_+ \times D \times P_{perc} \times \mathbb{N}_+$, gdje je D diskretni skup s vrijednostima koje je moguće trgovati izraženim u dolarima, a P_{perc} diskretni skup vrijednosti postotnih promjena cijene s obzirom na najmanju moguću promjenu cijene. $f_{diff}(T^{i-1}, T^i)$ računamo na sljedeći način:

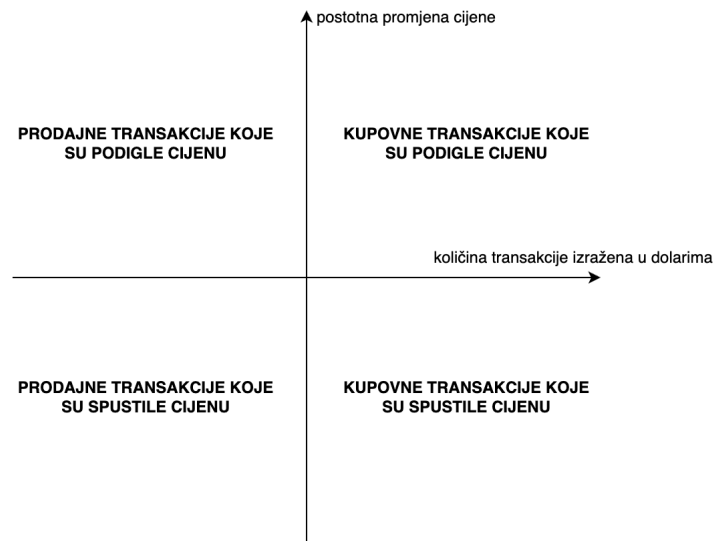
$$f_{diff}(\vec{T}^{i-1}, \vec{T}^i) = \begin{bmatrix} n^i & s^i \cdot p^i \cdot bt^i & \frac{p^i - p^{i-1}}{p^{i-1}} & t^i - t^{i-1} \end{bmatrix}^T \quad (2.2)$$

Primjenom 2.2 na susjedne transakcije \vec{T}^{i-1}, \vec{T}^i (odbacujući prvu transakciju jer ona nema prethodnika), dobivamo skup T_{diff}^N unutar kojeg za svaku transakciju sada imamo sljedeće:

1. **Broj transakcije** - zadržavamo zbog uređenosti izvršavanja transakcija
2. **Impuls na sustav** uzrokovan izvršenom transakcijom izražen u dolarima (negativan za prodajne transakcije oduzimatelja likvidnosti, pozitivan za kupovne)
3. **Postotna promjena cijene** kao posljedica impulsa
4. **Razlika vremenskih oznaka** dvije susjedne transakcije (uvijek pozitivna vrijednost)

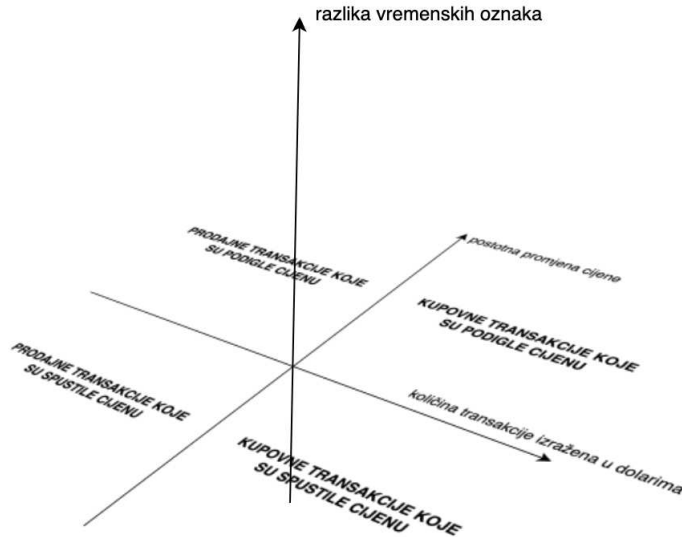
Ako ignoriramo podatak o razlici vremenskih oznaka dvije susjedne transakcije tada svaki izlaz funkcije $f_{diff}(\vec{T}^{i-1}, \vec{T}^i)$ može biti postavljen u dvodimenzionalnom prostoru gdje je x-os količina transakcije izražen u dolarima (impuls na sustav), a y-os postotna promjena cijene koja je posljedica tog impulsa. Takav prostor je robustan u prikazu stanja tržišta kada se događa pomak razlike najbolje kupovne i prodajne cijene u odnosu na zadnju izvršenu transakciju (što posljedično potencijalno dovodi do prodajnih transakcija

koje dižu cijenu i kupovnih transakcija koje spuštaju cijenu; pristranost održavatelja tržišta). Također, takav prostor je robustan u prikazu stanja tržišta kada izvršene transakcije relativno male količine rade velike postotne promjene cijene (stanja tržišta kada je knjiga naloga oko zadnje cijene relativno rijetka s obzirom na količinu), te prikazu stanja kada izvršene transakcije relativno velike količine rade male postotne promjene cijene (stanja tržišta kada je knjiga naloga oko zadnje cijene relativno gusta s obzirom na količinu).



Slika 2.5. Prostor dinamike dvostruke aukcije (bez razlike vremenskih oznaka).

Ako u razmatranju zadržimo podatak o razlici vremenskih oznaka dvije susjedne transakcije \vec{T}^{i-1} i \vec{T}^i tada svaki izlaz funkcije $f_{diff}(\vec{T}^{i-1}, \vec{T}^i)$ može biti postavljen u trodimenzionalni prostor gdje dodatna os-z sadrži informaciju o frekventnosti transakcija. Takav prostor je robustan u prikazu stanja tržišta kada velikom ili malom frekvencijom oduzimatelji likvidnosti izvršavaju transakcije. Kombinacija podataka o frekventnosti izvršavanja naloga oduzimatelja likvidnosti i izvršenih količina pruža uvid u pristranost oduzimatelja likvidnosti. Vrijednosti na z-osi uvijek su pozitivne (poštujemo uređenost transakcija).



Slika 2.6. Prostor dinamike dvostruke aukcije.

Ovakvim prikazom uspjeli smo prikazati stanja tržišta koja smo naveli u opažanjima unutar prošlog potpoglavlja. Takav prikaz zasigurno sadrži više informacija od standardnog OHLC prikaza te posljedično smanjuje efekt aliasinga. Svaki skup transakcija T^N na kojeg primijenimo funkciju $f_{\text{diff}}(\vec{T}^{i-1}, \vec{T}^i)$ možemo pozicionirati unutar ovog prostora te će ta reprezentacija nuditi informaciju o lokalnom stanju tržišta unutar nekog vremenskog intervala. Ako primijenimo standardni način uzorkovanja skupova transakcija (fiksni vremenski okvir) možemo očekivati da broj transakcija unutar uzorka T^N neće biti stalan. Neka je T^{N^l} skup transakcija gdje je N^l broj transakcija u l -tom uzorku. Označimo s N^{l_1} broj transakcija unutar uzorka fiksnog vremenskog okvira u trenucima visoke frekventnosti trgovanja (otvaranje burze ili dolazak izvještaja), te s N^{l_2} broj transakcija unutar uzorka fiksnog vremenskog okvira u trenucima niske frekventnosti. Podaci pokazuju da $N^{l_1} \gg N^{l_2}$. To nam govori da su naši uzorci pristrani s obzirom na vremenski okvir, samim time i vrijeme uzorkovanja. Također, uzorkovanje s obzirom na fiksni okvir u istrgovanoj količini dovesti će do istog efekta. Da bi odstranili navedenu pristranost primjenjujemo uzorkovanje fiksnog transakcijskog okvira. Takva metoda uzorkovanja osigurava da svaki uzorkovani skup transakcija T^N sadrži jednak broj transakcija. U tablici 2.1. predstavljene su glavne razlike između standardnog OHLC prikaza i prikaza u prostoru dinamike.

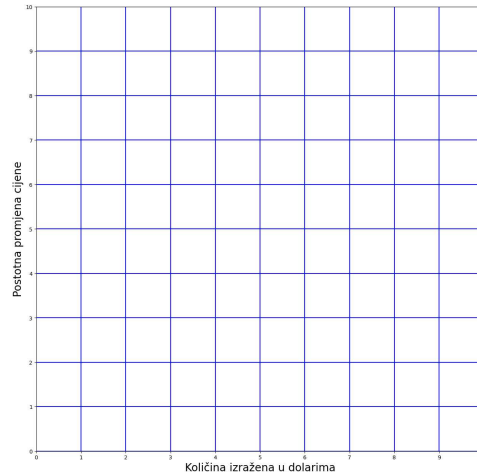
	OHLC prikaz	prikaz u prostoru dinamike
način uzorkovanja transakcija	fiksni vremenski okvir	fiksni transakcijski okvir
podaci o uzorku transakcija	$[p^0 \max(p^i) \min(p^i) p^{N-1}]^T$	$f_{diff}(\vec{T}^{i-1}, \vec{T}^i), i \in [1, N - 1]$
dimenzionalnost	vektor: 4 dimenzije	matrica: $(N - 1) \times 3$

Tablica 2.1. Usporedba OHLC prikaza i prikaza u prostoru dinamike. Podrazumijeva se da redci matrice (prostor dinamike) prate uređenost transakcija.

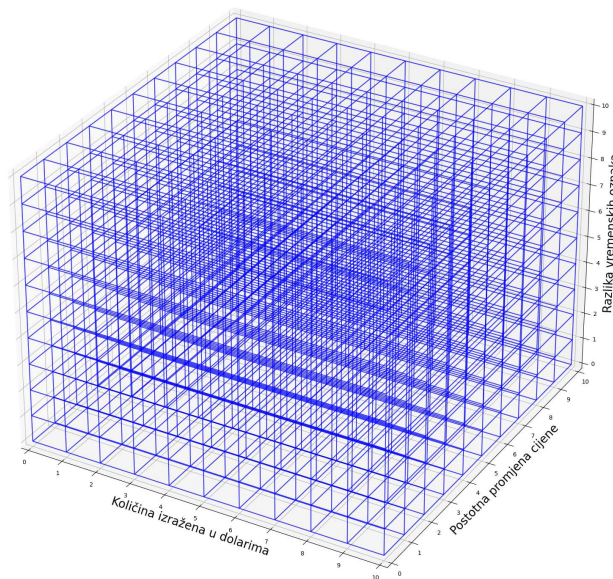
2.2.2. Prikaz skupa transakcija u formi histograma

Pozicioniranje jedne transakcije u prostoru dinamike nam opisuje stanje tržišta u trenutku kada je izvršena ta transakcija. Pozicioniranje skupa (susjednih) transakcija opisuje stanje tržišta unutar vremenskog intervala u kojem su transakcije uzorkovane. Jedna od posljedica postavki sustava dvostruke aukcije i latencije internetske veze (između korisnika i burze) je stohastičnost sustava. Budući da prije postavljanja naloga ne možemo biti apsolutno sigurni kojim točno redoslijedom će se izvesti nalozi, jako je teško predvidjeti obrasce dinamike koji se pojavljuju unutar transakcija. Međutim, ako promatramo relativno mali transakcijski okvir, možemo reći da nam sam poredak transakcija nije bitan, već nam je bitna distribucija transakcija u prostoru dinamike. Distribucija, odnosno vjerojatnosni histogram je prikaz podataka koji je permutacijski invarijantan zbog čega je *otporan* na stohastičnost redoslijeda izvršavanja transakcija. Takav prikaz podataka isključivo mjeri koncentraciju transakcija unutar nekog dijela prostora, a to je upravo ono što želimo dobiti. Još jedna motivacija za uvođenje histograma je podešavanje rezolucije detekcije različitih stanja tržišta. Ako svaka od osi dinamike ima svoju najmanju moguću promjenu (istrgovana količina izražena u dolarima, postotna promjena cijene i razlika između vremenskih oznaka) to znači da već postoji distinkcija između različitih diskretnih stanja, međutim, rezolucija distinkcije je jako velika. Ono što bi htjeli postići opisat ćemo jednostavnim primjerom: ako imamo dva slučaja gdje u prvom slučaju \$100 pomiče cijenu za 0.001% te je razlika vremenskih oznaka 100ms i drugi slučaj gdje \$105 pomiče cijenu za 0.001001% te je razlika vremenskih oznaka 980ms - ta dva slučaja bi htjeli staviti u istu kategoriju (stanje tržišta) s obzirom na to da se radi o jako maloj semantičkoj različitosti između ta dva tržišna stanja. U svrhu grupiranja sličnih tržišnih stanja na različite načine mogu se definirati razredi histograma. Konkretno, unutar ovog rada, prvo radimo mjerenje na kompletom skupu transakcija (od trenutka kada je financijski

instrument objavljen na burzi) tako da se izračunaju određene percentilne vrijednosti svake od osi u prostoru dinamike te se zatim principom rešetke pronađu združena stanja između različitih percentilnih kategorija svake od osi.



Slika 2.7. Razredi histograma: istrgovana količina izražena u dolarima - postotna promjena cijene. Na ovom primjeru razredi su definirani svakim desetim percentilom.



Slika 2.8. Razredi histograma: istrgovana količina izražena u dolarima - postotna promjena cijene - razlika vremenskih oznaka.

Nakon što ovako reprezentiramo podatke, skup transakcija T^N više nisu podaci u strukturi matrice dimenzionalnosti $(N-1) \times 3$ (prostor dinamike) već su u strukturi tenzora dimenzionalnosti $D_{\text{kategorija}} \times P_{\text{kategorija}} \times TS_{\text{kategorija}}$, gdje je $D_{\text{kategorija}}$ skup percentilnih kategorija količine izvršenih transakcija izražene u dolarima, $P_{\text{kategorija}}$ skup percentilnih kategorija postotne promjene cijene, a $TS_{\text{kategorija}}$ skup percentilnih kategorija razlike

vremenskih oznaka između transakcija. Konkretno, ako uzmemo svaki deseti percentil, bilo koji skup transakcija T^N možemo prikazati kao tenzor (vjerojatnosni histogram) dimenzija $9 \times 9 \times 9$.

Nakon pretvaranja skupa transakcija T^N u vjerojatnosni histogram osigurali smo da nam je ulaz u model tenzor koji je uvijek jednake dimenzionalnosti (neovisno o metodi uzorkovanja skupa transakcija T^N), prikaz podataka je permutacijski invarijantan te su vrijednosti tenzora normalizirane na način da je suma svih vrijednosti unutar tenzora jednaka 1 jer se radi o vjerojatnosnom histogramu.

