VILNIAUS UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

Magistro baigiamasis darbas

Gilieji generatyviniai modeliai Baltijos akcijų grąžoms: Empirinis Wasserstein GAN vertinimas

Juozapas Kilikevičius

Darbo vadovas: Doc. Dr. Martynas Manstavičius

Vilnius
2025

Turinys

Santrauka

Finansinės laiko eilutės, tokios kaip akcijų indeksų kasdienės grąžos, pasižymi sudėtingomis statistinėmis savybėmis: sunkiomis uodegomis, nepastovumo grupavimusi, asimetrija ir netiesinėmis priklausomybėmis. Tradiciniai ekonometriniai modeliai (ARMA, GARCH) geba užfiksuoti tiesines priklausomybes ir sąlyginį nepastovumą, tačiau dažnai nepakankamai aprašo sudėtingesnius ryšius ir staigius rinkos režimų pasikeitimus.

Generatyviniai priešpriešiniai tinklai (angl. *Generative Adversarial Networks*, GAN) siūlo lankstų požiūrį sudėtingiems duomenų pasiskirstymams modeliuoti. Tačiau tradiciniai GAN, pagrįsti Jensen–Shannon nuokrypiu, nukenčia nuo mokymo nestabilumo, gradientų išnykimo ir modų kolapsavimo. Wasserstein GAN (WGAN) įveda Wasserstein atstumą kaip tikslo funkciją, užtikrinančią sklandesnius gradientus ir stabilesnį mokymą net esant dideliems pasiskirstymų skirtumams. Gradiento baudos metodas (angl. *gradient penalty*) užtikrina Lipšico sąlygą ir pagerina praktinį stabilumą.

Šiame darbe WGAN taikomas OMX Baltic Benchmark Gross indekso (OMXBBGI) kasdienių grąžų duomenims. Modelis vertinamas slenkančio lango metodu 2020–2024 m. periode: kiekvienas mokymo langas apima 2000 prekybos dienų istorinių duomenų (apie 8 metus), po to generuojamos 60 dienų prognozės, langas pastumiamas 20 dienų į priekį, ir procesas kartojamas.

Rezultatai rodo, kad WGAN pasiekia stabilų mokymą be modų kolapsavimo ir geba aproksimuoti centrinę pasiskirstymo dalį. Tačiau modelis patiria reikšmingų apribojimų: skirstinių testai (KS $p < 10^{-13}$, Wasserstein $W_1 = 0.995$, Didžiausias vidutinis neatitikimas $= 0.881$) rodo statistiškai reikšmingus skirtumus nuo tikrų duomenų; autokoreliacinės funkcijos atstumai viršija 1.0, liudydami nepakankamą nepastovumo klasterizavimo užfiksavimą; Pagrindinė šio darbo rezultatuose išryškėjusi problema – per plonos modeliuojamo skirstinio uodegos ir netinkamas ekstremalių įvykių modeliavimas, kas yra kritiškai svarbu rizikos valdymui.

Išvados patvirtina, kad nors WGAN architektūra suteikia stabilumo privalumų palyginti su tradiciniais GAN, ji nepakankama pilnai užfiksuoti fi-

nansinių grąžų dinamikos. Patikimam sintetinių duomenų generavimui būtini tolesni modelio tobulinimai, įtraukiantys laiko eilučių struktūros geresnį užfiksavimą, griežtesnius apribojimus ekstremalių reikšmių generavimui ir rizikos metrikų tiesioginio kalibravimo mechanizmus.

Raktiniai žodžiai: generatyviniai priešpriešiniai tinklai, Wasserstein GAN, finansinės laiko eilutės, akcijų grąžos, rizikos valdymas, sintetiniai duomenys

Summary

Financial time series, such as daily stock index returns, exhibit complex statistical properties: heavy tails, volatility clustering, asymmetry, and nonlinear dependencies. Traditional econometric models (ARMA, GARCH) can capture linear dependencies and conditional volatility, but often inadequately describe more complex relationships and quick market changes.

Generative Adversarial Networks (GAN) offer a flexible approach to modeling complex data distributions. However, traditional GANs based on the Jensen-Shannon divergence suffer from training instability, vanishing gradients, and mode collapse. Wasserstein GAN (WGAN) introduces the Wasserstein distance as the objective function, ensuring smoother gradients and more stable training even when distributions differ significantly. The gradient penalty method enforces the Lipschitz constraint and improves practical stability.

In this work, WGAN is applied to daily return data from the OMX Baltic Benchmark Gross Index (OMXBBGI). The model is evaluated using a rolling window approach over the 2020–2024 period: each training window covers 2000 trading days of historical data (around 8 years), then 60-day forecasts are generated, the window is shifted forward by 20 days, and the process repeats.

Results show that WGAN achieves stable training without mode collapse and can approximate the central part of the distribution. However, the model exhibits significant limitations: distribution tests (KS $p < 10^{-13}$, Wasserstein $W_1 = 0.995$, Maximum Mean Discrepancy $= 0.881$) show statistically significant differences from real data; autocorrelation function distances exceed 1.0, indicating insufficient capture of volatility clustering; VaR exceedance rates (7.95% and 16.38%) substantially exceed expected levels (1% and 5%), and backtesting tests reject the calibration null hypothesis. The primary issue is overly thin tails and inadequate modeling of extreme events, which is critical for risk management.

The conclusions confirm that while WGAN architecture provides stability advantages over traditional GANs, it remains insufficient to fully capture

the dynamics of financial returns. Reliable synthetic data generation requires further model improvements, including better capture of time series structure, stricter constraints on generating extreme values, and direct calibration mechanisms for risk metrics.

Keywords: generative adversarial networks, Wasserstein GAN, financial time series, stock returns, risk management, synthetic data

1 skyrius

Įvadas

Sintetinių finansinių duomenų generavimas tapo vis svarbesne tema kiekybinėse finansuose, rizikos valdyme ir finansinėse technologijose. Sintetiniai duomenys leidžia tyrėjams tikrinti prekybos strategijas, vertinti rizikos modelius ir kurti naujus algoritmus kontroliuojamoje aplinkoje neatskleisdami jautrios ar konfidencialios informacijos. Taip pat yra palengvinamas streso testavimas, scenarijų analizė ir patikimų mašininio mokymosi modelių kūrimas, suteikiant įvairius ir realistiškus duomenų rinkinius [1, 2].

Pastaraisiais metais reguliavimo institucijos pabrėžė patikimo modelių patvirtinimo ir streso testavimo svarbą, ypač po finansinių krizių. Sintetinių duomenų generavimas, ne tik finansų srityje, yra gerai, nes atitinka duomenų privatumo reglamentams (pvz., GDPR) ir leidžia institucijoms dalytis duomenimis bendradarbiaujant tyrimuose nepažeidžiant konfidencialumo [3].

Finansinės laiko eilutės, tokios kaip akcijų kasdienės grąžos, pasižymi sudėtingomis statistinėmis savybėmis, įskaitant nepastovumo grupavimąsi, sunkias uodegas, asimetriją ir netiesines priklausomybes [4, 5]. Tradiciniai ekonometriniai modeliai buvo plačiai naudojami tiesinėms priklausomybėms ir nepastovumo dinamikai finansiniuose duomenyse užfiksuoti [6, 7].

Tačiau šie modeliai dažnai nepakankamai gerai aprašo sudėtingesnes priklausomybes, staigius rinkos režimų pasikeitimus ir kitus netiesinių ryšių atvejus, pastebimus realiose finansų rinkose [4, 8].

Giluminio mokymosi pažanga lėmė generatyvinių priešpriešinių tinklų (GAN) vystymąsi, kurie siūlo lankstų ir galingą sistemą sudėtingiems duomenų pasiskirstymams modeliuoti [9]. GAN parodė puikius rezultatus generuojant realistiškus sintetinius duomenis įvairiose srityse, ypač kompiuterinio matymo užduotyse, tokiose kaip nuotraukų generavimas, veido sintezė ir vaizdų transformacijos. Finansuose GAN buvo taikomi akcijų grąžoms, užsakymų knygos (angl. Order Book) duomenims ir duomenų kintamumą generuoti [10, 11]. Skirtingai nuo klasikinių modelių, GAN gali užfiksuo-

ti sudėtingus, netiesius ryšius ir atkartoti stilizuotus faktus, kuriuos sunku modeliuoti tradiciniais metodais [1, 12].