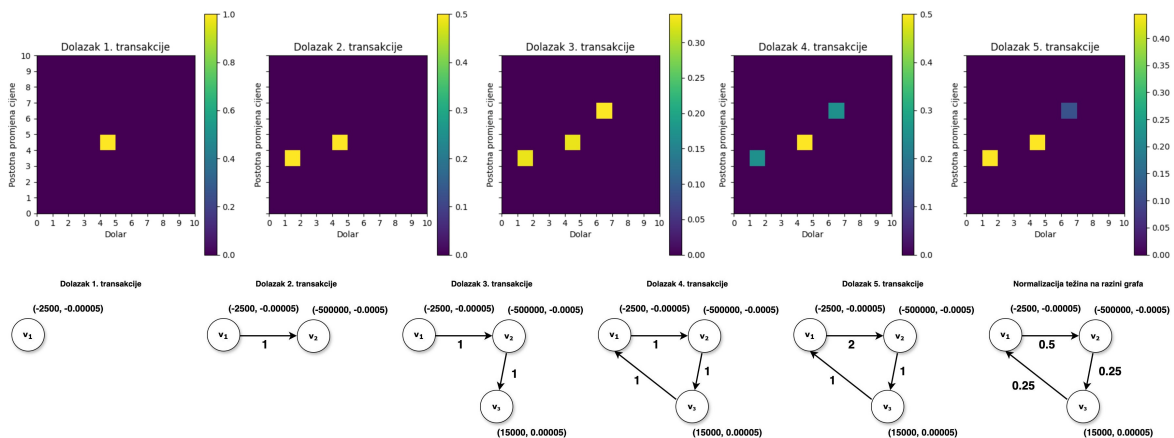
2.2.3. Izgradnja grafa: motivacija i postupak

Jedan od nedostataka prikaza skupa transakcija kao vjerojatnosnog histograma je potencijalno veliki broj neinformativnih značajki. Primjenjujući uzorkovanje prostora koje generira jako puno razreda u trenucima kada je koncentracija transakcija isključivo unutar malog broja razreda, svi ostali razredi imat će vjerojatnost jednaku 0. Zbog toga želimo strukturu podataka koja nudi mogućnost prikaza histograma u rijetkoj formi kako bi izbjegli ulaz u mrežu koji nudi puno neinformativnih značajki. Još jedan nedostatak je što histogram (kojeg ako promatramo kao sliku) modeliramo konvolucijskim neuronskim mrežama koje se nameću kao standardni način za ovakvu strukturu podataka. Konvolucijske neuronske mreže pretpostavljaju topologiju rešetke među elementima ulaza (u ovom slučaju razredima histograma). Receptivno polje ulaza određenog sloja ovisi o veličini jezgre konvolucije svih slojeva prije i trenutnog sloja. Jezgra konvolucije uzima u obzir samo usko susjedstvo oko trenutnog elementa uvodeći pristranost da naučene značajke trenutnog elementa ovise samo o neposrednom susjedstvu unutar topologije rešetke. Takva pristranost ne može modelirati odnos među razredima koji su na rešetki međusobno udaljeni daleko sve do onog sloja mreže čije receptivno polje uzima u obzir ta dva razreda. Razmatrajući ekstremni slučaj gdje želimo modelirati odnos među razredom koji je u donjem lijevom uglu histograma i razredom u gornjem desnom uglu histograma, konvolucijska neuronska mreža moći će modelirati takav odnos tek u onome sloju gdje uzima u obzir kompletni ulaz. Budući da svaki konvolucijski sloj smanjuje prostornu finoću (smanjujući prostornu dimenzionalnost ulaza) i sprema informaciju u semantičke kanale, u slučaju rijetkog histograma prostorna finoća ranih slojeva ostaje neiskorištena jer modelira jednostavan slučaj gdje je većina razreda jednaka 0. Također, konvolucija

nije najbolji izbor zbog pretpostavki u načinu agregiranja informacija susjednih razreda (želimo definirati susjednost preko uređenosti transakcija, a ne preko rešetke).

Kako bi modelirali stvaran odnos među razredima histograma, predstaviti ćemo histogram kao usmjereni težinski graf. Takav graf za vrhove posjeduje stanja tržišta (razrede histograma) koja su se pojavila unutar uzorka transakcija. Usmjereni bridovi predstavljaju prelazak iz stanja u stanje, dok težine služe kao brojač koliko se puta dogodio prelazak iz određenog stanja u određeno stanje. Ako sve težine podijelimo s $N - 1$ (N je broj transakcija unutar uzorka) težine se skaliraju na interval $[0, 1]$ tako da vrijedi $\sum_e w_e = 1$. Također, uzorak transakcija zadaje nenormalizirane vjerojatnosne distribucije prelaska iz određenog stanja u određeno stanje.

Na taj način dobili smo rijetki prikaz histograma kojemu topologija među razredima nije definirana rešetkom već stvarnim izmjenama tržišnih stanja unutar nekog uzorka transakcija.



Slika 2.9. Prikaz izgradnje grafa. Transakcije koje su dolazile redom su pripadale razredima: $(4.5, 4.5)$, $(1.5, 3.5)$, $(6.5, 6.5)$, $(4.5, 4.5)$, $(1.5, 3.5)$.

Algoritam 1 Pseudokod izgradnje grafa iz skupa transakcija u prostoru dinamike dvostruke aukcije

Ulaz: Skup transakcija u prostoru dinamike dvostruke aukcije $T_{diff}^N \in (N - 1) \times 3$ i granice razreda histograma.

Izlaz: Usmjereni težinski graf \mathcal{G} (matrica značajki vrhova X_v i matrica susjedstva A).

Inicijaliziraj: skup značajki vrhova $X_v = \emptyset$ i skup bridova $\mathcal{E} = \emptyset$.

za $i = 1$ **do** N

1. T_{diff}^i dodijeli pripadajući razred histograma.
2. Srednje vrijednosti intervala na svakoj od osi koje obuhvaća pridijeljeni razred su značajke trenutnog vrha. Ako se vrh s tim značajkama ne nalazi u X_v , dodaj ga.
3. Dodaj usmjereni brid koji povezuje vrh generiran od prethodne transakcije i trenutni vrh (ako takav brid već postoji njegovu težinu povećaj za 1).

kraj za

Normaliziraj težine bridova na razini grafa: $w_e \leftarrow \frac{w_e}{N-1}, \forall e \in \mathcal{E}$

Primijetimo da suma normaliziranih težina ulaznih bridova za svaki vrh i vjerojatnosti razreda histograma kojeg predstavlja taj vrh nisu jednake (zbog toga što prva transakcija nema prethodnika). Ako prilikom dodavanja prvog vrha dodamo usmjereni brid koji kreće iz njega i završava na njemu, onda ćemo nakon normalizacije (normaliziramo dijeleći s N) da suma normaliziranih težina ulaznih bridova za svaki vrh odgovara vjerojatnosti razreda histograma kojeg predstavlja.

3. Neuronske mreže za grafove

Duboke neuronske mreže poput povratnih ili konvolucijskih postigle su veliki uspjeh u obradi i modeliranju euklidskih podataka poput nizova (jednodimenzionalni podaci s topologijom lanca) i slika (dvodimenzionalni podaci s topologijom rešetke). Međutim, postoji jako puno skupova podataka koje želimo modelirati takvih da se najbolje opisuju kompleksnim strukturama poput grafa ili mnogostrukosti (engl. manifold). Neke od primjena dubokih neuronskih mreža nad takvim skupovima podataka su modeliranje interakcija na društvenim mrežama, sustavi za preporuku, modeliranje kemijskih struktura, modeliranje izvođenja automata, itd...

3.1. Notacija i svojstva

Graf koji dolazi na ulaz u duboku neuronsku mrežu za grafove može biti zapisan kao $\mathcal{G} = (\mathcal{V}, \mathcal{E}, X_{\mathcal{V}}, X_{\mathcal{E}})$:

1. \mathcal{V} je skup vrhova. Ukupan broj vrhova u \mathcal{G} označimo s $N = |\mathcal{V}|$.
2. \mathcal{E} je skup bridova (skup uređenih parova).
3. $X_{\mathcal{V}} = \{\vec{x}_v | v \in \mathcal{V}\}$ su značajke vrhova. $\vec{x}_v \in \mathbb{R}^{D_{\mathcal{V}}}$ gdje je $D_{\mathcal{V}}$ dimenzionalnost ulaznih značajki vrhova. $X_{\mathcal{V}} \in N \times D_{\mathcal{V}}$, matrica $X_{\mathcal{V}}$ je složena od vektor redaka \vec{x}_v^T .
4. $X_{\mathcal{E}} = \{\vec{x}_{uv} | (u, v) \in \mathcal{E}\}$ su značajke bridova. $\vec{x}_{uv} \in \mathbb{R}^{D_{\mathcal{E}}}$ gdje je $D_{\mathcal{E}}$ dimenzionalnost ulaznih značajki bridova. $X_{\mathcal{E}} \in N \times N \times D_{\mathcal{E}}$ je tenzor koji za svaki par vrhova posjeduje značajke brida koji ih povezuje. Ako je $D_{\mathcal{E}} = 1$ (*standardni težinski graf*) onda umjesto $X_{\mathcal{E}}$ možemo koristiti matricu susjedstva $A \in N \times N$.

Dodatna notacija:

- $H^{(l)} = \{\vec{h}_v^{(l)} | v \in \mathcal{V}\}$ označava matricu skrivenih značajki vrhova l -tog sloja. $\vec{h}_v^{(l)} \in \mathbb{R}^{\delta^{(l)}}$ gdje $\delta^{(l)}$ označava dimenzionalnost skrivenih značajki l -tog sloja. $H^{(l)} \in N \times \delta^{(l)}$, matrica $H^{(l)}$ je složena od vektor redaka \vec{h}_v^T .

Dogovorno vrijedi $H^{(0)} = X_{\mathcal{V}}$ ($\delta^{(0)} = D_{\mathcal{V}}$). $H_v^{(l)}$ podrazumijeva vektor-redak sa značajkama vrha v u sloju l : $H_v^{(l)} = (\vec{h}_v^{(l)})^T$.

- $\mathcal{N}(v)$ označava funkciju susjedstva; $\mathcal{N} : \mathcal{V} \rightarrow 2^{\mathcal{V}}$. Funkcija susjedstva svakom vrhu v pridružuje podskup skupa \mathcal{V} koji predstavlja susjede vrha v .

Osnovna ideja neuronskih mreža za grafove je iterativno ažurirati značajke vrhova integrirajući informaciju o susjednim vrhovima i međusobnoj vezi. Općeniti postupak koji provode neuronske mreže za grafove možemo zapisati s dva koraka: [1]

1. **Agregiraj** (engl. aggregate): za svaki vrh v unutar grafa \mathcal{G} agregiraj informacije svojih susjeda $u \in \mathcal{N}(v)$
2. **Integriraj** (engl. combine): za svaki vrh v unutar grafa \mathcal{G} ažuriraj značajke integrirajući agregiranu informaciju svih susjeda iz prethodnog koraka.

Poopćeno gledano radi se o algoritmu propagacije poruka (engl. message passing algorithm). Unutar koraka agregiraj svaki vrh prima *poruke* (informaciju) od susjednih vrhova te ih agregira funkcijom agregacije f_M . Unutar koraka integriraj svaki vrh ažurira svoje značajke funkcijom ažuriranja f_U integrirajući informaciju iz susjedstva.

Kako bi iz značajki vrhova $H_{\mathcal{V}}^{(l)}$ mogli izračunati reprezentaciju cijelog grafa $\vec{h}_{\mathcal{G}}$ potrebno je izračunati funkciju očitavanja (engl. readout function) $f_R^{(l)} : \mathbb{R}^{N \times \delta^l} \rightarrow \mathbb{R}^{\delta^l}$. Funkcija očitavanja računa se nad značajkama vrhova u l -tom sloju agregirajući značajke svih vrhova u vektor $\vec{h}_{\mathcal{G}}^{(l)}$ koji predstavlja reprezentaciju grafa na izlazu l -tog sloja.

Algoritam 2 Pseudokod neuronske mreže za grafove s L slojeva dubine i funkcijom očitavanja u zadnjem sloju

Ulaz: Matrica značajki vrhova $X_{\mathcal{V}}$ i matrica značajki bridova $X_{\mathcal{E}}$.

Izlaz: Matrica reprezentacija vrhova $H_{\mathcal{V}}^{(L)}$ i reprezentacija cijelog grafa $\vec{h}_{\mathcal{G}}$.

Inicijaliziraj: $H^{(0)} = X_{\mathcal{V}}$

za $l = 1$ do L

Agregiraj:

$$\vec{a}_v^{(l)} = f_M^{(l)}(\{(\vec{h}_v^{(l-1)}, \vec{h}_u^{(l-1)}, \vec{x}_{uv}) : u \in \mathcal{N}(v)\}) \quad (3.1)$$

Integriraj:

$$\vec{h}_v^{(l)} = f_U^{(l)}(\vec{h}_v^{(l-1)}, \vec{a}_v^{(l)}) \quad (3.2)$$

kraj za

Funkcija očitavanja:

$$\vec{h}_{\mathcal{G}} = f_R^{(L)}(\{\vec{h}_v^{(L)} | v \in \mathcal{V}\}) \quad (3.3)$$

Svaki graf \mathcal{G} s N vrhova možemo zapisati s $N!$ različito permutiranih matrica susjedstva A (poopćeno vrijedi i za $X_{\mathcal{E}}$), i pripadnih matrica značajki vrhova $X_{\mathcal{V}}$. Odabrani poredak vrhova unutar A i $X_{\mathcal{V}}$ nije bitan, bitno je da vrhovi unutar A i $X_{\mathcal{V}}$ međusobno poštuju poredak. Formalno to iskazujemo na sljedeći način: neka su matrice $X_{\mathcal{V}_{\pi}} \in R^{N \times D_{\mathcal{V}}}$ i $A_{\pi} \in R^{N \times N}$ zapisane u poretku vrhova $\pi \in \Pi$ gdje je Π skup svih $N!$ permutacija nad \mathcal{V} . Definirajmo permutacijsku matricu P koja permutira vrhove iz poretka π u poredak π' kao $P_{\pi \rightarrow \pi'} = (p_{ij}) \in \{0, 1\}^{N \times N}$ s vrijednostima $p_{ij} = 1$ ako $\pi(i) = \pi'(j)$ i $p_{ij} = 0$ inače, [2]. Budući da su grafovi permutacijski invarijantni vrijedi:

$$\mathcal{G}_{\pi} = \mathcal{G}(X_{\mathcal{V}_{\pi}}, A_{\pi}) = \mathcal{G}(P_{\pi \rightarrow \pi'} X_{\mathcal{V}_{\pi}}, P_{\pi \rightarrow \pi'} A_{\pi} P_{\pi \rightarrow \pi'}^T) = \mathcal{G}(X_{\mathcal{V}_{\pi'}}, A_{\pi'}) = \mathcal{G}_{\pi'} \quad (3.4)$$

Dakle, svako preslikavanje unutar neuronske mreže za grafove mora imati svojstvo permutacijske invarijantnosti: koju god permutaciju vrhova π odaberemo na ulazu, skrivene reprezentacije unutar mreže moraju biti jednake. Funkcija očitavanja f_R također mora biti permutacijski invarijantna (npr. zbroj ili srednja vrijednost).

3.2. Konvolucijske neuronske mreže za grafove

Zbog svoje efikasnosti, konvolucijske neuronske mreže za grafove postale su najpopularnija arhitektura neuronskih mreža za grafove. Ideja konvolucijskih neuronskih mreža za

grafove je poopćiti standardnu konvolucijsku mrežu tako da može procesirati podatke s proizvoljnom topologijom. Osnovna postavka konvolucijskih neuronskih mreža za grafove uzima za pretpostavku da je $D_\varepsilon = 1$ (u računu koristimo matricu susjedstva A) te da je graf neusmjeren. Određenim modifikacijama možemo poopćiti postupak kako bi procesirali usmjerene grafove, [3]. Unutar svakog sloja konvolucijske neuronske mreže za grafove značajke svakog vrha ažuriraju se sljedećim izrazom:

$$H^{(l)} = f_A^{(l)}(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l-1)}W^{(l)}) \quad (3.5)$$

gdje je $\tilde{A} = A + I$ matrica susjedstva neusmjerenog grafa \mathcal{G} ($I \in \mathbb{R}^{N \times N}$ je matrica identiteta). Matrica identiteta I dodaje bridove koji povezuju vrhove same sa sobom kako bi se omogućilo očuvanje trenutnih informacija koje vrh posjeduje. $\tilde{D} \in \mathbb{R}^{N \times N}$ je dijagonalna matrica sa stupnjevima svakog od vrhova takva da $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$, $\forall i \in N$. $f_A^{(l)}$ je odabrana aktivacijska funkcija. $W^{(l)} \in \mathbb{R}^{\delta^{(l-1)} \times \delta^{(l)}}$ je matrica parametara koji se uče prilikom optimizacije modela. Primijetimo da je transformacija permutacijski invarijantna. Dodatno, možemo raščlaniti izraz 3.5 na korake agregiraj i integriraj. Konkretno, za vrh v vrijedi:

1. Agregiraj:

$$\tilde{a}_v^{(l)} = f_A^{(l)}\left(\sum_{u \in \{\mathcal{N}(v) \cup v\}} \frac{\tilde{A}_{vu}}{\sqrt{\tilde{D}_{vv}\tilde{D}_{uu}}} H_u^{(l-1)} W^{(l)}\right) \quad (3.6)$$

2. Integriraj:

$$H_v^{(l)} = f_A^{(l)}\left(\sum_{u \in \mathcal{N}(v)} \frac{A_{vu}}{\sqrt{\tilde{D}_{vv}\tilde{D}_{uu}}} H_u^{(l-1)} W^{(l)} + \frac{1}{\tilde{D}_v} H_v^{(l-1)} W^{(l)}\right) \quad (3.7)$$

U koraku agregiranja vidimo da je funkcija definirana kao nelinearno preslikavanje težinskog prosjeka značajki susjednih vrhova. Težina susjednog vrha u određena je težinom brida (v, u) (te je dodatno normalizirana s obzirom na stupanj vrhova). U koraku integriranja vidimo da je funkcija definirana kao zbroj agregiranih značajki susjednih vrhova i vlastitih značajki normaliziranih vlastitim stupnjem vrha (lijevi član u zbroju predstavlja rezultat koraka agregiranja bez uzimanja u obzir povezanosti vrha sa samim sobom), [1].

3.3. Neuronske mreže za grafove s pažnjom

Neuronske mreže za grafove s pažnjom (engl. graph attention networks) možemo promatrati kao nadogradnju konvolucijskih neuronskih mreža za grafove. Kod konvolucijskih neuronskih mreža za grafove važnost veze ciljnog vrha v sa susjedom u određena je težinom brida A_{vu} koji ih povezuje (normaliziranim s obzirom na stupanj vrhova). Budući da uvijek postoji šum unutar skupa podataka, za očekivati je da A_{vu} neće uvijek dobro reprezentirati važnost veze vrhova v i u . Ideja neuronskih mreža za grafove s pažnjom je da se dodatno nauče važnosti veza susjeda bazirane na mehanizmu pažnje (engl. attention mechanism), [1]. Sloj neuronske mreže za grafove s pažnjom možemo izraziti sljedećom transformacijom:

$$H_v^{(l)} = f_A\left(\sum_{u \in \mathcal{N}(v)} \alpha_{vu} H_u^{(l-1)} W^{(l)}\right) \quad (3.8)$$

Ako transformaciju usporedimo s jednadžbom 3.5 vidimo da se razlikuje u tome što je umnožak $\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$ zamijenjen s koeficijentom pažnje α_{vu} . S obzirom na to da je zamijenjen član \tilde{A} , pretpostavlja se da je svaki vrh također sam sebi susjed. Računanje α_{vu} možemo općenito obaviti na sljedeći način:

- Računanje koeficijenta pažnje definira se nad vrhovima. Mehanizam pažnje primjenjuje se za svaki par susjednih vrhova funkcijom $a : \mathbb{R}^{\delta^{(l)}} \times \mathbb{R}^{\delta^{(l)}} \rightarrow \mathbb{R}$:

$$e_{vu} = a(H_v^{(l-1)} W_a^{(l)}, H_u^{(l-1)} W_a^{(l)}) \quad (3.9)$$

gdje e_{vu} predstavlja važnost veze vrha v i u . $W_a^{(l)} \in \delta^{(l-1)} \times \delta^{(l)}$ je matrica težina za mehanizam pažnje u sloju l .

- Koeficijent pažnje α_{vu} računamo primjenom *Softmax* funkcije nad važnostima veze e kako bi za svaki vrh dobili distribuciju važnosti veze s njegovim susjedima: [1]

$$\alpha_{vu} = \text{Softmax}_u(e_{vu}) = \frac{\exp(e_{vu})}{\sum_{k \in \mathcal{N}(v)} \exp(e_{vk})} \quad (3.10)$$

- Kod neuronskih mreža za grafove s pažnjom mehanizam pažnje je definiran kao unaprijedni sloj neuronske mreže koji uključuje linearnu transformaciju s vektorom

parametara $W_{a2} \in R^{2 \cdot \delta^{(l)} \times 1}$ i LeakyReLU aktivacijsku funkciju:

$$\alpha_{vu} = \frac{\exp(\text{LeakyReLU}([H_v^{(l-1)} W_a^{(l)} || H_u^{(l-1)} W_a^{(l)}] W_{a2}))}{\sum_{k \in \mathcal{N}(i)} \exp(\text{LeakyReLU}([H_v^{(l-1)} W_a^{(l)} || H_k^{(l-1)} W_a^{(l)}] W_{a2}))} \quad (3.11)$$

gdje je $||$ operator konkatencije dva vektora, [4]. Nakon izračunatih koeficijenta pažnje primjenjujemo izraz 3.8 kako bi izračunali značajke $H_v^{(l)}$.

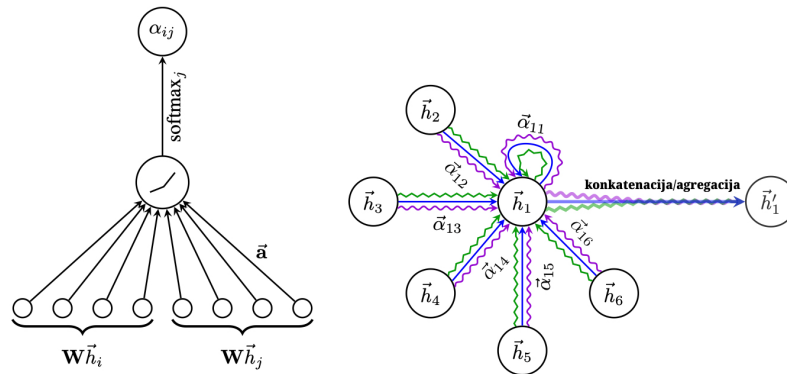
Dodatno, umjesto jednog mehanizma pažnje možemo primijeniti višestruku pažnju (engl. multi-head attention):

$$H_v^{(l)} = \bigparallel_{t=1}^T f_A \left(\sum_{u \in \mathcal{N}(i)} \alpha_{vu}^t H_u^{(l-1)} W^t \right) \quad (3.12)$$

gdje je T ukupan broj mehanizama pažnje unutar sloja l , α_{vu}^t koeficijent pažnje t -tog mehanizma pažnje, a W_t matrica težina linearne transformacije t -tog sloja pažnje. U izvornom znanstvenom radu svi mehanizmi pažnje se na izlazu zadnjeg sloja agregiraju sumom kako bi se dobila potrebna dimenzionalnost izlaza mreže:

$$H_v^{(l)} = f_A \left(\frac{1}{T} \sum_{t=1}^T \sum_{u \in \mathcal{N}(i)} \alpha_{vu}^t H_u^{(l-1)} W^t \right) \quad (3.13)$$

Operacija agregiranja sumom može biti zamijenjena bilo kojom drugom funkcijom agregacije, [4].



Slika 3.1. Lijevo: mehanizam pažnje parametriziran vektorom \vec{a} . Desno: primjena višestruke pažnje ($T = 3$) nad vrhom 1 i njegovim susjedstvom. Izvor: [3]

3.4. Izomorfne neuronske mreže za grafove

Iako su konvolucijske neuronske mreže za grafove i neuronske mreže za grafove s pažnjom postigle najbolje rezultate za određene skupove podataka, ne postoji teorijska podloga koja može objasniti njihovu ekspresivnost, ograničenja i robusnost učenja određenih struktura unutar grafova. Pokazalo se da konvolucijske mreže za grafove ne mogu naučiti razlikovati određene strukture unutar grafova, [5]. Zbog toga, razvijene su izomorfne neuronske mreže za grafove koje su u teorijskoj podlozi zasnovane na Weisfeiler-Lehman testu izomorfности grafova. Problem izomorfizma grafova postavlja pitanje jesu li dva grafa topološki identična. Radi se o izazovnom problemu jer još uvijek nije poznat algoritam koji ga rješava u polinomijalnom vremenu.

Multiskup je generalizirani koncept skupa koji dopušta višestruke instance elemenata. Formalnije, multiskup je uređeni par $\mathcal{X} = (\mathcal{S}, m)$ gdje je \mathcal{S} klasičan skup (sastavljen od različitih elemenata), a $m : \mathcal{S} \rightarrow \mathbb{N}$ funkcija koja svakom elementu pridružuje broj pojavljivanja unutar multiskupa. Agregiranje i integriranje informacija iz susjedstva možemo sagledati u kontekstu klasa funkcija nad multiskupovima.

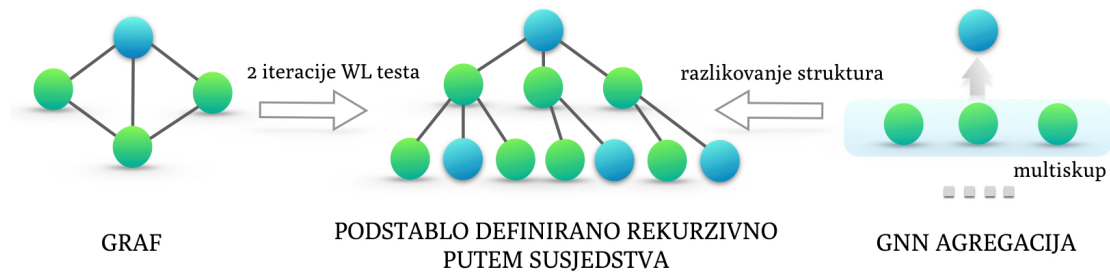
Reprezentativna moć neuronskih mreža za grafove može se analizirati provjeravajući kada neuronska mreža mapira dva vrha u istu reprezentaciju. Intuitivno, maksimalno robusna neuronska mreža mapira dva vrha u istu reprezentaciju samo ako imaju identične strukture podstabala susjedstava s identičnim značajkama na odgovarajućim vrhovima i bridovima (slika 3.2.). Budući da su strukture podstabala definirane rekurzivno putem susjedstava vrhova, možemo analizu svesti na pitanje hoće li neuronska mreža mapirati dva susjedstva u istu reprezentaciju. To znači da agregacijski mehanizam neuronske mreže mora biti injektivan (neuronska mreža nikada neće mapirati dva različita susjedstva u istu reprezentaciju). Konkretno, želimo takvu neuronsku mrežu GNN da vrijedi sljedeće: neka su \mathcal{G}_1 i \mathcal{G}_2 dva neizomorfna grafa; ako graf neuronska mreža $GNN : \mathcal{G} \rightarrow \mathbb{R}^{\delta^{(L)}}$ mapira \mathcal{G}_1 i \mathcal{G}_2 u dvije različite reprezentacije tada Weisfeiler-Lehman test izomorfizma će također pokazati da su grafovi \mathcal{G}_1 i \mathcal{G}_2 neizomorfni. Uz dovoljan broj slojeva neuronske mreže i negativan rezultat Weisfeiler-Lehman testa izomorfizma nad grafovima \mathcal{G}_1 i \mathcal{G}_2 neuronska mreža će mapirati \mathcal{G}_1 i \mathcal{G}_2 u različite reprezentacije ako vrijedi da:

1. Funkcije agregiranja f_M i integriranja f_U su injektivne.
2. Funkcija očitavanja f_R je injektivna.

Izomorfne neuronske mreže za grafove ažuriraju značajke vrha v u sloju l izrazom:

$$h_v^{(l)} = \text{MLP}^{(l)} \left((1 + \epsilon^{(l)}) \cdot h_v^{(l-1)} + \sum_{u \in \mathcal{N}(v)} h_u^{(l-1)} \right). \quad (3.14)$$

gdje je MLP višeslojni perceptron, a ϵ konstanta ili parametar kojeg neuronska mreža uči. Može se pokazati da izomorfne neuronske mreže za grafove zadovoljavaju dva navedena uvjeta, generaliziraju Weisfeiler-Lehman test izomorfности i postižu maksimalnu robusnost razlikovanja struktura grafova u odnosu na sve druge arhitekture neuronskih mreža za grafove, [5].



Slika 3.2. Pregled teorijskog okvira izomorfних neuronskih mreža za grafove. Lijevo: izvorni graf. Sredina: podstablo susjedstava (s obzirom na plavi vrh) koje Weisfeiler-Lehman test koristi za razlikovanje grafova. Desno: ako funkcija agregiranja GNN-a primi potpuni multiskup susjednih vrhova, GNN rekurzivno ima informaciju svih podstabala susjedstva i jednako je robusan kao i Weisfeiler-Lehman test izomorfности. Izvor: [5]

4. Metodologija

4.1. Samonadzirano učenje reprezentacije na razini grafa

Općenito, standardni sloj neuronske mreže za grafove svodi se na dva koraka: agregacija i integracija. Nakon što ulazni graf prođe kroz prvi sloj mreže, svaki vrh unutar svojih značajki integrira informaciju koja je udaljena za jedan skok od njega (susjedstvo udaljeno za jedan brid od vrha). U l -tom sloju neuronske mreže za grafove značajke svakog vrh sadrže informaciju o susjedstvu koje je za l skokova udaljeno od njega (u slučaju da je graf usmjeren svaki vrh prima značajke iz susjedstva koje su, s obzirom na brid, ulazne). Značajke vrha v l -tog sloja neuronskih mreže za grafove općenito možemo zapisati izrazom:

$$H_v^{(l)} = f_U^{(l)}(H_v^{(l-1)}, f_M^{(l)}(\{(H_u^{(l-1)}, H_u^{(l-1)}, X_{uv}) : u \in \mathcal{N}(v)\})) \quad (4.1)$$

$H_v^{(l)}$ nazivamo reprezentacijom podgraфа centriranog oko vrha v u sloju l konstruiranog od susjedstva udaljenog za l skokova. Želimo da reprezentacije sličnih podgrafova $\vec{h}_v^{(l)}$ budu blizu unutar naučenog prostora (tj. naučiti lokalne strukture unutar grafa) te da reprezentacije grafova \vec{h}_g izračunate nad značajkama vrhova na izlazu zadnjeg sloja $H_v^{(L)}$ odražavaju globalnu semantiku istih unutar skupa podataka. Globalnu semantiku učimo tražeći prototipove unutar naučenog prostora koji najbolje opisuju grafove iz skupa podataka. Dodatno, želimo globalnu semantiku strukturirati formom stabla kako bi dobili hijerarhijski odnos među pronađenim prototipima. Model GraphLoG, koristeći samonadzirano okruženje, nudi rješenje učenja lokalnih struktura unutar grafa i globalne semantike provodeći sljedeće korake:

1. Nad uzorkovanim grafom \mathcal{G} iz skupa podataka napravimo malu perturbaciju i dobijemo njegovi korelirani graf \mathcal{G}' . Svaki vrh u \mathcal{G} ima svoj odgovarajući vrh u \mathcal{G}' .