Tačiau, tradiciniai GAN finansinėms laiko eilutėms kelia specifinių iššūkių. Originalūs GAN naudoja Jensen-Shannon nuokrypį, kuris gali sukelti mokymo nestabilumą ir gradientų išnykimą, ypač kai tikrų ir sintetinių duomenų pasiskirstymai nesutampa - dažna situacija su finansinėmis grąžomis, pasižyminčiomis sunkiomis uodegomis ir nepastovumo klasterizavimu. Wasserstein GAN (WGAN) siūlo alternatyvią geometriją, pagrįstą Wasserstein atstumu, kuris suteikia sklandesnius gradientus ir stabilesnį mokymą net kai pasiskirstymai yra labai skirtingi. Ši savybė yra ypač reikšminga finansiniams duomenims, kur ekstremalūs įvykiai ir struktūriniai pokyčiai yra dažni [13].

Nepaisant jų perspektyvumo, tiek GAN, tiek WGAN kelia didelių iššūkių, įskaitant mokymo nestabilumą, vertinimo sunkumus ir interpretuojamumo stoką [13, 14]. Dėl to literatūroje vyksta nuolatinė diskusija dėl klasikinių statistinių modelių ir šiuolaikinių mašininio mokymosi metodų santykinių pranašumų sintetiniams finansiniams duomenims generuoti.

Nors plati literatūra tyrinėja tiek klasikinius ekonometrinius modelius, tiek giluminius generatyvinius modelius, pagrindinis praktinis šio darbo klausimas yra, ar šiuolaikiniai priešpriešiniai generatoriai gali pateikti ekonomiškai patikimas sintetines grąžas konkrečiai rinkai.

Šiame darbe Wassersteino GAN taikomas OMX Baltic Benchmark Gross indekso (OMXBBGI) dienos grąžoms, siekiant įvertinti, ar Wasserstein atstumo įvedimas pagerina mokymo stabilumą ir generavimo kokybę, palyginti su standartiniu Jensen–Shannon pagrindu [13, 15]. Darbas apima teorinį GAN ir WGAN pagrindą, metodologijos aprašymą, empirinius rezultatus OMXBBGI duomenyse ir diskusiją apie likusius iššūkius bei būsimus tyrimus.

2 skyrius

Generatyviniai priešpriešiniai tinklai finansinėms laiko eilutėms

2.1 Įvadas

Generatyviniai priešpriešiniai tinklai (GAN) yra numanomojo generavimo modelių klasė, pristatyta Goodfellow ir bendaautorių [9]. Jų apibrėžiantis bruožas yra priešiškas mokymo tikslas, kuriame generatoriaus tinklas G_θ (parametrizuotas θ) mokosi kurti sintetines imtis, kurios norima, kad būtų neatskirtos nuo tikrų duomenų, o diskriminatoriaus tinklas D_ϕ (parametrizuotas ϕ) mokosi atskirti tikras imtis nuo sugeneruotų. Tinklai yra mokomi vienu metu dviejų žaidėjų minimax žaidime.

Formaliai, tarkime, kad $p_{\text{data}}(x)$ žymi tikrąjį, mums nežinomą duomenų pasiskirstymą aibėje $\mathcal{X}$. Šiame darbe $\mathcal{X} = \mathbb{R}^L$ yra baigtinio ilgio sekų erdvė, kur $L \in \mathbb{N}$ žymi fiksuotą laiko eilutės ilgį (pavyzdžiui, kasdienių grąžų skaičių). Kitaip tariant, kiekvienas $x \in \mathcal{X}$ yra baigtinio ilgio finansinių grąžų seka $(x_1, x_2, \ldots, x_L) \in \mathbb{R}^L$.

Erdvėje $\mathcal{X}$ apibrėžta standartinė Euklido norma, kurią lemia skaliarinė sandauga $\langle x, y \rangle = \sum_{i=1}^{L} x_i y_i$:

$$\|x\| = \sqrt{\sum_{i=1}^{L} x_i^2}. \tag{2.1}$$

Ši norma leidžia apibrėžti metriką

$$d(x, y) = \|x - y\| = \sqrt{\sum_{i=1}^{L} (x_i - y_i)^2}. \tag{2.2}$$

Metrika nusako erdvės topologiją per atvirus rutulius

$$B_\epsilon(x) = \{y \in \mathbb{R}^L : d(x,y) < \epsilon\}, \quad x \in \mathcal{X}, \varepsilon > 0. \tag{2.3}$$

Visas aibes galima sukonstruoti iš atvirų intervalų, pasitelkiant skaičių sąjungas, sankirtas ir papildinius [16]. Tokia topologinė struktūra leidžia apibrėžti Borelio σ-algebrą $\mathcal{B}(\mathbb{R}^L)$ kaip mažiausią σ-algebrą, kurioje yra visos atvirosios aibės. Kadangi $\mathbb{R}^L$ yra baigtinės dimensijos Euklido erdvė, joje egzistuoja standartinis L-matis Lebego matas λ^L, apibendrininantis ilgio, ploto ir tūrio sąvokas L matmenų erdvėje [17].

Bendrąja prasme, $p(x)$ gali žymėti tiek tikimybių tankio funkciją absoliučiai tolydiems atsitiktiniams dydžiams (pavyzdžiui, vaizdų ar garso bangų generavimui), tiek tikimybių masės funkciją diskretiems dydžiams (pavyzdžiui, teksto ar kategorijų generavimui).

Finansinių grąžų modeliavime pasiskirstymas laikomas absoliučiai tolydžiu Lebego mato λ^L prasme [4]. Tai reiškia, kad egzistuoja tikimybių tankio funkcija tokia, kad

$$P(X \in A) = \int_A p_{\text{data}}(x)\, dx \tag{2.4}$$

bet kuriai Borelio aibei $A \subseteq \mathcal{X}$. Nors praktikoje stebime tik baigtinį duomenų skaičių, tolydžio atsitiktinio dydžio su tankio funkcija modeliavimas yra standartinė ekonometrinės literatūros praktika [18]. Todėl toliau $p_{\text{data}}(x)$ ir $p_{\text{g}}(x)$ žymi tikimybių tankio funkcijas.

Taip pat įvedamas paprastas, iš anksto žinomas tikimybinis skirstinys, aprašomas tankio funkcija $p_Z(z)$, iš kurio bus imamas atsitiktinis triukšmas. Dažniausiai naudojamas daugiamatis normalusis skirstinys $\mathcal{N}(0, I_d)$, kur I_d yra $d \times d$ matmenų vienetinė matrica, apibrėžta kaip

$$I_d = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}. \tag{2.5}$$

Vienetinės matricos naudojimas reiškia, kad kiekviena triukšmo vektoriaus komponentė Z_i yra nepriklausoma nuo kitų ir turi vienetinę dispersiją. Toks pasirinkimas suteikia lankstumą – generatorius pats išmoksta transformuoti šį paprastą triukšmą į sudėtingas priklausomybes duomenyse, todėl nereikia iš anksto spėlioti, kokia kovariacinė struktūra reikalinga.

Generatorius tada matematiškai apibrėžiamas kaip funkcija

$$G_\theta : \mathcal{Z} \to \mathcal{X}, \qquad X = G_\theta(Z),\ Z \sim p_Z(z), \tag{2.6}$$

kur $\mathcal{Z} \subseteq \mathbb{R}^d$ yra nestebimo triukšmo erdvė (angl. *latent space*). Generatorius paima atsitiktinį vektorių Z iš paprastos erdvės $\mathcal{Z}$ ir transformuoja jį į sudėtingesnę duomenų erdvę $\mathcal{X}$. Šis transformavimo procesas sukuria naują modelio pasiskirstymą $p_{\mathrm{g}}(x)$ (indeksas "g" rodo generatoriaus pasiskirstymą) per push-forward matą: sugeneruotų duomenų tikimybinis matas yra pradinio triukšmo skirstinio transformacija per generatoriaus funkciją (formalus apibrėžimas pateiktas vėliau šiame skyriuje).