2. Učenje lokalnih struktura unutar grafa:

- Za reprezentacije vrhova (podgrafova) $\vec{h}_v^{(L)}$ grafa \mathcal{G} zahtijevamo da su sličnije u naučenom prostoru reprezentacijama odgovarajućih vrhova korelirajućeg grafa \mathcal{G}' nego reprezentacijama vrhova nasumično odabranog grafa iz skupa podataka.
- Za reprezentaciju $\vec{h}_g^{(L)}$ grafa \mathcal{G} zahtijevamo da je sličnija u naučenom prostoru reprezentaciji $\vec{h}_{g'}^{(L)}$ njegovog koreliranog grafa \mathcal{G}' nego reprezentaciji nekog drugog nasumično odabranog grafa iz skupa podataka.

3. Učenje globalne semantike grafa G unutar skupa podataka:

Nad reprezentacijom \vec{h}_g grafa \mathcal{G} provodimo online EM algoritam zahtijevajući da pridijeljeni prototipovi koji opisuju graf \mathcal{G} maksimiraju izglednost skupa podataka.

Učenje lokalnih struktura unutar grafa u prostoru reprezentacije grafova

Za graf $\mathcal{G} = (\mathcal{V}, \mathcal{E}, X_{\mathcal{V}}, X_{\mathcal{E}})$ uzorkovan iz distribucije podataka $P_{\mathcal{G}}$ generiramo njegov korelirani graf $\mathcal{G}' = (\mathcal{V}', \mathcal{E}', X_{\mathcal{V}'}, X_{\mathcal{E}'})$ slučajnom perturbacijom. Reprezentacije podgrafova i grafova na izlazu iz L -tog sloja označimo na sljedeći način:

$$(H_{\mathcal{V}}, \vec{h}_g) = \text{GNN}_{\theta}(\mathcal{G}), \quad (H_{\mathcal{V}'}, \vec{h}_{g'}) = \text{GNN}_{\theta}(\mathcal{G}'), \quad (4.2)$$

Prilikom učenja zahtijevamo da je sličnost reprezentacija podgrafova $H_{\mathcal{V}}^{(L)}$ i reprezentacija odgovarajućih podgrafova koreliranog grafa $H_{\mathcal{V}'}^{(L)}$ veća nego sličnost s reprezentacijama podgrafova iz nasumično odabranog grafa unutar skupa podataka. Također, zahtijevamo da je sličnost reprezentacije grafa \vec{h}_g i reprezentacije koreliranog grafa $\vec{h}_{g'}$ veća nego sličnost s nasumično odabranim grafom iz skupa podataka. Korelirani graf i pripadajuće podgrafove nazivamo pozitivnim primjerima u odnosu na uzorkovani graf, a slučajno odabrani graf iz skupa podataka i pripadajuće podgrafove negativnim primjerima. Za mjeru sličnosti koristimo kosinusnu sličnost: [6]

$$s(\vec{x}, \vec{y}) = \frac{\vec{x}^{\top} \vec{y}}{\|\vec{x}\| \|\vec{y}\|} \quad (4.3)$$

Funkcije gubitka za učenje reprezentacije lokalnih struktura unutar grafa u prostoru reprezentacije grafova računa se na sljedeći način:

$$\mathcal{L}_{\text{graph}} = -\mathbb{E}_{(\mathcal{G}_+, \mathcal{G}'_+) \sim p(\mathcal{G}, \mathcal{G}'), (\mathcal{G}_-, \mathcal{G}'_-) \sim p_n(\mathcal{G}, \mathcal{G}')} [s(\mathcal{G}_+, \mathcal{G}'_+) - s(\mathcal{G}_-, \mathcal{G}'_-)], \quad (4.4)$$

$$\mathcal{L}_{\text{sub}} = -\mathbb{E}_{(\mathcal{G}_u, \mathcal{G}'_u) \sim p(\mathcal{G}_v, \mathcal{G}'_v), (\mathcal{G}_v, \mathcal{G}'_v) \sim p_n(\mathcal{G}_v, \mathcal{G}'_v)} [s(\mathcal{G}_u, \mathcal{G}'_u) - s(\mathcal{G}_v, \mathcal{G}'_v)], \quad (4.5)$$

gdje $p_n(\mathcal{G}, \mathcal{G}')$ i $p_n(\mathcal{G}_v, \mathcal{G}'_v)$ predstavljaju distribucije šuma iz kojih su uzorkovani negativni primjeri za graf \mathcal{G} . Učenje lokalnih struktura unutar grafova svodi se na minimizaciju 4.4 i 4.5:

$$\min_{\theta} \mathcal{L}_{\text{local}}, \quad (4.6)$$

$$\mathcal{L}_{\text{local}} = \mathcal{L}_{\text{graph}} + \mathcal{L}_{\text{sub}}. \quad (4.7)$$

Učenje globalne semantike u prostoru reprezentacije grafova

Unutar raznih skupova podataka koji sadrže grafove možemo primijetiti hijerarhijsku strukturu među podacima (npr. skup podataka molekula). Za podatke unutar takvih skupova htjeli bismo naučiti reprezentaciju koja odražava njihov međusobni hijerarhijski odnos (globalnu semantiku određenog podatka unutar skupa podataka). U slučaju da nemamo oznake koje upućuju na globalnu semantiku ne možemo ovom problemu pristupiti iz pogleda nadziranog učenja. Kao rješenje za nedostatak oznaka uvodimo dodatni skup parametara kojeg zovemo hijerarhijski prototipi. Hijerarhijski prototipi su primjeri koji najbolje predstavljaju određenu grupu grafova u prostoru reprezentacije grafova formiranu s obzirom na globalnu semantiku podataka unutar skupa. Takvi primjeri se nužno ne moraju nalaziti unutar skupa podataka. Hijerarhijske prototipove formalno definiramo kao $C = \{c_i^{(l)}\}_{i=1}^{M^{(l)}} (l = 1, 2, \dots, L_p)$ gdje je $c_i^{(l)} \in \mathbb{R}^{\delta^{(l)}}$ i -ti prototip l -te dubine hijerarhijskog stabla, L_p dubina hijerarhijskog stabla i $M^{(l)}$ broj prototipova l -te dubine hijerarhijskog stabla. Unutar jednog takvog stabla vrh predstavlja jedan prototip i svi vrhovi osim listova posjeduju skup djece $C(c_i^l)$ ($1 \leq i \leq M^{(l)}, l = 1, 2, \dots, L_p - 1$), [6]. Cilj učenja globalne semantike je da dodatno *zbliži* reprezentacije grafova slične globalne semantike unutar naučenog prostora. Problem učenja hijerarhijskih prototipova formaliziramo definiranjem i optimiranjem modela sa skrivenim varijablama. Danom skupu podataka $G = \{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_M\}$ pridružujemo skup skrivenih varijabli $Z = \{z_{\mathcal{G}_1}, z_{\mathcal{G}_2}, \dots, z_{\mathcal{G}_M}\}$ gdje

je $z_{\mathcal{G}_m}$ skup prototipova koji najbolje opisuju G_m unutar naučenog prostora. Skup parametara modela sadrži skup parametara neuronske mreže θ i hijerarhijske prototipove C . Zbog nedostatka stvarnih oznaka prototipova teško je direktno maksimizirati funkciju izglednosti kompletnog skupa podataka $p(G, Z|\theta, C)$. Stoga, koristeći EM algoritam maksimiziramo očekivanje izglednosti kompletnog skupa podataka uzimajući u obzir skrivene varijable. Dodatno, koristimo online verziju EM algoritma koja uz pretpostavku identično i nezavisno distribuiranog skupa podataka provodi ažuriranje parametara nad mini-grupama. Izglednost kompletnog skupa podataka sa skrivenim varijablama i aposteriorna vjerojatnost skrivenih varijabli može biti faktorizirana s obzirom na svaki par grafa i njegove pripadajuće skrivene varijable:

$$p(G, Z|\theta, C) = \prod_{m=1}^M p(\mathcal{G}_m, z_{\mathcal{G}_m}|\theta, C), \quad (4.8)$$

$$p(Z|G, \theta, C) = \prod_{m=1}^M p(z_{\mathcal{G}_m}|\mathcal{G}_m, \theta, C). \quad (4.9)$$

Koristeći online algoritam izbjegnut je standardni postupak EM algoritma koji zahtjeva račun nad potpunim skupom podataka kako bi ažurirao parametre (što u slučaju jako velikih skupova podataka nije izračunski efikasno).

Općeniti EM algoritam

Zadatak EM algoritma je pronaći rješenje koje maksimizira izglednost podataka i pridruženih skrivenih varijabli. U E-koraku koristeći podatke G i parametre modela θ_{t-1} procijenjene u prethodnom EM ciklusu procjenjujemo aposteriornu distribuciju latentnih varijabli $p(Z|G, \theta_{t-1})$, t.j. odgovornost latentnih varijabli Z za pojavljivanje očitanih podataka unutar skupa G . U M-koraku, računamo log-očekivanje kompletnog skupa podataka $Q(\theta)$ uzimajući u obzir procijenjenu aposteriornu distribuciju iz E-koraka:

$$Q(\theta) = \mathbb{E}_{p(Z|G, \theta_{t-1})}[\log p(G, Z|\theta)]. \quad (4.10)$$

Ažuriramo parametre modela tako da maksimiziramo očekivanje u M-koraku:

$$\theta_t = \arg \max_{\theta} Q(\theta). \quad (4.11)$$

Svaki EM ciklus povećava izglednost kompletnog skupa podataka. Pokazano je da EM algoritam ima mogućnost maksimiziranja marginalne funkcije izglednosti $p(G|\theta)$, [6].

E-korak

Nasumično uzorkujemo mini-grupe grafova $\tilde{G} = \{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_B\}$ iz skupa podataka G (B je veličina mini-grupe). Neka su $\tilde{Z} = \{z_{\mathcal{G}_b}\}_{b=1}^B$ odgovarajuće skrivene varijable pridružene grafovima iz mini-grupe. Svaka skrivena varijabla $z_{\mathcal{G}_b} = \{\vec{z}_{\mathcal{G}_b}^{(1)}, \vec{z}_{\mathcal{G}_b}^{(2)}, \dots, \vec{z}_{\mathcal{G}_b}^{(L_p)}\}$ je lanac pripadajućih prototipova koji najbolje opisuje graf \mathcal{G}_b . U hijerarhijskom odnosu prototipova $\vec{z}_{\mathcal{G}_b}^{(l+1)}$ je dijete vrhu $\vec{z}_{\mathcal{G}_b}^{(l)}$. Aposteriorna distribucija varijable \tilde{Z} uz pretpostavku o jednakoj i nezavisnoj distribuiranosti može se faktorizirati na sljedeći način:

$$p(\tilde{Z}|\tilde{G}, \theta_{t-1}, C_{t-1}) = \prod_{b=1}^B p(z_{\mathcal{G}_b}|\mathcal{G}_b, \theta_{t-1}, C_{t-1}), \quad (4.12)$$

gdje su θ_{t-1} i C_{t-1} parametri modela iz prošlog EM ciklusa. Budući da direktna evaluacija svake od posteriornih distribucija $p(\tilde{Z}|\tilde{G}, \theta_{t-1}, C_{t-1})$ zahtjeva prolaz kroz sve moguće lance hijerarhijskih prototipova, uvodimo pristup stohastičkog EM algoritma uzorkujući $\hat{z}_{\mathcal{G}_b} \sim p(z_{\mathcal{G}_b}|\mathcal{G}_b, \theta_{t-1}, C_{t-1})$ uz Monte Carlo procjenu. Prvo uzorkujemo prototip s obzirom na prvi sloj $\vec{z}_{\mathcal{G}_b}^1 \sim \text{Cat}(\vec{z}_{\mathcal{G}_b}^1|\{\alpha_i\}_{i=1}^{M^{(1)}})$ ($\alpha_i = \text{Softmax}(s(\vec{c}_i^1, \vec{h}_{\mathcal{G}_b}))$), gdje je s kosinusna sličnost. Za sve ostale slojeve l ($l \geq 2$) uzorkujemo prototip s obzirom na kategoričku distribuciju djece prototipa $\vec{z}_{\mathcal{G}_b}^{(l-1)}$: $\vec{z}_{\mathcal{G}_b}^l \sim \text{Cat}(\vec{z}_{\mathcal{G}_b}^l|\{\alpha_c\})$ ($\alpha_c = \text{Softmax}(s(\vec{c}, \vec{h}_{\mathcal{G}_b}))$, $\forall c \in \mathbb{C}(\vec{z}_{\mathcal{G}_b}^{(l-1)})$). Provedeći uzorkovanje (lanca) hijerarhijskih prototipova, uzorkovali smo skrivenu varijablu $\hat{z}_{\mathcal{G}_b} = \{\vec{z}_{\mathcal{G}_b}^1, \vec{z}_{\mathcal{G}_b}^2, \dots, \vec{z}_{\mathcal{G}_b}^L\}$, [6].

M-korak

Koristeći $\hat{z}_{\mathcal{G}_b}$ iz E-koraka maksimiziramo očekivanje log-izglednosti kompletnog skupa podataka:

$$Q(\theta, C) = \mathbb{E}_{p(Z|G, \theta_{t-1}, C_{t-1})}[\log p(G, Z|\theta, C)]. \quad (4.13)$$

Budući da ovako definirano očekivanje zahtjeva prolazak kroz sve podatke unutar skupa podataka, htjeli bi dodatno prilagoditi izraz kako bi ga mogli optimirati na online način (optimizacija koristeći mini-grupe). Očekivanje log-izglednosti mini-grupe \tilde{G} procijenjeno

s obzirom na $\tilde{Z}_{est} = \{\hat{z}_{g_b}\}_{b=1}^B$ iz E-koraka možemo zapisati na sljedeći način:

$$\begin{aligned}\tilde{Q}(\theta, C) &= \mathbb{E}_{p(\tilde{Z}|\tilde{G}, \theta_{t-1}, C_{t-1})}[\log p(\tilde{G}, \tilde{Z}|\theta, C)] \\ &\approx \log p(\tilde{G}, \tilde{Z}_{est}|\theta, C) \\ &= \sum_{b=1}^B \log p(\mathcal{G}_b, \hat{z}_{g_b}|\theta, C).\end{aligned}\tag{4.14}$$

Može se pokazati da je $\tilde{Q}(\theta, C)$ primjeren zamjena za $Q(\theta, C)$:

$$\tilde{Q}(\theta, C) \approx \frac{N}{M} Q(\theta, C).\tag{4.15}$$

Dodatno skaliramo $\tilde{Q}(\theta, C)$ s veličinom mini-grupe kako bi dobili funkciju log-izglednosti $\mathcal{L}(\theta, C)$ koja je računski stabilnija:

$$\mathcal{L}(\theta, C) = \frac{1}{N} \tilde{Q}(\theta, C).\tag{4.16}$$

Kako bi procijenili $\mathcal{L}(\theta, C)$ definiramo združenu izglednost grafa \mathcal{G} i skrivene varijable $z_{\mathcal{G}}$ predstavljenu energetsom funkcijom:

$$p(\mathcal{G}, z_{\mathcal{G}}|\theta, C) = \frac{1}{Z(\theta, C)} \exp(f(\vec{h}_{\mathcal{G}}, z_{\mathcal{G}})),\tag{4.17}$$

gdje je $Z(\theta, C)$ particijska funkcija, a f negativna energetska funkcija. Negativnu energetska funkciju f definiramo kao mjeru sličnosti reprezentacije grafa $\vec{h}_{\mathcal{G}}$ i prototipova u $\vec{z}_{\mathcal{G}}^{(l)}$ i sličnosti prototipova $\vec{z}_{\mathcal{G}}^{(l)}$ i $\vec{z}_{\mathcal{G}}^{(l+1)}$ koji se nalaze u uzastopnim hijerarhijskim slojevima prototipova:

$$f(\vec{h}_{\mathcal{G}}, z_{\mathcal{G}}) = \sum_{l=1}^{L_p} s(\vec{h}_{\mathcal{G}}, \vec{z}_{\mathcal{G}}^{(l)}) + \sum_{l=1}^{L_p-1} s(\vec{z}_{\mathcal{G}}^{(l)}, \vec{z}_{\mathcal{G}}^{(l+1)}).\tag{4.18}$$

Netraktabilnost particijske funkcije čini optimizaciju $p(\mathcal{G}, z_{\mathcal{G}}|\theta, C)$ netrivialnom. Zbog toga koristimo neku od metoda za rad s nenormaliziranim statističkim modelima (unutar modela GraphLoG koristi se *Noise-Contrastive Estimation* metoda): optimiziramo nenormalizirane izglednosti $\tilde{p}(\mathcal{G}, z_{\mathcal{G}}|\theta, C) = \exp(f(\vec{h}_{\mathcal{G}}, z_{\mathcal{G}}))$ uspoređujući pozitivne parove promatranih i skrivenih varijabli s negativnim parovima uzorkovanih iz neke distribucije

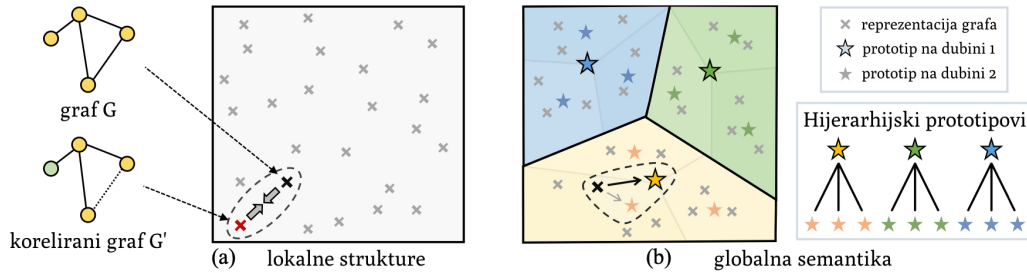
šuma. Funkciju gubitka koja dobro aproksimira $\mathcal{L}(\theta, C)$ možemo zapisati kao:

$$\mathcal{L}_{\text{global}} = -\mathbb{E}_{(\mathcal{G}^+, z_{\mathcal{G}}^+) \sim p(\mathcal{G}, z_{\mathcal{G}})} \left\{ \log \tilde{p}(\mathcal{G}^+, z_{\mathcal{G}}^+ | \theta, C) \right. \\ \left. - \mathbb{E}_{(\mathcal{G}^-, z_{\mathcal{G}}^-) \sim p_n(\mathcal{G}, z_{\mathcal{G}})} \left[\log \tilde{p}(\mathcal{G}^-, z_{\mathcal{G}}^- | \theta, C) \right] \right\}, \quad (4.19)$$

gdje je $p_n(\mathcal{G}, z_{\mathcal{G}})$ distribucija šuma. Vanjsko očekivanje računamo nad svim pozitivnim parovima u mini-grupi $(\mathcal{G}_n, \hat{z}_{\mathcal{G}_n})$ ($1 \leq n \leq N$), a za unutarnje očekivanje konstruiramo L_p negativnih parova za pozitivne parove $(\mathcal{G}_n, \hat{z}_{\mathcal{G}_n})$ fiksirajući graf \mathcal{G}_n i slučajno supstituirajući jedan od L_p prototipova u $\hat{z}_{\mathcal{G}_n}$ s nekim drugim prototipom u istom sloju svaki put. Učenje globalne semantike u prostoru reprezentacije grafova svodi se na minimizaciju funkcije $\mathcal{L}_{\text{global}}$ s obzirom na parametre neuronske mreže θ i hijerarhijske prototipove C :

$$\min_{\theta, C} \mathcal{L}_{\text{global}}. \quad (4.20)$$

Predloženi online EM algoritam nastoji maksimizirati zajedničku vjerojatnost $p(G, Z | \theta, C)$ s obzirom na parametre θ i C . Može se pokazati da takav online EM algoritam može uistinu maksimizirati marginalnu funkciju izglednosti $p(G | \theta, C)$, [6].



Slika 4.1. Ilustracija GraphLoG modela. (a) Zahtjeva se da reprezentacije koreliranih grafova (i vrhova) budu slične unutar naučenog prostora. (b) Hijerarhijski prototipovi koriste se za otkrivanje globalne semantičke strukture unutar skupa podataka. Izvor: [6]

4.2. Optimizacija i evaluacija modela

Optimizacija

Prvi korak treniranja je predtreniranje neuronske mreže za grafove kako bi minimizirali funkciju cilja $\mathcal{L}_{\text{local}}$ koja uči lokalne strukture unutar grafova. Tako predtreniranu neuronsku mrežu koristimo za inicijalizaciju prototipova $\{c_i^{(L_p)}\}_{i=1}^{M^{(L_p)}}$ računajući centroide reprezentacija cijelog skupa podataka uz pomoć algoritma k-sredina (engl. k-means). U svakom koraku učenja uzorkujemo mini-grupu \tilde{G} i provodimo EM ciklus. U E-koraku

pripadajuće skrivene varijable \tilde{Z}_{est} uzorkovane su iz aposteriorne distribucije definirane modelom. U M-koraku parametri modela se ažuriraju kako bi maksimizirali očekivanu log-izglednost mini-grupe. Dodatno, unutar M-koraka optimiziramo i \mathcal{L}_{local} budući da se pokazuje da u praksi to dodatno osigurava *glatkoću* među rješenjima prilikom traženja struktura koje predstavljaju globalnu semantiku.

Algoritam 3 Optimizacijski algoritam GraphLoG modela [6]

Ulaz: Neoznačeni skup podataka G , broj koraka učenja T .

Izlaz: Predtrenirani model neuronske mreže za grafove GNN_{θ_T} .

Predtreniraj neuronsku mrežu za grafove s funkcijom cilja 4.7

Inicijaliziraj parametre modela θ_0 (predtrenirani parametri neuronske mreže za grafove) i C_0 (C_0 inicijaliziraj koristeći algoritam k-sredina).

za $t = 1$ **do** T

Uzorkuj mini-grupu \tilde{G} iz G .

◇ *E-korak:*

Uzorkuj skrivenu varijablu \tilde{Z}_{est} s $GNN_{\theta_{t-1}}$ i C_{t-1} .

◇ *M-korak:*

Ažuriraj parametre modela:

$$\theta_t \leftarrow \theta_{t-1} - \nabla_{\theta}(\mathcal{L}_{local} + \mathcal{L}_{global}),$$

$$C_t \leftarrow C_{t-1} - \nabla_C(\mathcal{L}_{local} + \mathcal{L}_{global}).$$

kraj za

Evaluacija

S obzirom na to da ne raspolažemo stvarnim oznakama lokalnih struktura unutar grafova i globalnim semantičkim obilježjima grafova unutar skupa podataka, ne možemo provesti neku od standardnih metoda evaluacije. Informacije na koje se možemo osloniti prilikom evaluacije naučenih reprezentacija grafova su vrijeme dolaska skupova transakcija koje su formirale grafove, količina istrgovanih dolara unutar skupova transakcija te postotna promjena cijene. Funkcija cilja je takva da reprezentacije sličnih grafova pozicionira blizu unutar naučenog prostora. Provođenje neke od metoda grupiranja te analiziranje statističkih obilježja grupa (vremena, istrgovane količine u dolarima i postotne promjene cijene) trebalo bi nam ukazati na pravilnosti unutar određenih dijelova prostora. U slučaju da postoje određene pravilnosti unutar tako dobivenih grupa, možemo reći da naučene reprezentacije dobro opisuju semantiku ulaznih podataka te da je hipoteza o predstavljanju skupa transakcija kao grafa koji predstavlja dinamiku dvostruke aukcije validna. Namjera učenja reprezentacija je da se one koriste kao ulaz u neki drugi model

koji bi imao svrhu predviđanja idućeg stanja tržišta. Takve reprezentacije ne bi dodatno diskretizirali (ne zamjenjujemo reprezentaciju određenog grafa s predstavnikom grupe unutar koje taj graf spada). Stvaran broj grupa ne znamo, međutim, uzimajući u obzir da ni ne želimo koristiti predstavnike grupe umjesto samih reprezentacija, grupe služe samo kao *grubi* test koji provjerava je li naučeni prostor uistinu takav da u njemu slični grafovi posjeduju slična statistička svojstva.

5. Rezultati

Podaci koji su korišteni za izračun rezultata su transakcije na financijskom instrumentu ETHUSDT (budućnosnice) izvršene na burzi kriptovaluta Binance od trenutka kada je financijski instrument objavljen do rujna 2022. godine. Kako bi izračunali statističke vrijednosti potrebne za odabir hiperparametara prilikom generiranja grafova korišten je cijeli skup podataka. Za treniranje neuronske mreže za grafove korišteno je 6 mjeseci podataka (od 1.9.2021. do 1.3.2022.), dok je za izračun rezultata korišten testni skup podataka koji sadrži narednih 6 mjeseci podataka (od 1.3.2022. do 1.9.2022.).

Za početak, potrebno odabrati hiperparametre za generiranje grafova: veličina uzorka transakcija nad kojom se formira graf i granice razreda histograma. S obzirom na to da veličina uzorka transakcija određuje vrijeme donošenja odluke prilikom trgovanja, želimo takvu veličinu uzorka koja nam omogućava donošenje pravovremene odluke unatoč latenciji internetske veze u interakciji s burzom. Uz samu izvedivost donošenja pravovremene odluke prilikom trgovanja, bitno je i da devijacija vremena među odlukama bude što manja: tražimo veličinu uzorka sa što stabilnijim vremenom dolaska takvu da očekivano vrijeme dolaska omogućava donošenje pravovremene odluke. U tablici 5.1. prikazana su statistička svojstva vremena dolaska uzoraka transakcija s obzirom na veličinu uzorka. Vrijednosti iz tablice 5.1. sugeriraju da veličina uzorka od 250 transakcija najviše odgovara navedenim zahtjevima te je ona odabrani hiperparametar veličine uzorka transakcija: najmanje vrijeme dolaska je prihvatljivo s obzirom na latenciju veze s burzom, dok je standardna devijacija skoro dvostruko manja od iduće veličine uzorka. Kako bi izračunali vjerojatnosni histogram uz pomoć kojeg generiramo grafove potrebno je odrediti granice razreda histograma koje definiraju različita tržišna stanja. Za svaku os računamo percentilne vrijednosti te ih koristimo kao granice koje definiraju razrede histograma. U tablici 5.2 prikazane su odabrane percentilne vrijednosti za količinu istrgovanu u dolarima, postotnu promjenu cijene i razliku vremenskih oznaka.

veličina uzorka	μ	σ	min	25%	50%	75%	max
125	56.74	96.97	0.00	10.97	25.71	59.33	10776.03
250	173.08	240.52	0.01	36.87	85.39	205.07	11173.39
375	334.31	404.39	1.90	71.07	185.37	450.90	11617.88
500	549.18	569.22	5.13	145.51	375.68	750.12	12019.57
625	796.17	735.11	7.55	262.47	605.23	1089.85	12473.14
750	1039.55	911.99	16.20	368.86	817.72	1417.96	12946.16
875	1219.38	1095.41	22.61	408.68	946.28	1675.10	13395.68
1000	1438.63	1271.24	28.44	490.53	1129.25	1986.96	13755.70
1125	1720.82	1466.42	33.59	590.25	1395.90	2395.00	14066.38
1250	2048.32	1635.65	32.81	803.38	1702.29	2781.88	14399.08
1375	2326.22	1838.57	41.15	918.57	1935.24	3252.23	14861.01
1500	2634.65	2037.06	55.13	1018.94	2242.12	3686.97	15296.03
1625	2960.72	2223.90	56.42	1161.69	2674.99	4101.15	15612.14
1750	3283.23	2410.90	72.06	1356.32	2915.02	4603.71	15971.82
1875	3632.11	2600.69	79.02	1502.05	3240.25	5110.70	16386.76

Tablica 5.1. Statističke vrijednosti vremena s obzirom na veličinu uzorka transakcija: srednja vrijednost, standardna devijacija, najmanja vrijednost, percentilne vrijednosti i najveća vrijednost. Vrijednosti su zaokružene na dvije decimale; vrijeme je izraženo u sekundama.

	0.01%	1%	3%	5%	10%	20%	30%
Dolar	-7.6×10^5	-8.7×10^4	-2.8×10^4	-8.3×10^3	-2.1×10^3	-1.1×10^3	-6.3×10^2
Cijena	-1.3×10^{-3}	-5.3×10^{-4}	-2.2×10^{-4}	-7.2×10^{-5}	-2.1×10^{-5}	-1.1×10^{-5}	-7.9×10^{-6}
Vrijeme	1.0	1.0	1.0	1.1	1.9	5.5	1.1×10^1

	40%	50%	60%	70%	80%	90%	95%
Dolar	-9.2×10^1	-2.3×10^1	1.2×10^2	6.2×10^2	1.6×10^3	3.1×10^3	9.4×10^3
Cijena	-3.1×10^{-6}	-1.1×10^{-6}	2.1×10^{-6}	7.2×10^{-6}	1.5×10^{-5}	3.1×10^{-5}	7.6×10^{-5}
Vrijeme	1.8×10^1	3.0×10^1	5.0×10^1	8.3×10^1	1.4×10^2	2.7×10^2	4.4×10^2

	97%	99%	99.9%
Dolar	2.2×10^4	1.4×10^5	8.8×10^5
Cijena	1.1×10^{-4}	5.6×10^{-4}	1.4×10^{-3}
Vrijeme	5.8×10^2	9.2×10^2	2.9×10^3

Tablica 5.2 Percentilne vrijednosti za istrgovanu količinu izraženu u dolarima, postotnu promjenu cijene i razliku vremenskih oznaka (izraženo u milisekundama).

Percentilne vrijednosti dodatno korigiramo kako bi postigli simetričnost pozitivnih i negativnih razreda na svakoj osi. Razlike vremenskih oznaka gusto su distribuirane uz 0 te zbog toga odbacujemo granice određenih razreda na toj osi. Sve vrijednosti u skupu podataka ograničavamo na minimalne, odnosno maksimalne vrijednosti granica razreda. Granice razreda za svaku os prikazane su u tablici 5.3

Dolar	Cijena	Vrijeme
-1×10^6	-1×10^{-3}	0
-1×10^5	-6×10^{-4}	2×10^2
-3.5×10^4	-2.5×10^{-4}	6×10^2
-1×10^4	-8.5×10^{-5}	1×10^3
-2.5×10^3	-2.2×10^{-5}	
-1×10^3	-1×10^{-5}	
-5×10^2	-7.5×10^{-6}	
-5×10^1	-1×10^{-6}	
0	0	
5×10^1	1×10^{-6}	
5×10^2	7.5×10^{-6}	
1×10^3	1×10^{-5}	
2.5×10^3	2.2×10^{-5}	
1×10^4	8.5×10^{-5}	
3.5×10^4	2.5×10^{-4}	
1×10^5	6×10^{-4}	
1×10^6	1×10^{-3}	

Tablica 5.3 Odabrane granice razreda histograma.