Diskriminatorius grąžina skaitinę reikšmę $[0, 1]$ intervale ir gali būti interpretuojamas kaip tikimybės, kad jo įvestis yra tikra, įvertis:

$$D_\phi : \mathcal{X} \to [0, 1]. \tag{2.7}$$

Reikšmė $D_\phi(x) = 1$ reiškia, kad diskriminatorius mano, jog įvestis x yra iš p_{data}, o $D_\phi(x) = 0$ reiškia, kad ji yra sugeneruota iš p_{g}. Diskriminatoriaus funkcija turi priklausyti Lipschitz tolydžių funkcijų klasei, žymima $\mathcal{D}$ (apibrėžimai bus pateikti vėliau), ir tenkinti kelis matematinius reikalavimus, būtinus tolesnei analizei. Pirma, D_ϕ turi būti Borelio funkcija, t.y. bet kuriai Borelio aibei $B \in \mathcal{B}(\mathbb{R})$ pirmvaizdis (angl. Preimage) $D_\phi^{-1}(B)$ priklauso Borelio σ-algebrai $\mathcal{B}(\mathbb{R}^L)$. Šis matumas garantuoja, kad kompozicija $\log D_\phi(X)$ su atsitiktiniu dydžiu X išlieka mati funkcija, todėl matematinis vidurkis $\mathbb{E}_{X \sim p_{\mathrm{data}}}[\log D_\phi(X)]$ egzistuoja kaip Lebego integralas ir yra gerai apibrėžtas. Antra, diskriminatorius turi būti tolydi funkcija su reikšmėmis kompaktinėje aibėje $[0, 1]$. Tolydumas užtikrina Borelio matumą ir yra būtinas sąlyga tam, kad diskriminatorių klasė būtų kompaktiška tolygaus konvergavimo topologijoje (kaip bus parodyta Arzela-Ascoli teoremos taikymu). Tai leis pritaikyti Weierstrass teoremą funkcionalo $V(D, G)$ maksimizavimui funkcijų erdvėje, kas garantuoja optimalaus diskriminatoriaus D^* egzistavimą. Kadangi $D(x) \in [0, 1]$, logaritmai $\log D(x)$ ir $\log(1 - D(x))$ gali būti neigiami arba siekti neigiamą begalybę, kai $D(x) \to 0$ arba $D(x) \to 1$. Tačiau neuroninio tinklo sigmoid aktyvacija užtikrina, kad diskriminatorius lieka griežtai tarp nulio ir vieneto, todėl vertės funkcijos integralas egzistuoja (nors gali būti neigiamas). Galiausiai, diferencijuojamumas beveik visur atvirame intervale $(0, 1)$ yra būtinas gradientiniam optimizavimui. Kadangi θ yra generatoriaus parametrai, išvestinės $\frac{\partial}{\partial \theta}\mathbb{E}[\log D_\phi(G_\theta(Z))]$ skaičiavimas pagal grandininę taisyklę reikalauja tiek diskriminatoriaus D_ϕ diferencijuojamumo pagal jo argumentą (kompozicijoje $D_\phi \circ G_\theta$), tiek generatoriaus G_θ diferencijuojamumo pagal parametrus θ (kaip parodyta generatoriaus diferencijavimo parametrų atžvilgiu žemiau). Tai užtikrina grįžtamojo perdavimo (angl. backpropagation) algoritmo veikimą mokymo metu. Praktikoje diskriminatorius realizuojamas neuroniniu tinklu su standartinėmis aktivacijos funkcijomis [19], kurios natūraliai garantuoja šias savybes. Toliau šiame skyriuje šios savybės bus įrodytos matematiškai.

Diskriminatoriaus matumas. Įrodysime, kad neuroninis tinklas yra matus Borelio sigma-algebrų atžvilgiu. Diskriminatorius $D_\phi : \mathbb{R}^L \to [0,1]$ sudarytas iš kelių sluoksnių. Kiekviename sluoksnyje atliekama afininė transformacija $h_i(x) = W_i x + b_i$, po kurios veikia aktivacijos funkcija a_i (sigmoid, tanh arba ReLU). Paskutiniame sluoksnyje naudojama sigmoid funkcija

$$\sigma(t) = \frac{1}{1+e^{-t}}, \tag{2.8}$$

kuri užtikrina, kad $D_\phi(x) \in [0,1]$. Visa diskriminatoriaus struktūra užrašoma kaip funkcijų kompozicija:

$$D_\phi(x) = \sigma\Big(h_{n-1} \circ a_{n-1} \circ \cdots \circ a_1 \circ h_0(x)\Big). \tag{2.9}$$

Pirma bus įrodyta, kad afininės transformacijos $h_i : \mathbb{R}^L \to \mathbb{R}^m$ yra Borelio funkcijos. Afinę funkciją $h_i(x) = W_i x + b_i$ galima užrašyti koordinatėmis. Kiekviena koordinatė $j = 1, \ldots, m$ apibrėžiama taip:

$$h_i(x)_j = \sum_{k=1}^{L} W_{ijk} x_k + b_{ij}. \tag{2.10}$$

Kiekviena koordinatė x_k yra mati funkcija. Projekcija $\pi_k : \mathbb{R}^L \to \mathbb{R}$, apibrėžta kaip $\pi_k(x) = x_k$, yra tolydi, nes bet kuriam $x, y \in \mathbb{R}^L$ galioja

$$|\pi_k(x) - \pi_k(y)| = |x_k - y_k| \leq \|x - y\|,$$

o tolydi funkcija yra mati [17]. Suma ar sandauga matuojamų funkcijų irgi yra mati funkcija [20]. Vadinasi, kiekviena koordinatė $h_i(x)_j$ yra mati. Funkcija $f : \mathbb{R}^L \to \mathbb{R}^m$ yra Borelio tada ir tik tada, kai kiekviena jos koordinatė f_j yra mati [17]. Todėl h_i afininė transformacija yra Borelio funkcija.

Antra bus įrodyta, jog standartinės aktivacijos funkcijos yra Borelio funkcijos. Sigmoid funkcija $\sigma(t) = (1+e^{-t})^{-1}$ griežtai didėja, todėl bet kurio intervalo (a,b) pirmvaizdis $\sigma^{-1}((a,b))$ yra intervalas, kuris priklauso $\mathcal{B}(\mathbb{R})$. Tas pats galioja hiperboliniam tangentui $\tanh(t)$. ReLU funkcija $\mathrm{ReLU}(t) = \max(0,t)$ apibrėžta dalimis: $\mathrm{ReLU}(t) = 0$ kai $t \leq 0$ ir $\mathrm{ReLU}(t) = t$ kai $t > 0$. Bet kurios atviros aibės pirmvaizdis yra intervalas arba intervalų sąjunga, todėl priklauso $\mathcal{B}(\mathbb{R})$. Kai aktivacija taikoma kiekvienai vektoriaus koordinatei atskirai, matumas išlieka [17].

Trečia teigsime, tai kad mačių funkcijų kompozicija yra mati. Tarkime, $f : \mathbb{R}^L \to \mathbb{R}^m$ ir $g : \mathbb{R}^m \to \mathbb{R}^k$ abi yra Borelio funkcijos. Bet kuriai Borelio aibei $C \in \mathcal{B}(\mathbb{R}^k)$ kompozicijos pirmvaizdis užrašomas:

$$(g \circ f)^{-1}(C) = f^{-1}\Big(g^{-1}(C)\Big). \tag{2.11}$$

Kadangi g yra mati, tai $g^{-1}(C) \in \mathcal{B}(\mathbb{R}^m)$. Kadangi f yra mati, tai $f^{-1}(g^{-1}(C)) \in \mathcal{B}(\mathbb{R}^L)$. Vadinasi, $(g \circ f)^{-1}(C) \in \mathcal{B}(\mathbb{R}^L)$, todėl kompozicija yra mati.

Dabar pažiūrėkime į visą diskriminatorių $D_\phi(x) = \sigma(h_{n-1} \circ a_{n-1} \circ \ldots \circ a_1 \circ h_0(x))$. Pirmasis sluoksnis h_0 yra matus (įrodyta koordinatėmis). Sudėtis $a_1 \circ h_0$ taip pat mati (nes abi funkcijos yra mačios). Pridėję antrąjį sluoksnį $h_1 \circ a_1 \circ h_0$, vėl gauname mačią funkciją. Kartojant šį procesą kiekvienam sluoksniui, galiausiai visa kompozicija D_ϕ yra mati. Vadinasi, bet kuriai $B \in \mathcal{B}(\mathbb{R})$ galioja $D_\phi^{-1}(B) \in \mathcal{B}(\mathbb{R}^L)$.

Diskriminatoriaus tolydumas. Parodysime, kad neuroninio tinklo struktūra garantuoja tolydumą. Diskriminatorius D_ϕ kaip kompozicija kelių funkcijų bus tolydus, jei kiekviena sudedamoji funkcija yra tolydi.

Pirma, tiesinės transformacijos $h_i(x) = W_i x + b_i$ yra tolydžios funkcijos, nes bet kuriam $x, y \in \mathbb{R}^L$ galioja

$$\|h_i(x) - h_i(y)\| = \|W_i(x - y)\| \leq \|W_i\| \cdot \|x - y\|, \tag{2.12}$$

kur $\|W_i\|$ yra operatorinė matricos norma, gauta iš Euklido normos vektoriuose, apibrėžta kaip $\|W_i\| = \sup_{\|x\| \neq 0} \frac{\|W_i x\|}{\|x\|}$. Ši nelygybė gauta taikant Cauchy-Schwarz nelygybę ir rodo, kad nedidelis įvesties pokytis $\|x - y\|$ sukelia nedidelį išvesties pokytį, kas ir apibrėžia tolydumą.

Antra, standartinės aktivacijos funkcijos yra tolydžios. Sigmoid funkcija $\sigma(t) = (1 + e^{-t})^{-1}$ ir hiperbolinis tangentas $\tanh(t)$ yra tolydžios visoje realiųjų skaičių ašyje kaip elementarių tolydžių funkcijų kompozicijos [17]. ReLU funkcija $\mathrm{ReLU}(t) = \max(0, t)$ taip pat yra tolydi, nors jos išvestinė neegzistuoja taške $t = 0$.

Trečia, tolydžių funkcijų kompozicija yra tolydi funkcija. Jei $f : \mathbb{R}^L \to \mathbb{R}^m$ ir $g : \mathbb{R}^m \to \mathbb{R}^k$ yra tolydžios, tai kompozicija $g \circ f : \mathbb{R}^L \to \mathbb{R}^k$ taip pat tolydi. Jei $x_n \to x$, tai iš f tolydumo seka $f(x_n) \to f(x)$, o iš g tolydumo seka $g(f(x_n)) \to g(f(x))$, taigi $g \circ f$ yra tolydi [17].

Kadangi diskriminatorius D_ϕ yra sudarytas iš baigtinio skaičiaus tolydžių funkcijų (tiesinių transformacijų ir aktivacijų), taikant tą patį argumentą kiekvienam sluoksniui iš eilės, gauname, kad visas tinklas yra tolydus. Kadangi diskriminatorius diferencijuojamas tik beveik visur (dėl ReLU), tolydumo įrodymas yra būtinas - diferencijavimas beveik visur neimplikuoja tolydumo. Tolydumas bus reikalingas vėliau apibrėžiant Lipschitz funkcijų klasę $\mathcal{D}$, kuriai taikysime Arzela-Ascoli ir Weierstrass teoremas optimalaus diskriminatoriaus D^* egzistavimui įrodyti.

Diskriminatoriaus diferencijavimas. Parodysime, kad neuroninio tinklo struktūra užtikrina diferencijavimą beveik visur. Diskriminatorius D_ϕ bus

diferencijuojamas, jei kiekviena sudedamoji funkcija yra diferencijuojama ir galime taikyti grandinės taisyklę.

Pirma, tiesinės transformacijos $h_i(x) = W_i x + b_i$ yra diferencijuojamos visur, o jų Jacobi matrica yra tiesiog W_i:

$$\frac{\partial h_i}{\partial x} = W_i. \tag{2.13}$$

Tai reiškia, kad kiekvienos tiesinės transformacijos gradientas egzistuoja ir yra pastovus.

Antra, standartinės aktivacijos funkcijos yra diferencijuojamos beveik visur. Sigmoid funkcija $\sigma(t) = (1 + e^{-t})^{-1}$ turi išvestinę

$$\sigma'(t) = \sigma(t)(1 - \sigma(t)), \tag{2.14}$$

kuri egzistuoja visur. Hiperbolinis tangentas $\tanh(t)$ taip pat turi išvestinę $\tanh'(t) = 1 - \tanh^2(t)$ visur. ReLU funkcija $\mathrm{ReLU}(t) = \max(0, t)$ turi išvestinę

$$\mathrm{ReLU}'(t) = \begin{cases} 1, & t > 0 \\ 0, & t < 0 \end{cases} \tag{2.15}$$

visur, išskyrus tašką $t = 0$. Kadangi vieno taško aibė turi nulinį Lebego matą, ReLU yra diferencijuojama beveik visur [21].

Trečia, diferencijuojamų funkcijų kompozicija yra diferencijuojama pagal grandinės taisyklę. Jei $f : \mathbb{R}^L \to \mathbb{R}^m$ ir $g : \mathbb{R}^m \to \mathbb{R}^k$ yra diferencijuojamos, tai kompozicijos $g \circ f : \mathbb{R}^L \to \mathbb{R}^k$ išvestinė yra

$$\frac{\partial (g \circ f)}{\partial x} = \frac{\partial g}{\partial f} \cdot \frac{\partial f}{\partial x}, \tag{2.16}$$

kur dešinėje matricų sandauga [17].

Kadangi diskriminatorius D_ϕ yra sudarytas iš baigtinio skaičiaus diferencijuojamų funkcijų (tiesinių transformacijų ir aktivacijų), kartojant tą pačią logiką su kiekvienu sluoksniu, gauname, kad visas tinklas yra diferencijuojamas beveik visur. Gradientas $\nabla_x D_\phi(x)$ egzistuoja beveik visiems $x \in \mathbb{R}^L$ (Lebego mato prasme) ir gali būti apskaičiuotas atgalinio sklidimo algoritmu. Nors diferencijavimo taškų, kur gradientas neegzistuoja (pvz., ReLU lūžio taškai), aibė turi Lebego matą nulį, tai netrukdo gradientiniam optimizavimui praktikoje, nes tokių taškų tikimybė yra nulinė [21].

Generatoriaus matematinės savybės. Generatorius $G_\theta : \mathbb{R}^d \to \mathbb{R}^L$ taip pat realizuojamas neuroniniu tinklu su standartinėmis aktivacijos funkcijomis, todėl pagal tuos pačius argumentus, taikytus diskriminatoriui, jis yra

tolydus, diferencijuojamas beveik visur pagal įvestį $z \in \mathbb{R}^d$, ir Borelio matus. Tačiau yra kelios svarbios savybės, unikalios generatoriui.

Pirma, išvesties erdvė. Skirtingai nuo diskriminatoriaus, kurio išvestis aprėžta $[0, 1]$ dėl sigmoid funkcijos paskutiniame sluoksnyje, generatorius kuria vektorius $x \in \mathbb{R}^L$ be apribojimų. Tai būtina finansinėms grąžoms, kurios gali būti tiek teigiamos (pelnas), tiek neigiamos (nuostolis). Todėl generatoriaus paskutinis sluoksnis paprastai neturi sigmoid aktivacijos.

Antra, sugeneruotų duomenų pasiskirstymas. Generatorius G_θ indikuoja tikimybinį matą p_{g} duomenų erdvėje $\mathcal{X}$. Bet kuriai Borelio aibei $A \subseteq \mathcal{X}$ galioja

$$p_{\mathrm{g}}(A) = P(G_\theta(Z) \in A) = P(Z \in G_\theta^{-1}(A)) = \int_{G_\theta^{-1}(A)} p_Z(z)\, dz. \tag{2.17}$$

Kadangi G_θ yra Borelio funkcija, pirmvaizdis $G_\theta^{-1}(A)$ priklauso Borelio σ-algebrai $\mathcal{B}(\mathbb{R}^d)$, todėl p_{g} yra gerai apibrėžtas tikimybinis matas.