Nakon definiranja veličine uzorka transakcija i granica razreda histograma, za svaki uzorak transakcija možemo generirati graf kako je opisano algoritmom 2. Generirani grafovi koriste se kao ulaz u neuronsku mrežu za grafove koja je sastavljena od 5 izomorfnih slojeva. Hiperparametri neuronske mreže za grafove i optimizacije:

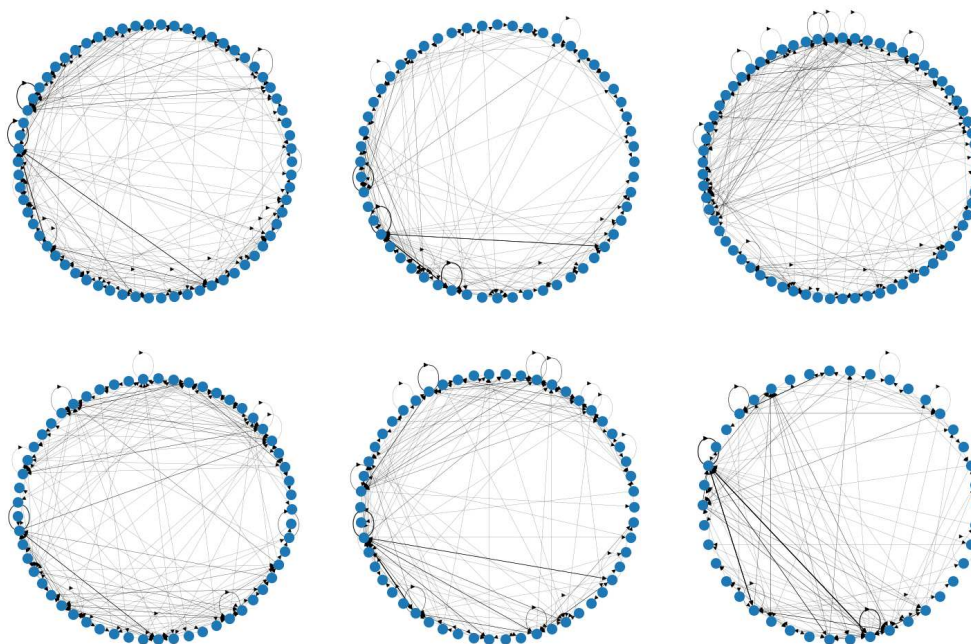
- **Velčina mini-grupe:** 256
- **Broj epoha predtreniranja (samo s $\mathcal{L}_{\text{local}}$):** 1
- **Broj epoha treniranja ($\mathcal{L}_{\text{local}}$ i $\mathcal{L}_{\text{global}}$):** 3
- **Stopa učenja:** 1×10^{-3}
- **Dimenzionalnost naučenih reprezentacija (i skrivenih slojeva):** 128
- **Vjerojatnost zašumljivanja brida unutar mini-grupe:** 8×10^{-1}
- **Srednja vrijednost šuma:** 0
- **Standardna devijacija šuma:** 7×10^{-1}
- **Funkcija očitavanja:** srednja vrijednost

- τ (temperaturni parametar *softmax* funkcije): 4×10^{-2}
- **Brzina umanjivanja** (eksponencijalno umanjujući prosjek): 9.5×10^{-1}
- **Dubina hijerarhijskog stabla**: 3
- **Broj inicijaliziranih prototipova po hijerarhijskom sloju**: 50
- α : 1; β : 1; γ : 0.1

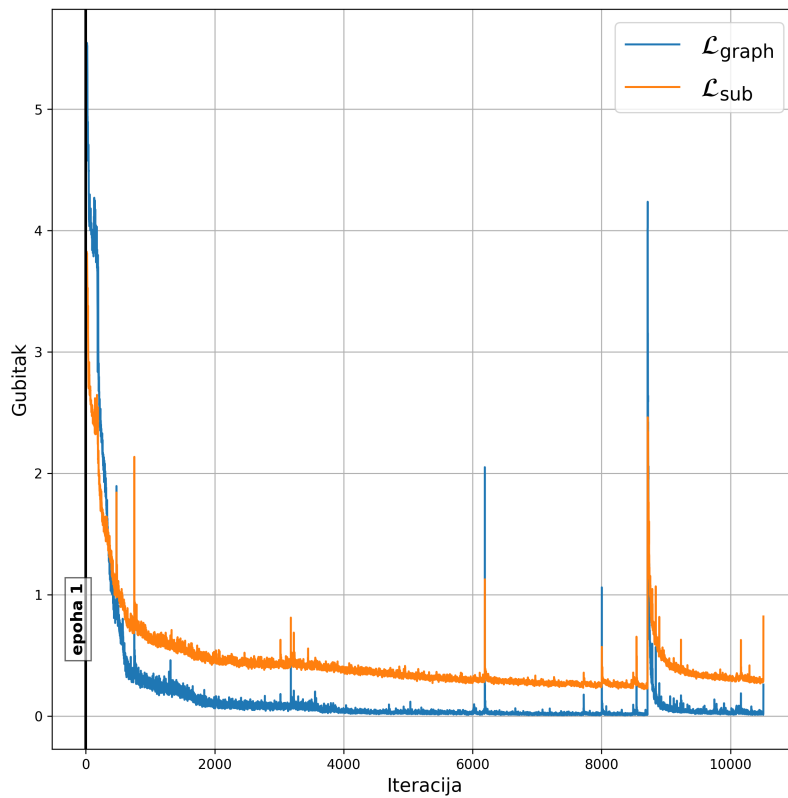
Ukupna funkcija gubitka dodatno skalira svaki pojedini gubitak:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{graph}} + \beta \mathcal{L}_{\text{sub}} + \gamma \mathcal{L}_{\text{global}} \quad (5.1)$$

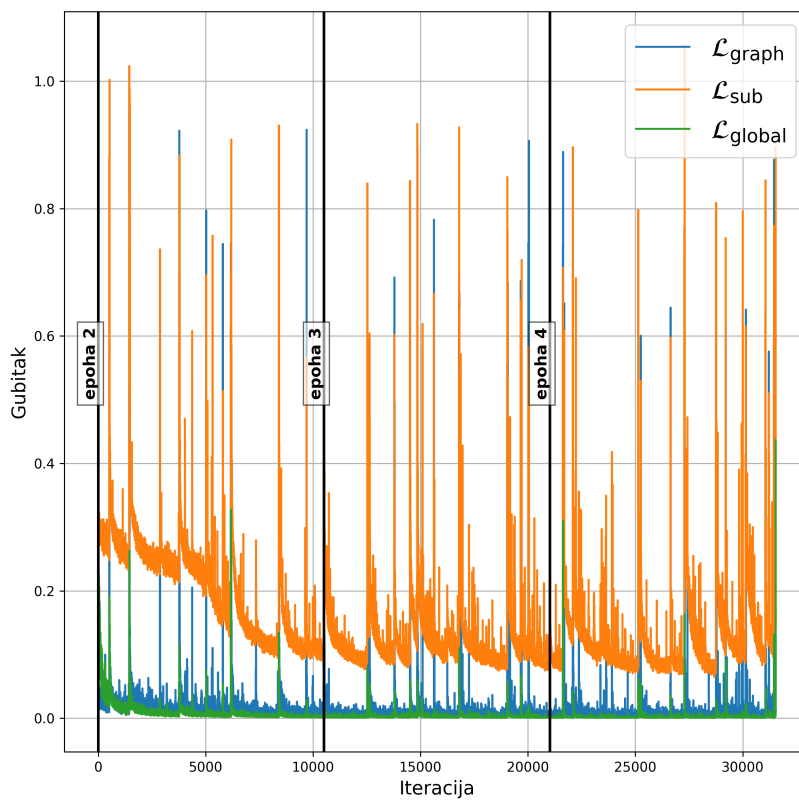
Perturbacija grafova provodi se na razini mini-grupe. Vjerojatnost zašumljivanja brida unutar mini-grupe određuje hoće li brid unutar mini-grupe biti zašumljen. Šum uzorkujemo iz normalne distribucije s određenom srednjom vrijednošću i standardnom devijacijom te ga nadodajemo na težinu brida. Šum se dodaje na nenormalizirane težine bridova te se potom težine bridova normaliziraju dijeljenjem s ukupnom sumom težina bridova. Unutar neuronske mreže za grafove, prilikom agregacije, značajke susjednih vrhova skalirane su težinom brida.



Slika 5.1. Primjer generiranih grafova. Grafovi su generirani nad 6 uzastopnih uzoraka transakcija veličine 256 transakcija na financijskom instrumentu ETHUSDT u 00:00 1.3.2022.

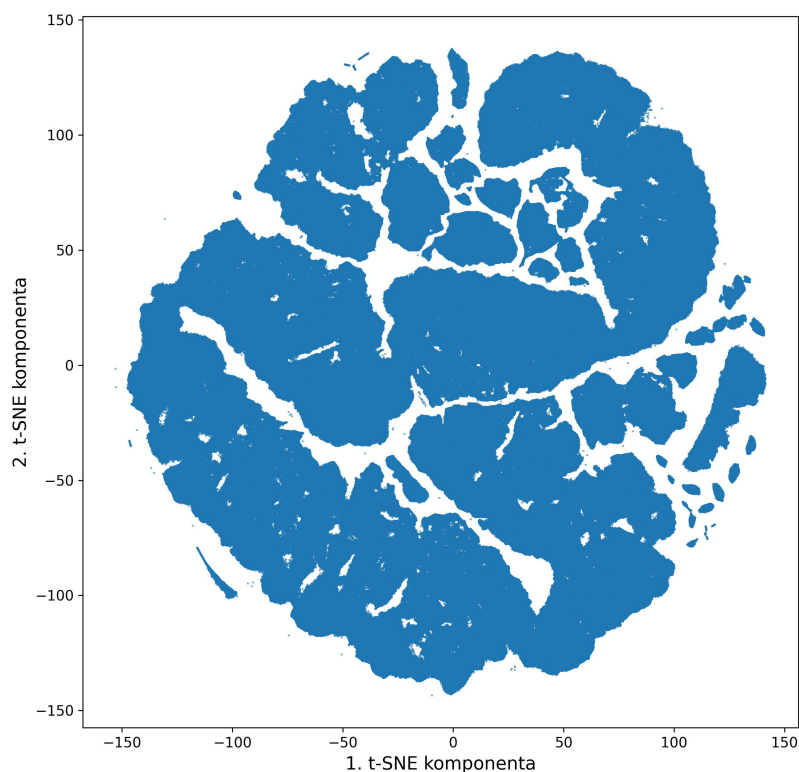


Slika 5.2. Gubitak tijekom predtreniranja.



Slika 5.3. Gubitak tijekom treniranja.

Kako bi vizualizirali rezultate i proveli opisanu evaluaciju reduciramo dimenzionalnost naučenih reprezentacija grafova preslikavajući značajke iz 128 dimenzija u 2 dimenzije algoritmom t-SNE (engl. t-distributed stochastic neighbor embedding). t-SNE reducira dimenzionalnost transformirajući sličnosti između točaka u zajedničke vjerojatnosti te minimizirajući Kullback-Leiblerovu divergenciju između visokodimenzionalnog i niskodimenzionalnog prostora. Provedbom t-SNE algoritma sigurni smo da je informacija o sličnosti točaka očuvana. Primjenom nekog od algoritama grupiranja nad niskodimenzionalnim reprezentacijama grubo zahvaćamo susjedstva unutar visokodimenzionalnog prostora. Budući da nemamo niti jednu stvarnu oznaku s kojom bi mogli provesti evaluaciju, ne želimo donositi ikakvu tvrdnju o grupama. Zanima nas samo postoje li statističke pravilnosti unutar grupa (grubo zahvaćenih dijelova susjedstva). Unutar niskodimenzionalnog prostora znamo da se kosinusna sličnost iz visokodimenzionalnog prostora odražava kroz blizinu reprezentacija. Provodimo grupiranje niskodimenzionalnih reprezentacija modelom Gaussove mješavine (engl. Gaussian mixture model) kako bi grubo zahvatili susjedstva unutar visokodimenzionalnog prostora.



Slika 5.4. Prikaz značajki nakon redukcije dimenzionalnosti algoritmom t-SNE. Korišteni podaci su testni skup transakcija na financijskom instrumentu ETHUSDT u razdoblju od 1.3.2022. do 1.9.2022.

Tri domene dinamike dvostruke aukcije su: istrgovana količina izražena u dolarima, postotna promjena cijene i razlika između vremenskih oznaka. Za svaku grupu dobivenu modelom Gaussove mješavine želimo vidjeti postoje li pravilnosti unutar te tri domene. To provodimo na sljedeći način:

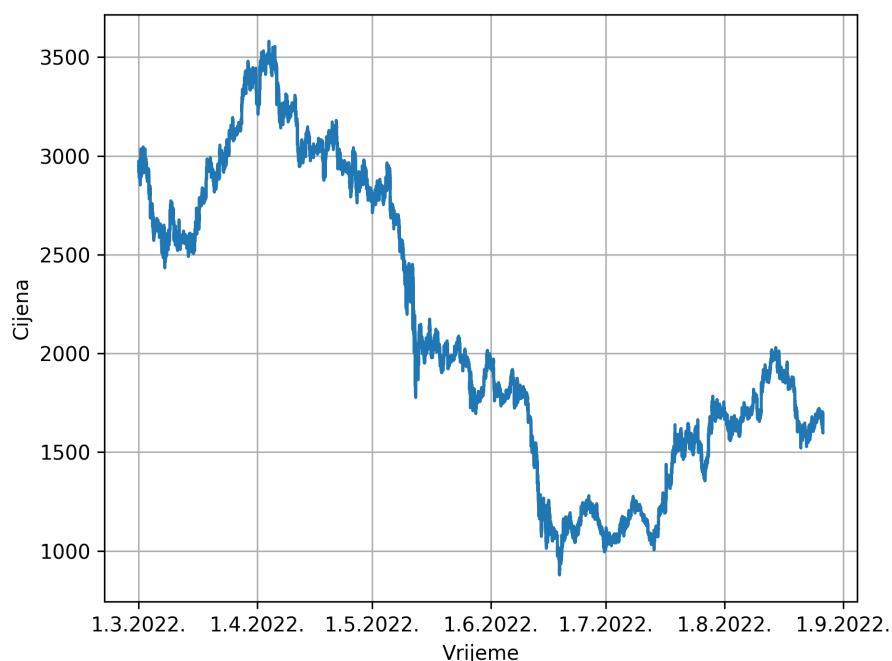
1. S obzirom na testni skupa podataka transakcija generiramo grafove (pamtimo par: generirani graf i uzorak transakcija).
2. Generirane grafove preslikamo u vektorsku reprezentaciju uz pomoć istrenirane neuronske mreže za grafove.
3. Reduciramo dimenzionalnost vektorskih reprezentacija uz pomoć algoritma t-SNE.
4. Unutar niskodimenzionalnog prostora provodimo algoritam grupiranja modelom Gaussove mješavine: svaki graf pripada određenoj grupi.
5. Za svaki graf unutar grupe, nad uzorkom transakcija koje su ga formirale računamo sumu istrgovane količine izražene u dolarima, postotne promjene cijene i razlike između vremenskih oznaka. Dobivene vrijednosti su promjene unutar te tri domene koje je uzrokovao taj skup transakcija. Nad svakom od te tri domene, unutar svake grupe izračunaj kumulativnu sumu dobivenih vrijednosti promjena.

Kumulativne sume vrijednosti promjena nad sve tri domene unutar grupa prikazane su na slikama 5.6., 5.7., 5.8., 5.9. i 5.10. Cijena financijskog instrumenta ETHUSDT na testnom skupu transakcija prikazana je na slici 5.5. Promatranjem cijene i izračunatih kumulativnih suma možemo uočiti da naučene reprezentacije odražavaju određene pravilnosti u domeni vremena. Budući da grupe grubo zahvaćaju susjedstva unutar visokodimenzionalnog prostora, određena odstupanja od pravilnosti možemo pripisati stršećim primjerima (engl. outlierima) unutar grupe. Zanimljivo je primijetiti da unutar određenih vremenskih intervala postoje linearni odnosi određenih grupa naspram sve tri domene dinamike dvostruke aukcije što upućuje na konzistentnost promjena koje nanosi ta grupa. Također, zanimljivo je primijetiti trenutke u kojima grupa počne odstupati od linearnih odnosa: na slici 5.10. vidimo da su se određene grupe počele realizirati brže u vremenu prije naglih promjena cijene, dok su se druge grupe počele realizirati sporije. Na slici 5.6. (koju promatramo zbog jednostavnosti) možemo primijetiti nekoliko

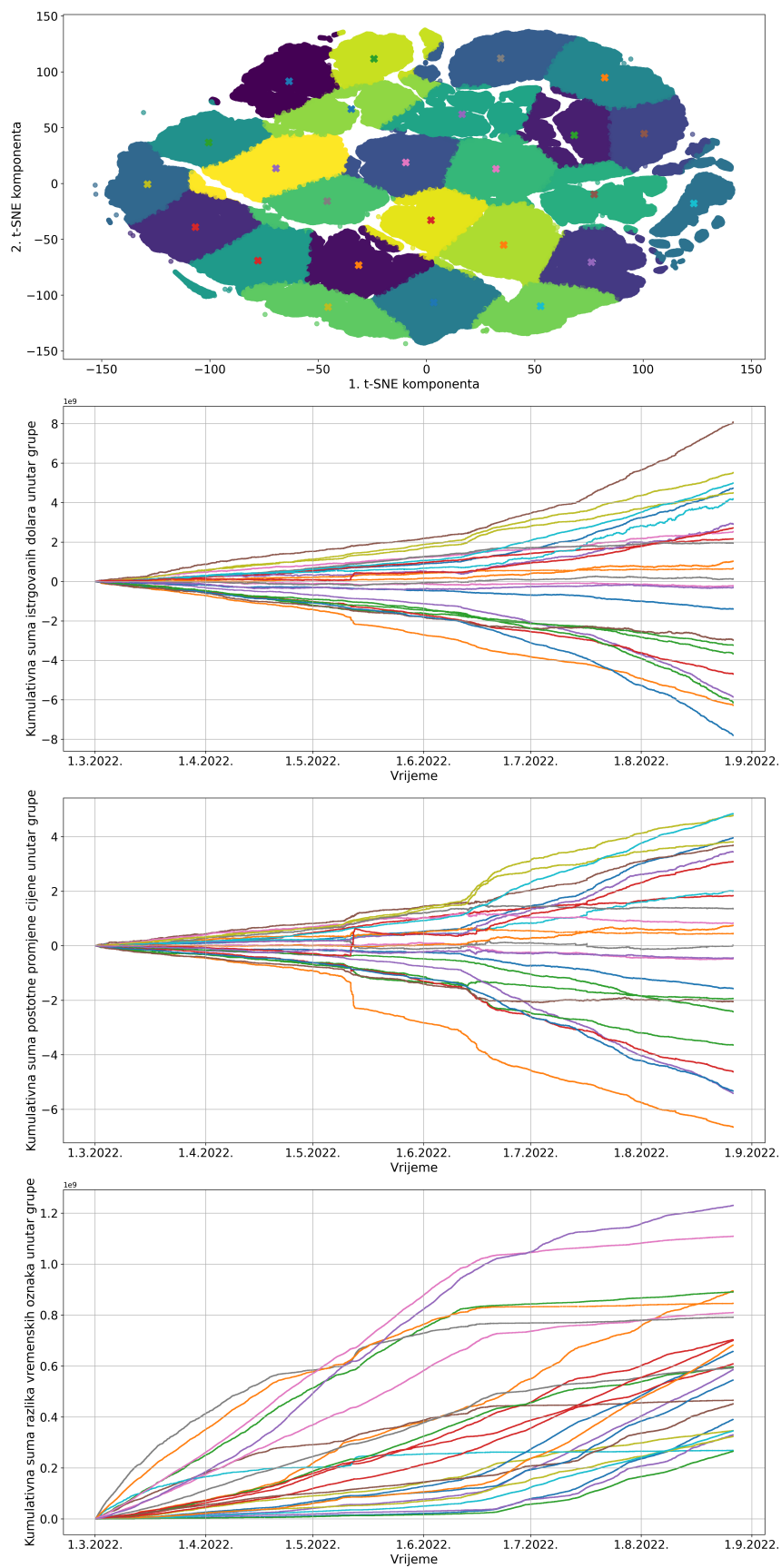
zanimljivosti:

- Grupe unutar kojih se kumulativno skuplja najviše kupovnih, odnosno prodajnih dolara brže se realiziraju u vremenu od grupa unutar kojih se kumulativno skuplja malo istrgovanih dolara.
- Grupe unutar kojih se kumulativno skuplja najviše kupovnih dolara brže se realiziraju u vremenu od grupa unutar kojih se kumulativno skuplja najviše prodajnih dolara.
- Grupa unutar koje se kumulativno skuplja najviše kupovnih dolara ne nanosi najveći rast cijene.
- Grupe kojima je kumulativna promjena cijene oko 0, također imaju kumulativnu promjenu istrgovanih dolara oko 0.
- U razdobljima prije velike volatilnosti određene grupe se počnu realizirati brže u vremenu, dok se neke druge grupe počnu realizirati sporije u vremenu.

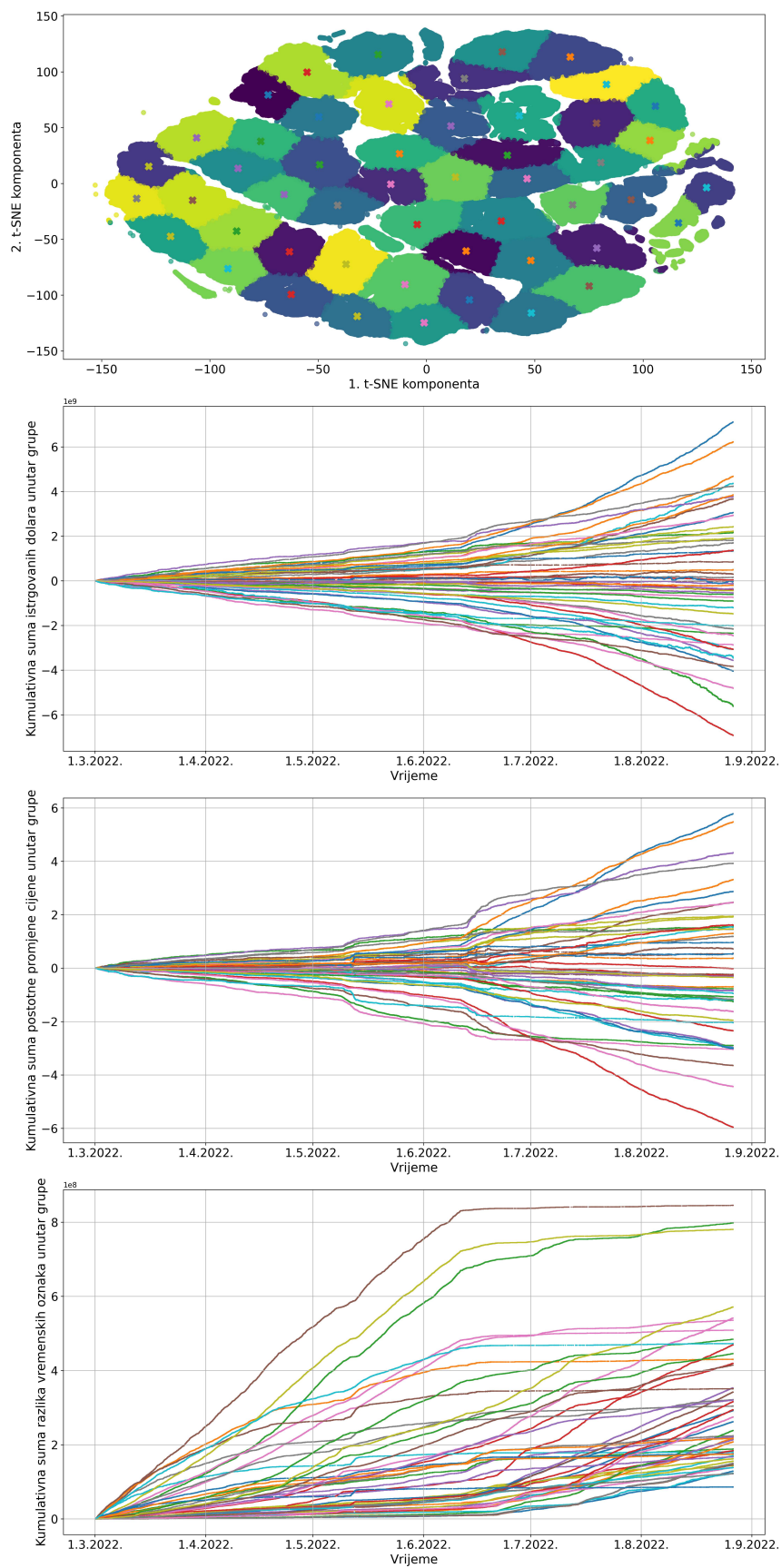
Podsjetimo se još jednom da ne znamo stvaran broj grupa, odnosno ne znamo je li jedna pravilnost potencijalno zbroj nekih drugih pravilnosti.



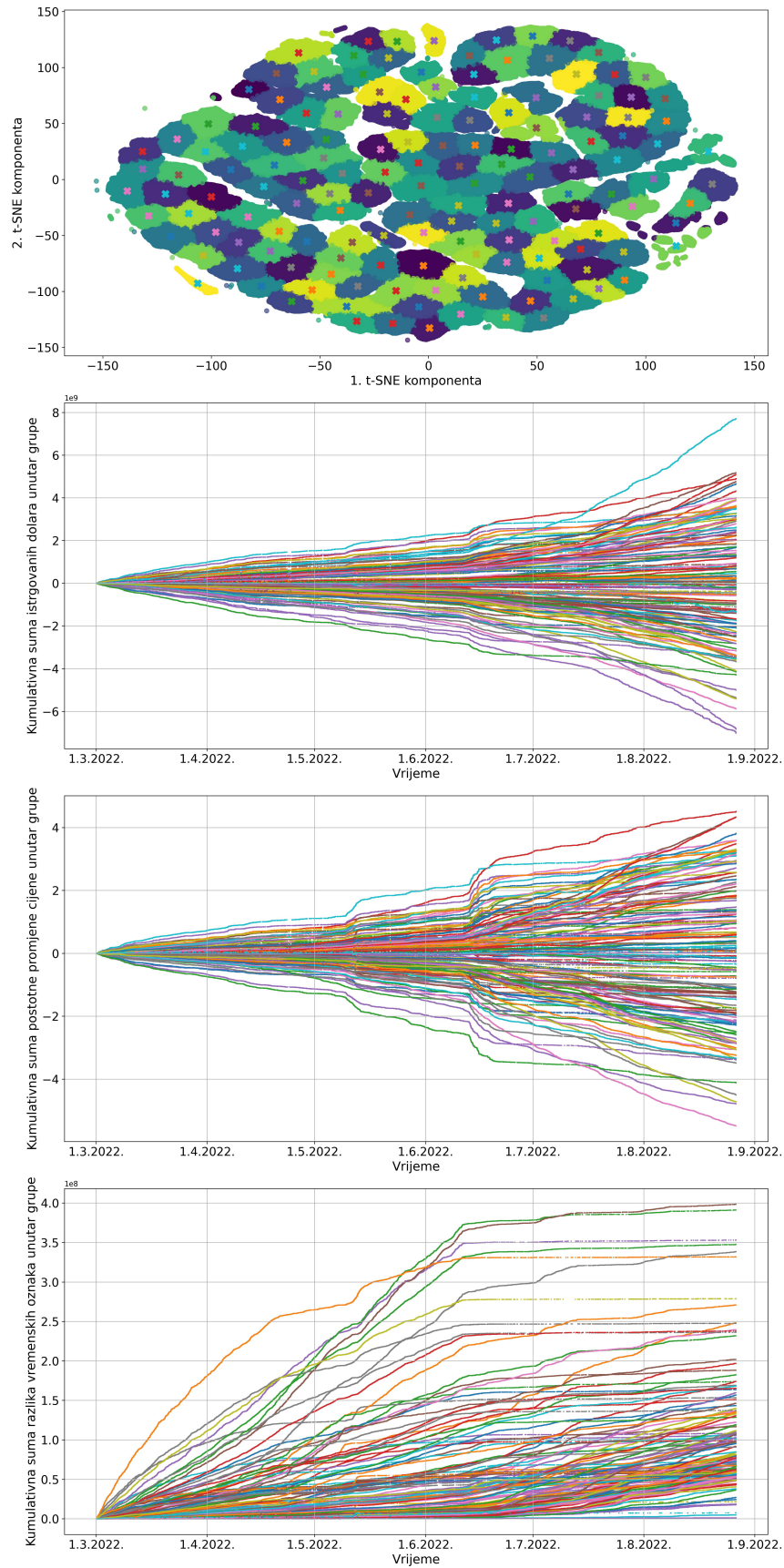
Slika 5.5. Cijena financijskog instrumenta ETHUSDT u razdoblju od 1.3.2022. do 1.9.2022. (testni skup transakcija).



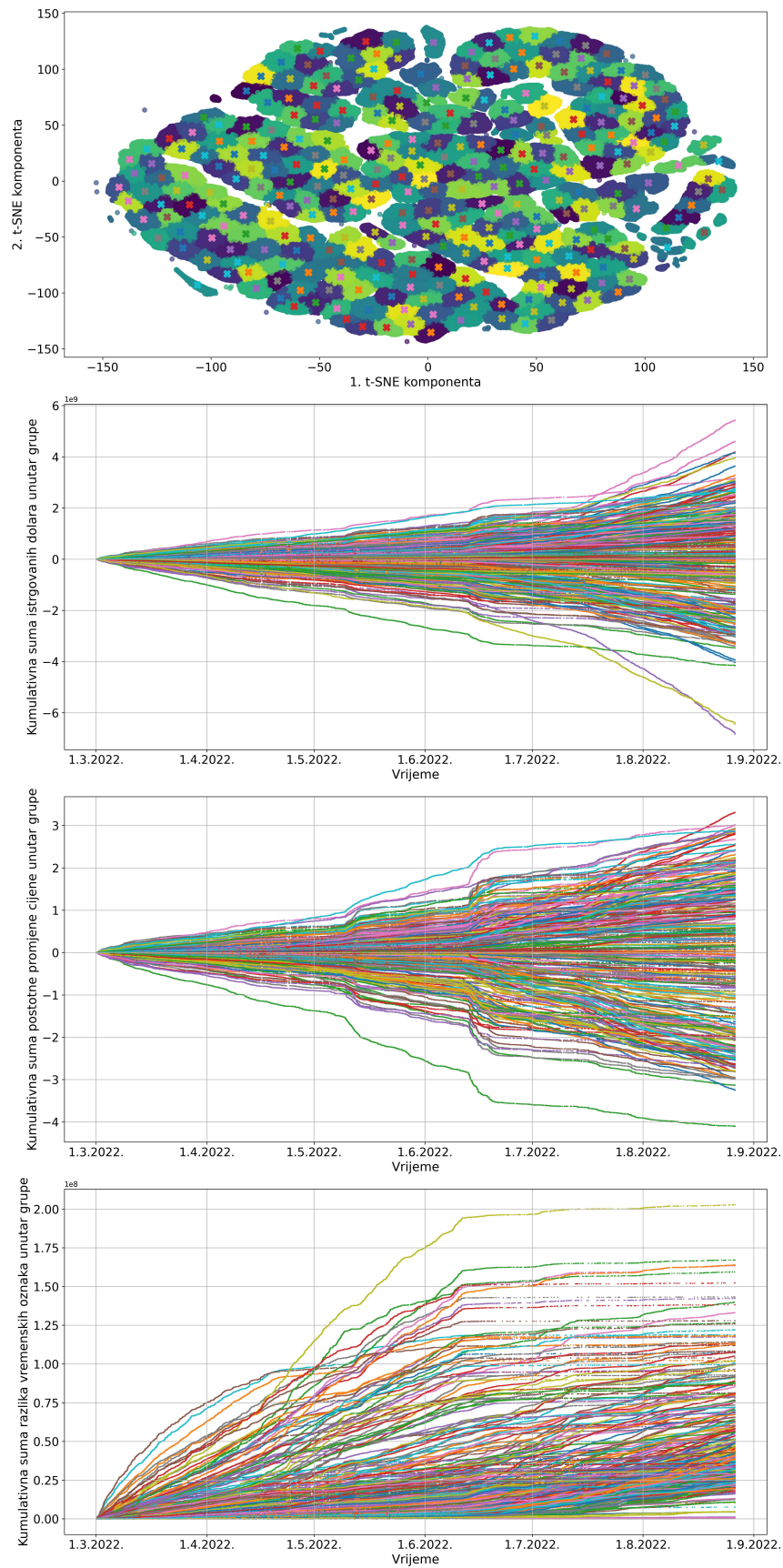
Slika 5.6. Kumulativne sume promjena unutar grupa. Grupe su formirane modelom Gaussove mješavine s 25 komponenti.