Svarbu pastebėti, kad nors p_{g} egzistuoja kaip matas, aiški tankio funkcija $p_{\mathrm{g}}(x)$ gali neegzistuoti tradicine Lebego mato prasme. Pavyzdžiui, jei triukšmo dimensija $d < L$, sugeneruoti duomenys neužpildo visos $\mathbb{R}^L$ erdvės, o koncentruojasi žemesnės dimensijos daugdaroje (angl. *manifold*), todėl Lebego tankis neapibrėžtas. Tačiau GAN formulėje tai netrukdo: vertės funkcijos narys $\mathbb{E}_{x\sim p_{\mathrm{g}}}[\log(1 - D(x))]$ apskaičiuojamas pakeičiant $z \sim p_Z$ ir transformuojant $x = G_\theta(z)$, nereikalaujant aiškaus $p_{\mathrm{g}}(x)$ tankio išraiškos.

Ketvirta, diferencijavimas parametrų atžvilgiu. Grįžtamojo perdavimo algoritmas reikalauja, kad generatorius būtų diferencijuojamas ne tik pagal įvestį z, bet ir pagal parametrus θ. Kadangi kiekvienas sluoksnis $h_i(z) = W_i z + b_i$ yra diferencijuojamas pagal W_i ir b_i su Jacobi matricomis $\frac{\partial h_i}{\partial W_i} = z^T$ ir $\frac{\partial h_i}{\partial b_i} = I$, grandinės taisyklė užtikrina, kad $\frac{\partial G_\theta(z)}{\partial \theta}$ egzistuoja beveik visur (išskyrus ReLU lūžio taškus, kurie turi Lebego matą nulį). Tai leidžia apskaičiuoti generatoriaus gradientą:

$$\frac{\partial}{\partial \theta} D_\phi(G_\theta(z)) = \nabla_x D_\phi(G_\theta(z)) \cdot \frac{\partial G_\theta(z)}{\partial \theta}, \tag{2.18}$$

kur pirmasis narys yra diskriminatoriaus gradientas, o antrasis narys yra generatoriaus Jacobi matrica pagal parametrus.

Penkta, kompozicijos savybės. Kompozicija $D_\phi \circ G_\theta : \mathbb{R}^d \to [0, 1]$ yra tolydi ir diferencijuojama funkcija kaip dviejų tokių funkcijų kompozicija. Tai užtikrina, kad vertės funkcijos narys $\mathbb{E}_{Z\sim p_Z}[\log(1-D_\phi(G_\theta(Z)))]$ yra gerai apibrėžtas kaip Lebego integralas, ir galima apskaičiuoti jo gradientus:

$$\nabla_\theta \mathbb{E}_{Z\sim p_Z}[D_\phi(G_\theta(Z))] = \mathbb{E}_{Z\sim p_Z}\left[\nabla_x D_\phi(G_\theta(Z)) \cdot \frac{\partial G_\theta(Z)}{\partial \theta}\right]. \tag{2.19}$$

Ši lygybė, vadinama reparametrizacijos triuku (angl. *reparameterization trick*), leidžia keisti vietomis gradiento ir vidurkio operacijas, kas yra būtina GAN mokymo algoritmui.

Nors teoriškai generatorius gali turėti bet kokią Lipschitz konstantą, praktikoje naudinga, kad jis būtų Lipschitz tolydus su konstanta K_G. Tai užtikrina mokymo stabilumą: nedideli triukšmo pokyčiai $\|z_1 - z_2\|$ nesukelia drastiškų sugeneruotų duomenų pokyčių $\|G_\theta(z_1) - G_\theta(z_2)\|$. Be to, generatoriaus pajėgumas priklauso nuo tinklo architektūros. Jei triukšmo dimensija $d < L$, generatorius negali užpildyti visos $\mathbb{R}^L$ erdvės, tačiau tai netrukdo praktiškai, nes tikri finansiniai duomenys taip pat dažnai koncentruojasi žemesnės dimensijos struktūrose [4].

GAN mokymo tikslas ir vertės funkcija. Įrodėme, kad diskriminatorius D_ϕ yra matus, tolydus ir diferencijuojamas. Šios savybės leidžia formaliai apibrėžti GAN mokymo tikslą.

Pradinė GAN vertės funkcija yra

$$V(D, G) = \mathbb{E}_{X \sim p_{\text{data}}}\Big[\log D_\phi(X)\Big] + \mathbb{E}_{Z \sim p_{\text{noise}}}\Big[\log\Big(1 - D_\phi(G_\theta(Z))\Big)\Big]. \quad (2.20)$$

Svarbu pabrėžti, kad šis apibrėžimas yra prasmingas tik tada, kai logaritmai $\log D_\phi(X)$ ir $\log(1 - D_\phi(G_\theta(Z)))$ yra absoliučiai integruojami, t.y. matematiniai vidurkiai egzistuoja kaip baigtiniai Lebego integralai. Tai užtikrinama, kai diskriminatorius D_ϕ priklauso Lipschitz klasei $\mathcal{D}$ su papildomomis aprėžimų sąlygomis $\epsilon_{\min} \leq D(x) \leq 1 - \epsilon_{\min}$ (kaip bus įrodyta vėliau apie diskriminatoriaus aprėžimus). Praktikoje neuroninio tinklo sigmoid funkcija paskutiniame sluoksnyje užtikrina, kad $D_\phi(x) \in (\epsilon_{\min}, 1 - \epsilon_{\min})$ tam tikram $\epsilon_{\min} > 0$, kas garantuoja logaritmų apibrėžtumą ir absoliutų integruojamumą visuose taškuose $x \in \mathbb{R}^L$.

Mokymas vykdomas sprendžiant minimax uždavinį:

$$\min_\theta \max_\phi V(D_\phi, G_\theta).$$

Šios procedūros tikslas yra rasti optimalius parametrus θ^* ir ϕ^*, kurie realizuoja pusiausvyrą tarp generatoriaus ir diskriminatoriaus, o ne tik optimalią tikslo funkcijos reikšmę. Kitaip tariant, ieškoma tokių parametrų, kad generatorius G_{θ^*} sukurtų duomenis, kurie būtų neatskirti nuo tikrų, o diskriminatorius D_{ϕ^*} nebegalėtų jų patikimai klasifikuoti. Tai žaidimo teorijos sąvoka, kurioje dvi pusės konkuruoja: generatorius G_θ siekia minimizuoti funkcijos vertę, o diskriminatorius D_ϕ siekia ją maksimizuoti. Šis antagonistinis procesas veda prie pusiausvyros, kurioje generatorius sukuria geriausius įmanomus sintetinius duomenis pagal savo pajėgumus ir mokymosi dinamiką.

2.2 Optimalus diskriminatorius

Kai generatorius G_θ yra fiksuotas, galima rasti optimalų diskriminatorių D^*, maksimizuojantį vertės funkciją

$$V(D, G_\theta) = \mathbb{E}_{X\sim p_{\text{data}}}\Big[\log D(X)\Big] + \mathbb{E}_{Z\sim p_{\text{noise}}}\Big[\log\Big(1 - D(G_\theta(Z))\Big)\Big]. \quad (2.21)$$

Kadangi nagrinėjame absoliučiai tolydžius atsitiktinius dydžius su tankiais, vidurkiai išreiškiami integralais:

$$V(D, G) = \int_{\mathcal{X}} p_{\text{data}}(x) \log D(x)\, dx + \int_{\mathcal{X}} p_{\text{g}}(x) \log(1 - D(x))\, dx. \quad (2.22)$$

Sujungus šiuos integralus į vieną, gauname:

$$V(D, G) = \int_{\mathcal{X}} \Big[p_{\text{data}}(x) \log D(x) + p_{\text{g}}(x) \log(1 - D(x))\Big]\, dx. \quad (2.23)$$

Kadangi integruojame visoje erdvėje $\mathcal{X}$, o pointegrinė funkcija priklauso nuo $D(x)$ kiekvienam fiksuotam x nepriklausomai, maksimizuoti integralą galima maksimizuojant integrantą kiekviename taške x. Tačiau svarbu, kad tokia taškinė optimizacija duotų funkciją D^*, priklausančią leistinai funkcijų klasei $\mathcal{D}$ (apibrėžtai žemiau). Jeigu $D^* \in \mathcal{D}$, tai maksimumas kiekviename taške x garantuoja ir globalų integralos $V(D, G)$ maksimumą klasėje $\mathcal{D}$. Šis Lipschitz tolydumas bus patikrintas po optimalaus diskriminatoriaus formulės išvedimo, įrodant, kad taškinė optimizacija tikrai duoda funkciją, priklausančią klasei $\mathcal{D}$.