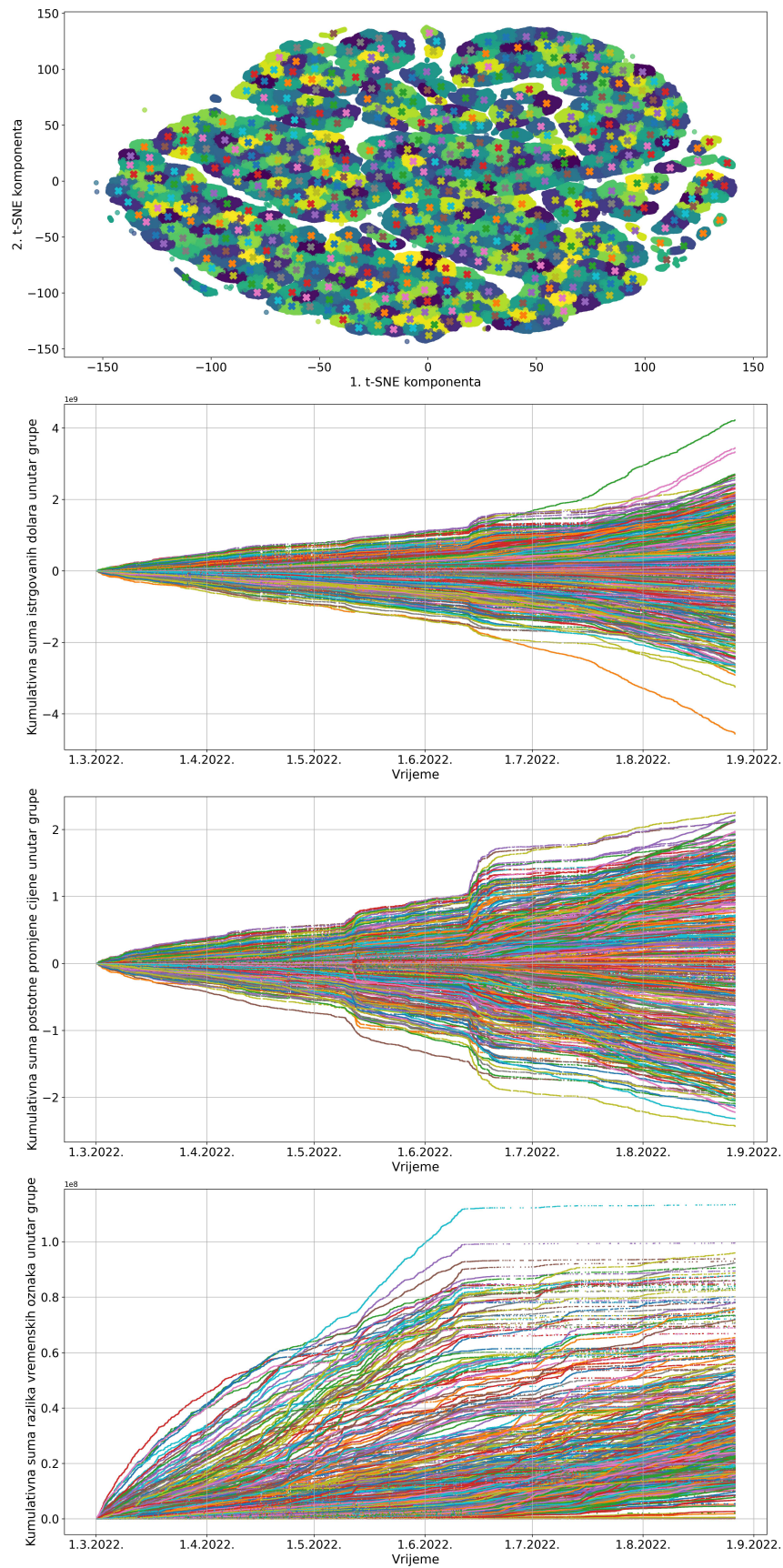
Slika 5.7. Kumulativne sume promjena unutar grupa. Grupe su formirane modelom Gaussove mješavine s 50 komponenti.



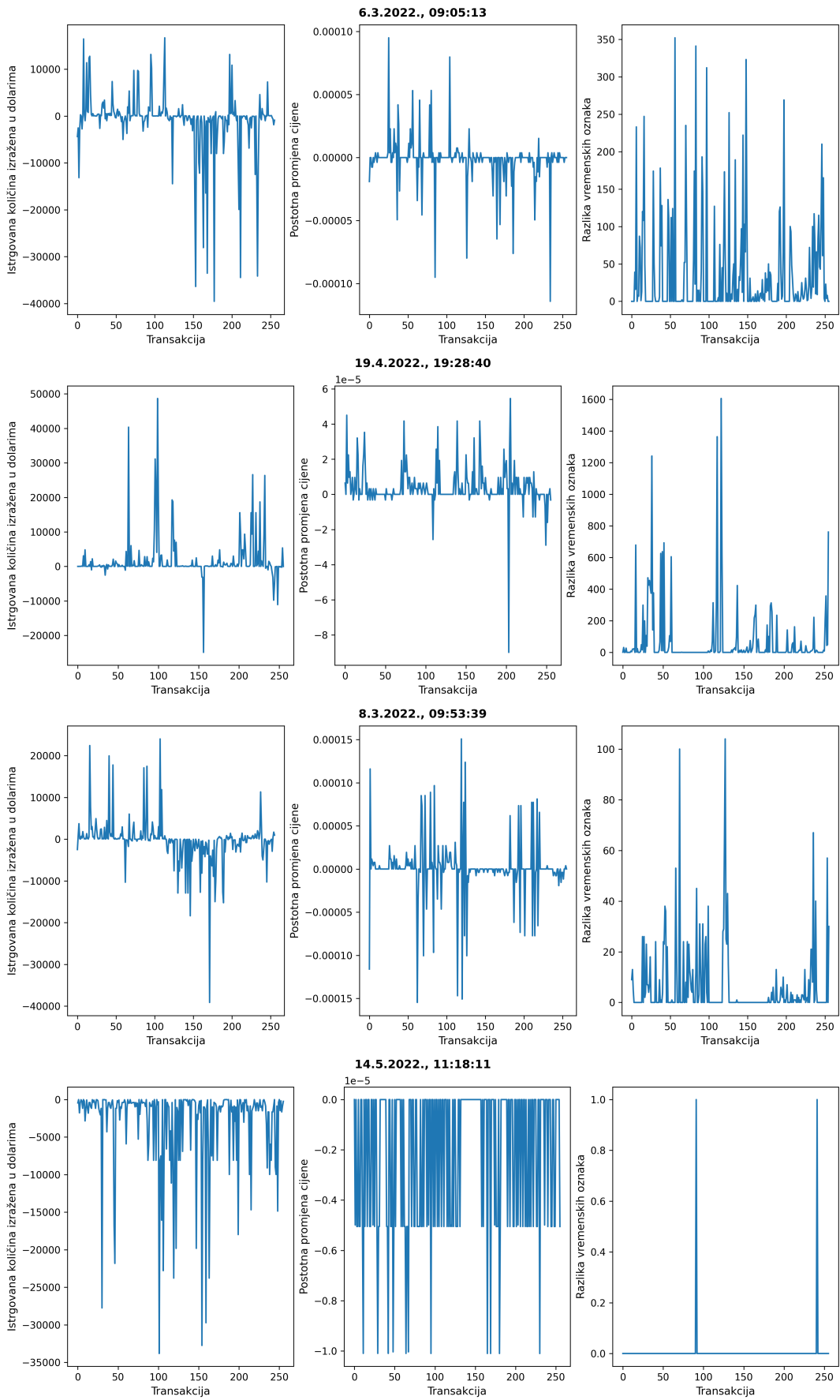
Slika 5.8. Kumulativne sume promjena unutar grupa. Grupe su formirane modelom Gaussove mješavine s 150 komponenti.



Slika 5.9. Kumulativne sume promjena unutar grupa. Grupe su formirane modelom Gaussove mješavine s 300 komponenti.



Slika 5.10. Kumulativne sume promjena unutar grupa. Grupe su formirane modelom Gaussove mješavine s 500 komponenti.



Slika 5.11. Uzorci transakcija s ekstremnim vrijednostima t-SNE komponenti.

Na slici 5.11. prikazani su uzorci transakcija koji su nakon redukcije dimenzionalnosti poprimili ekstremne vrijednosti t-SNE komponenti (najmanja i najveća vrijednost po prvoj t-SNE komponenti te najmanja i najveća vrijednost po drugoj t-SNE komponenti). Zanimljivo je kako se radi o skroz različitim uzorcima transakcija: prvi uzorak transakcija količinski sadrži više prodajnih dolara od kupovnih, postotna promjena cijene kreće se na skali između 10^{-4} i 10^{-5} te je razlika vremenskih oznaka u prosjeku oko 150 milisekunda; drugi uzorak transakcija količinski sadrži više kupovnih dolara od prodajnih, postotna promjena cijene kreće se na skali 10^{-5} te je razlika vremenskih oznaka u prosjeku oko 600 milisekunda; treći uzorak transakcija količinski sadrži malo više prodajnih dolara od kupovnih, postotna promjena cijene kreće se na skali između 10^{-4} i 10^{-5} te je razlika vremenskih oznaka u prosjeku oko 40 milisekunda; četvrti uzorak transakcija količinski sadrži pretežito prodajne dolare, postotna promjena cijene kreće se na skali 10^{-5} te je razlika vremenskih oznaka u prosjeku malo veća od 0 milisekunda. Možemo primijetiti da ovi uzorci međusobno predstavljaju značajno različita stanja tržišta.

6. Zaključak

Različita stanja tržišta predstavljena unutar ovog rada ne mogu biti detektirana OHLC svijećama, standardnom metodom agregiranja signala zadnjih izvršenih transakcija. Kako bi mogli detektirati ta stanja potrebno je transakcije preslikati u prostor dinamike dvostruke aukcije te pronaći alternativnu metodu agregiranja signala zadnjih izvršenih transakcija koja će biti informacijski robusnija od OHLC svijeća. Računanjem percentilnih vrijednosti nad cijelim skupom podataka transakcija financijskog instrumenta ETHUSDT primjećujemo koje izvršene količine izražene u dolarima, odnosno postotne promjene cijene i razlike vremenskih oznaka su učestalije, a koje spadaju u ekstremne slučajeve. S obzirom na tu informaciju definirali smo različita stanja tržišta uz pomoć kojih formiramo graf dinamike dvostruke aukcije. Koristeći neuronske mreže za grafove (unutar samonadziranog okruženja) provodimo učenje koje zahtjeva da grafovi koji sadrže slična stanja tržišta (te su topološki slični) u prostoru naučenih reprezentacija budu sličniji s obzirom na kosinusnu sličnost. Pokazuje se da susjedstva naučenih reprezentacija odražavaju određene pravilnosti u prostoru dinamike. S obzirom da su informacije o dinamici dvostruke aukcije ugrađene u naučenu reprezentaciju možemo tvrditi da takve reprezentacije sadrže više informacija od OHLC svijeća. Idući korak istraživanja jest korištenje naučenih reprezentacija kao ulaz u model koji bi predviđao iduće stanje tržišta (cijenu, razred dinamike dvostruke aukcije ili neku drugu vrijednosti). Svakako, predstavljeni pogled na dinamiku dvostruke aukcije i popratni rezultati koji potvrđuju postavljenu hipotezu otvaraju mogućnost istraživanja dinamike tržišta u novom smjeru.

Literatura

- [1] L. Wu, P. Cui, J. Pei, i L. Zhao, *Graph Neural Networks: Foundations, Frontiers, and Applications*. Singapore: Springer Singapore, 2022.
- [2] R. Winter, F. Noé, i D.-A. Clevert, “Permutation-invariant variational autoencoder for graph-level representation learning”, 2021.
- [3] T. N. Kipf i M. Welling, “Semi-supervised classification with graph convolutional networks”, 2017.
- [4] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, i Y. Bengio, “Graph attention networks”, 2018.
- [5] K. Xu, W. Hu, J. Leskovec, i S. Jegelka, “How powerful are graph neural networks?” 2019.
- [6] M. Xu, H. Wang, B. Ni, H. Guo, i J. Tang, “Self-supervised graph-level representation learning with local and global structure”, 2021.

Sažetak

Učenje reprezentacije dinamike dvostruke aukcije pomoću dubokih neuronskih mreža za grafove

Pavao Dužević

U ovom radu predstavljen je prostor dinamike dvostruke aukcije, metoda formiranja grafa dinamike dvostruke aukcije te je primijenjena metoda učenja reprezentacije grafova u samonadziranom okruženju. Grafovi dinamike dvostruke aukcije u sebi sadrže više informacija o trenutnom stanju tržišta u odnosu na OHLC svijeće, standardnu metodu agregiranja signala zadnjih izvršenih transakcija. Svrha ovog rada je ispitati informacije ugrađene unutar naučenih reprezentacija te je li naučene reprezentacije odražavaju pravilnosti unutar prostora dinamike dvostruke aukcije koje se ne mogu detektirati OHLC svijećama. Rezultati su dobiveni koristeći transakcijske podatke s burze kriptovaluta.

Ključne riječi: dvostruka aukcija; dinamika dvostruke aukcije; neuronske mreže za grafove; učenje reprezentacije; samonadzirano učenje; transakcijski podaci

Abstract

Learning the representation of double auction dynamics using deep graph neural networks

Pavao Dužević

This thesis presents the dynamics space of a double auction, the method of forming a double auction dynamics graph, and applies the method of learning graph-level representations in a self-supervised environment. Double auction dynamics graphs contain more information about the current market state compared to OHLC candlesticks, the standard method of aggregating signals from the last executed transaction. The purpose of this work is to examine the information embedded within the learned representations and whether the learned representations reflect patterns within the double auction dynamics space that cannot be detected by OHLC candlesticks. The results were obtained using transaction data from a cryptocurrency exchange.

Keywords: double auction; double auction dynamics; graph neural networks; representation learning; self-supervised learning; transaction data