Diskriminatorių funkcinė klasė ir vertės funkcija Apibrėžiame diskriminatorių funkcinę klasę $\mathcal{D}$ ir įrodome, kad vertės funkcija $V(D, G)$ pasiekia maksimumą šioje klasėje.

Apibrėžimas 2.1 (Diskriminatorių klasė)**.**

$$\mathcal{D} = \Big\{D : \mathbb{R}^L \to [0, 1] \,\Big|\, D \text{ Lipschitz tolydi su konstanta } K > 0\Big\}, \quad (2.24)$$

kur $|D(x) - D(y)| \le K\|x - y\|$ visiems $x, y \in \mathbb{R}^L$.

Diskriminatorius priklauso klasei $\mathcal{D}$, nes jis yra Lipschitz tolydžių funkcijų kompozicija.

Kompaktiškumas. Klasė $\mathcal{D}$ su fiksuota konstanta K yra uždara tolygaus konvergavimo topologijoje. Pagal Arzela-Ascoli teoremą [17], funkcijų šeima $\mathcal{F}$ metrinėje erdvėje X yra kompaktiška tolygaus konvergavimo topologijoje tada ir tik tada, kai: (i) $\mathcal{F}$ tolygiai aprėžta, t.y. $\sup_{f\in\mathcal{F}, x\in X} |f(x)| < \infty$;

(ii) $\mathcal{F}$ tolygiai tolydi, t.y. bet kuriam $\varepsilon > 0$ egzistuoja $\delta > 0$ tokia, kad visiems $f \in \mathcal{F}$ ir $x, y \in X$ su $\|x - y\| < \delta$ galioja $|f(x) - f(y)| < \varepsilon$.

Patikrinsime, kad $\mathcal{D}$ tenkina abi sąlygas. *Sąlyga (i):* Pagal apibrėžimą, visiems $D \in \mathcal{D}$ ir $x \in \mathbb{R}^L$ galioja $D(x) \in [0, 1]$, todėl $\sup_{D\in\mathcal{D}, x\in\mathbb{R}^L} |D(x)| \leq 1 < \infty$. *Sąlyga (ii):* Tegu $\varepsilon > 0$. Pasirenkame $\delta = \varepsilon/K$. Tuomet bet kuriam $D \in \mathcal{D}$ ir bet kuriems $x, y \in \mathbb{R}^L$ su $\|x - y\| < \delta$ galioja

$$|D(x) - D(y)| \leq K\|x - y\| < K\delta = K \cdot \frac{\varepsilon}{K} = \varepsilon.$$

Vadinasi, $\mathcal{D}$ yra kompaktiška.

Operatoriaus tolydumas. Apibrėžiame netiesinį operatorių $\Phi : \mathcal{D} \to L^1(\mathbb{R}^L)$:

$$\Phi(D)(x) = p_{\text{data}}(x) \log D(x) + p_{\text{g}}(x) \log(1 - D(x)). \tag{2.25}$$

Šis operatorius transformuoja diskriminatoriaus funkciją $D \in \mathcal{D}$ į integruojamą funkciją $\Phi(D) : \mathbb{R}^L \to \mathbb{R}$, kurios Lebego integralas yra vertės funkcija $V(D, G) = \int_{\mathbb{R}^L} \Phi(D)(x)\, dx$. Kadangi operatorius netiesinis, jo tolydumo įrodymas reikalauja atidaus normų pasirinkimo ir įverčių su tankiais p_{data} bei p_g.

Klasei $\mathcal{D}$ taikome supremumo normą $\|\cdot\|_{\sup}$, kadangi tai Lipschitz tolydžių funkcijų aibė, kompaktiška tolygaus konvergavimo prasme. Operatoriaus Φ vaizdams taikome $L^1(\mathbb{R}^L)$ normą, nes vertės funkcija $V(D, G)$ yra Lebego integralas - integralinė funkcijos elgsena yra svarbesnė už pavienes reikšmes taškuose. Su tokia normų pora $(\|\cdot\|_{\sup}, \|\cdot\|_{L^1})$ nedideli diskriminatoriaus pakitimai supremumo prasme sukelia nedidelius integruojamų funkcijų pakitimus L^1 prasme, o tai tiesiogiai garantuoja $V(D, G)$ tolydumą.

Nors $\mathbb{R}^L$ nebūtinai yra kompaktiška erdvė, praktikoje finansinių grąžų reikšmės koncentruojasi kompaktiškoje aibėje (pvz., $[-M, M]^L$, $M > 0$), o už šios aibės tankiai p_{data} ir p_g eksponentiškai mažėja. Todėl integralai yra apibrėžti.

Lema 2.2 (Diskriminatoriaus aprėžimai)**.** *Tegu $D \in \mathcal{D}$ yra neuroninis tinklas su sigmoid paskutiniame sluoksnyje. Tuomet egzistuoja konstantos $0 < \epsilon_{\min} < 1 - \epsilon_{\min} < 1$ tokios, kad*

$$\epsilon_{\min} \leq D(x) \leq 1 - \epsilon_{\min}, \quad \textit{visiems } x \in \mathbb{R}^L. \tag{2.26}$$

Įrodymas. Diskriminatorius $D(x) = \sigma(h(x))$, kur $\sigma(t) = (1+e^{-t})^{-1}$. Kadangi neuroninis tinklas h Lipschitz tolydus su aprėžtais svoriais, egzistuoja $M > 0$ tokia, kad $|h(x)| \leq M$ visiems $x \in \mathbb{R}^L$. Vadinasi,

$$\sigma(-M) \leq D(x) \leq \sigma(M).$$

Žymėdami $\epsilon_{\min} = \sigma(-M) = \frac{1}{1+e^M} > 0$ ir pastebėję, kad $\sigma(M) = 1 - \epsilon_{\min}$, gauname reikiamus apibrėžimus. $\square$

Kadangi $D(x) \geq \epsilon_{\min} > 0$ ir $D(x) \leq 1 - \epsilon_{\min} < 1$ visiems $x \in \mathbb{R}^L$, logaritmai $\log D(x)$ ir $\log(1 - D(x))$ yra gerai apibrėžti ir absoliučiai integruojami su tankiais p_{data} ir p_g. Vadinasi, vertės funkcija $V(D, G)$ egzistuoja visiems $D \in \mathcal{D}$. Be to, funkcijų klasė su tokiais aprėžimais lieka uždara, todėl kompaktiškumas išlieka.

Teorema 2.3 (Operatoriaus Φ tolydumas)**.** *Operatorius $\Phi : \mathcal{D} \to L^1(\mathbb{R}^L)$ yra tolydus, kai $\mathcal{D}$ erdvėje naudojama supremumo norma $\|\cdot\|_{\sup}$, o $L^1(\mathbb{R}^L)$ erdvėje - L^1 norma.*

Įrodymas. Tegu $D_1, D_2 \in \mathcal{D}$. Įvertinsime $\|\Phi(D_1) - \Phi(D_2)\|_{L^1}$:

$$\begin{aligned}
&\|\Phi(D_1) - \Phi(D_2)\|_{L^1} \\
&= \int_{\mathbb{R}^L} \Big| p_{\text{data}}(x)[\log D_1(x) - \log D_2(x)] \\
&\quad + p_{\text{g}}(x)[\log(1 - D_1(x)) - \log(1 - D_2(x))] \Big| \, dx.
\end{aligned}$$

Pagal trikampio nelygybę (Minkowski nelygybės L^1 atveju):

$$\begin{aligned}
&\leq \int_{\mathbb{R}^L} p_{\text{data}}(x) |\log D_1(x) - \log D_2(x)| \, dx \\
&\quad + \int_{\mathbb{R}^L} p_{\text{g}}(x) |\log(1 - D_1(x)) - \log(1 - D_2(x))| \, dx.
\end{aligned}$$

Įvertinsime pirmąjį integralą. Logaritmo vidutinės reikšmės teorema: $|\log a - \log b| = \frac{|a-b|}{\xi}$, kur $\xi \in [\min(a, b), \max(a, b)]$. Kadangi $D_1(x), D_2(x) \geq \epsilon_{\min}$ visiems x (pagal Lemą), tai $\xi \geq \epsilon_{\min}$. Čia naudojame įprastą supremumą, nes funkcijos visur apibrėžtos:

$$|\log D_1(x) - \log D_2(x)| \leq \frac{|D_1(x) - D_2(x)|}{\epsilon_{\min}} \leq \frac{\|D_1 - D_2\|_{\sup}}{\epsilon_{\min}}.$$

Vadinasi,

$$\int_{\mathbb{R}^L} p_{\text{data}}(x) |\log D_1(x) - \log D_2(x)| \, dx \leq \frac{\|D_1 - D_2\|_{\sup}}{\epsilon_{\min}} \underbrace{\int_{\mathbb{R}^L} p_{\text{data}}(x) \, dx}_{=1}.$$

Analogiškai, kadangi $1 - D_i(x) \geq \epsilon_{\min}$, gauname

$$\int_{\mathbb{R}^L} p_{\text{g}}(x) |\log(1 - D_1(x)) - \log(1 - D_2(x))| \, dx \leq \frac{\|D_1 - D_2\|_{\sup}}{\epsilon_{\min}}.$$

Sujungę, gauname:

$$\|\Phi(D_1) - \Phi(D_2)\|_{L^1} \leq \frac{2}{\epsilon_{\min}} \|D_1 - D_2\|_{\sup}.$$

Vadinasi, jei $\|D_1 - D_2\|_{\sup} < \delta$, tai $\|\Phi(D_1) - \Phi(D_2)\|_{L^1} < \varepsilon$ su $\delta = \frac{\epsilon_{\min}\varepsilon}{2}$, kas įrodo tolydumą. □

Taigi parodėme, kad $\mathcal{D}$ yra kompaktiška aibė pagal Arzela-Ascoli teoremą, operatorius $\Phi : \mathcal{D} \to L^1$ yra tolydus su normų pora $(\|\cdot\|_{\sup}, \|\cdot\|_{L^1})$, o diskriminatoriaus reikšmės kontroliuojamos sąlyga $D(x) \geq \epsilon_{\min} > 0$ garantuoja logaritmų egzistavimą ir absoliutų integruojamumą. Kadangi $\mathcal{D}$ kompaktiška ir Φ tolydus, integralas $V(D, G) = \int \Phi(D)(x)\, dx$ yra tolydus kompaktinėje aibėje. Pagal Weierstrass teoremą, egzistuoja $D^* \in \mathcal{D}$, maksimizuojantis $V(D, G)$.

Optimalaus diskriminatoriaus formulė Nustatę, kad optimalus diskriminatorius $D^* \in \mathcal{D}$ egzistuoja, išvesime jo eksplicitinę formulę.

Kiekvienam $x \in \mathcal{X}$ diskriminatorius $D^*(x)$ maksimizuoja išraišką:

$$f(D(x)) = p_{\text{data}}(x) \log D(x) + p_{\text{g}}(x) \log(1 - D(x)). \tag{2.27}$$

Taškiniam maksimumui rasti diferencijuojame $f(D(x))$ argumento $D(x)$ atžvilgiu ir prilyginame nuliui:

$$\frac{\partial f}{\partial D(x)} = \frac{p_{\text{data}}(x)}{D(x)} - \frac{p_{\text{g}}(x)}{1 - D(x)} = 0. \tag{2.28}$$

Išsprendę šią lygtį, gauname:

$$p_{\text{data}}(x)(1 - D(x)) = p_{\text{g}}(x) D(x), \tag{2.29}$$

iš kur

$$p_{\text{data}}(x) = D(x)\Big(p_{\text{data}}(x) + p_{\text{g}}(x)\Big). \tag{2.30}$$

Antroji išvestinė

$$\frac{\partial^2 f}{\partial (D(x))^2} = -\frac{p_{\text{data}}(x)}{D(x)^2} - \frac{p_{\text{g}}(x)}{(1 - D(x))^2} < 0 \tag{2.31}$$

patvirtina, kad tai yra maksimumas, o ne minimumas.

Taigi optimalus diskriminatorius yra:

$$D^*(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_{\text{g}}(x)}. \tag{2.32}$$

Dabar įrodysime, kad funkcija $D^*(x)$ tikrai priklauso Lipschitz tolydžių funkcijų klasei $\mathcal{D}$. Tarkime, kad finansinių grąžų tankiai p_{data} ir p_{g} yra diferencijuojami su apibrėžtomis išvestinėmis kompaktiškoje aibėje $[-M, M]^L$. Žymėkime $s(x) = p_{\text{data}}(x) + p_{\text{g}}(x)$ sumą ir tarkime, kad egzistuoja konstantos $C_p > 0$ ir $s_{\min} > 0$ tokios, kad

$$\|\nabla_x p_{\text{data}}(x)\| \leq C_p, \quad \|\nabla_x p_{\text{g}}(x)\| \leq C_p, \quad s(x) \geq s_{\min}$$

visiems $x \in [-M, M]^L$.

Apskaičiuokime gradientą $\nabla_x D^*(x)$ naudojant dalinių trupmenų taisyklę:

$$\begin{aligned}\nabla_x D^*(x) &= \nabla_x \left(\frac{p_{\text{data}}(x)}{s(x)} \right) \\ &= \frac{\nabla_x p_{\text{data}}(x) \cdot s(x) - p_{\text{data}}(x) \cdot \nabla_x s(x)}{s(x)^2}.\end{aligned}$$

Kadangi $\nabla_x s(x) = \nabla_x p_{\text{data}}(x) + \nabla_x p_{\text{g}}(x)$, gauname:

$$\begin{aligned}\|\nabla_x D^*(x)\| &= \left\| \frac{\nabla_x p_{\text{data}}(x) \cdot s(x) - p_{\text{data}}(x) \cdot (\nabla_x p_{\text{data}}(x) + \nabla_x p_{\text{g}}(x))}{s(x)^2} \right\| \\ &\leq \frac{\|\nabla_x p_{\text{data}}(x)\| \cdot s(x) + p_{\text{data}}(x) \cdot (\|\nabla_x p_{\text{data}}(x)\| + \|\nabla_x p_{\text{g}}(x)\|)}{s(x)^2} \\ &\leq \frac{C_p \cdot s(x) + 1 \cdot (C_p + C_p)}{s_{\min}^2} = \frac{C_p \cdot s(x) + 2C_p}{s_{\min}^2}.\end{aligned}$$

Kadangi $s(x) \leq 2$ (nes p_{data} ir p_{g} yra tikimybių tankiai), gauname

$$\|\nabla_x D^*(x)\| \leq \frac{C_p \cdot 2 + 2C_p}{s_{\min}^2} = \frac{4C_p}{s_{\min}^2} \equiv K.$$

Pagal vidutinės reikšmės teoremą daugiamačiame atvejui, bet kuriems $x, y \in [-M, M]^L$ galioja

$$|D^*(x) - D^*(y)| \leq \sup_{z \in [x,y]} \|\nabla_x D^*(z)\| \cdot \|x - y\| \leq K\|x - y\|.$$

Vadinasi, D^* yra Lipschitz tolydi su konstanta $K = \frac{4C_p}{s_{\min}^2}$ kompaktiškoje aibėje, kur koncentruojasi duomenų tankiai, todėl $D^* \in \mathcal{D}$.

Kadangi $D^* \in \mathcal{D}$ ir tenkina apibrėžimus $\epsilon_{\min} \leq D^*(x) \leq 1 - \epsilon_{\min}$ (kaip parodyta Lemoje), logaritmai yra gerai apibrėžti ir funkcija $f(D^*(x))$ yra Lebego integruojama.

Įstatę D^* į vertės funkciją, galime gauti generatoriaus kriterijų, išreikštą per Jensen-Shannon (JS) divergenciją. Pirmiausia, pateikiame Kullback-Leibler (KL) divergencijos apibrėžimą. Dviejų pasiskirstymų, kurių tankiai yra p ir q KL divergencija apibrėžiama taip:

$$\mathrm{KL}(p\|q) = \int_{\mathcal{X}} p(x) \log \frac{p(x)}{q(x)}\, dx. \tag{2.33}$$

KL divergencija yra nesimetrinė, t.y. $\mathrm{KL}(p\|q) \neq \mathrm{KL}(q\|p)$. Jensen-Shannon divergencija yra simetrinis variantas, apibrėžtas kaip

$$\mathrm{JS}(p\|q) = \frac{1}{2}\mathrm{KL}\Big(p \,\Big\|\, \frac{p+q}{2}\Big) + \frac{1}{2}\mathrm{KL}\Big(q \,\Big\|\, \frac{p+q}{2}\Big). \tag{2.34}$$

JS divergencija matuoja atstumą tarp dviejų pasiskirstymų ir turi savybę $\mathrm{JS}(p\|q) = \mathrm{JS}(q\|p) \geq 0$, lygybė pasiekiama tada ir tik tada, kai $p = q$ [22].

Dabar įstatome optimalų diskriminatorių $D^*(x) = \frac{p_{\mathrm{data}}(x)}{p_{\mathrm{data}}(x)+p_{\mathrm{g}}(x)}$ į vertės funkciją:

$$\begin{aligned} V(D^*, G) &= \int_{\mathcal{X}} p_{\mathrm{data}}(x) \log D^*(x)\, dx + \int_{\mathcal{X}} p_{\mathrm{g}}(x) \log(1 - D^*(x))\, dx \\ &= \int_{\mathcal{X}} p_{\mathrm{data}}(x) \log \frac{p_{\mathrm{data}}(x)}{p_{\mathrm{data}}(x) + p_{\mathrm{g}}(x)}\, dx \\ &\quad + \int_{\mathcal{X}} p_{\mathrm{g}}(x) \log \frac{p_{\mathrm{g}}(x)}{p_{\mathrm{data}}(x) + p_{\mathrm{g}}(x)}\, dx. \end{aligned} \tag{2.35}$$

Žymėdami $m(x) = \frac{p_{\mathrm{data}}(x)+p_{\mathrm{g}}(x)}{2}$ (dviejų tankių vidurkį), galime perrašyti:

$$\begin{aligned} V(D^*, G) &= \int_{\mathcal{X}} p_{\mathrm{data}}(x) \log \frac{p_{\mathrm{data}}(x)}{2m(x)}\, dx \\ &\quad + \int_{\mathcal{X}} p_{\mathrm{g}}(x) \log \frac{p_{\mathrm{g}}(x)}{2m(x)}\, dx \\ &= \int_{\mathcal{X}} p_{\mathrm{data}}(x) \Big[\log \frac{p_{\mathrm{data}}(x)}{m(x)} - \log 2\Big]\, dx \\ &\quad + \int_{\mathcal{X}} p_{\mathrm{g}}(x) \Big[\log \frac{p_{\mathrm{g}}(x)}{m(x)} - \log 2\Big]\, dx \\ &= \mathrm{KL}(p_{\mathrm{data}}\|m) - \log 2 + \mathrm{KL}(p_{\mathrm{g}}\|m) - \log 2 \\ &= \mathrm{KL}(p_{\mathrm{data}}\|m) + \mathrm{KL}(p_{\mathrm{g}}\|m) - 2\log 2. \end{aligned} \tag{2.36}$$

Pagal JS divergencijos apibrėžimą

$$\mathrm{JS}(p\|q) = \frac{1}{2}\mathrm{KL}(p\|m) + \frac{1}{2}\mathrm{KL}(q\|m),$$

kur $m = \frac{p+q}{2}$, gauname generatoriaus funkciją:

$$C(G) = V(D^*, G) = -\log 4 + 2\,\mathrm{JS}\Big(p_{\mathrm{data}} \,\|\, p_{\mathrm{g}}\Big), \tag{2.37}$$

kur $-\log 4 = -2\log 2$ yra konstanta.

Jei diskriminatorius mokymas veikia gerai, sugeneruoti duomenys vis labiau panašėja į tikrus duomenis. Ši formulė rodo, kad optimalus rezultatas pasiekiamas tada, kai sugeneruotų duomenų pasiskirstymas visiškai sutampa su tikrų duomenų pasiskirstymu. Tuomet generatorius sukuria duomenis, kurių diskriminatorius nebegali atskirti nuo tikrų [9].

Alternatyvi generatoriaus nuostolių funkcija. Praktikoje pradinė minimax formulė dažnai sukelia silpnus gradientus generatoriui, ypač kai diskriminatorius tampa labai geras. Konkrečiau, kai generatorius kuria prastos kokybės duomenis, diskriminatorius lengvai juos atpažįsta: $D(G_\theta(z)) \approx 0$. Tokiu atveju generatoriaus nuostoliai $\mathcal{L}_G = \mathbb{E}[\log(1 - D(G_\theta(Z)))]$ tampa artimi konstantai $\log(1) = 0$, o gradientas

$$\frac{\partial \mathcal{L}_G}{\partial \theta} = -\mathbb{E}\Big[\frac{1}{1 - D(G_\theta(Z))} \cdot D'(G_\theta(Z)) \cdot \frac{\partial G_\theta(Z)}{\partial \theta}\Big] \tag{2.38}$$

tampa labai mažas moduliu ($|\nabla_\theta \mathcal{L}_G| \ll 1$), todėl parametrų atnaujinimai yra nereikšmingi ir generatorius nebesimoko. Todėl dažnai naudojama alternatyvi generatoriaus nuostolių funkcija, vadinama nesočiuoju nuostoliu. Ji apibrėžiama taip:

$$\mathcal{L}_G^{\mathrm{NS}}(\theta) = -\mathbb{E}_{Z \sim p_Z}\Big[\log D_\phi\Big(G_\theta(Z)\Big)\Big], \tag{2.39}$$

Ši funkcija padeda generatoriui gauti stipresnius gradientus mokymo pradžioje ir greičiau mokytis. Pradinė minimax formulė siekia, kad diskriminatorius nesugebėtų atpažinti sugeneruotų duomenų (minimizuoja $\log(1 - D(G(z)))$), tuo tarpu nesočiųjų nuostolių funkcija tiesiogiai siekia, kad diskriminatorius manytų, jog sugeneruoti duomenys yra tikri (maksimizuoja $\log D(G(z))$). Antrasis požiūris duoda stipresnį mokymo signalą, kai generatorius dar mokosi. Kitaip, jei diskriminatorius išmoksta atskirti tikrus ir sugeneruotus duomenis per greitai, generatorius nebegali pasivyti ir mokymas sustoja.

2.3 GAN finansinėms laiko eilutėms

GAN taikymas finansinėms laiko eilutėms reikalauja atsižvelgti į laiko struktūrą, duomenų pastovumą ir finansams būdingas savybes, tokias kaip didelės svyravimo uodegos, nepastovumo grupavimasis ir sverto efektas. Dvi įprastos modeliavimo strategijos yra:

1. Visos sekos generatoriai ir diskriminatoriai: generatorius sukuria grąžų seką tam tikro ilgio L, pavyzdžiui,

$$G_\theta : \mathbb{R}^d \to \mathbb{R}^L, \qquad r_{1:L} = G_\theta(u), \tag{2.40}$$

kur u yra atsitiktinis įvesties vektorius. Diskriminatorius vertina visą seką ir gali naudoti konvoliucinius ar rekurentinius sluoksnius, kad aptiktų tiek trumpalaikes, tiek ilgalaikes priklausomybes.

2. Autoregresinis arba sąlyginis generavimas: generatorius kuria seką žingsnis po žingsnio, remdamasis ankstesniais stebėjimais. Tegu $x_{1:t}$ žymi stebėtą istoriją; generatorius sukuria prognozę

$$G_\theta(x_{1:t}) \to \hat{x}_{t+1:t+H}, \tag{2.41}$$

kur $\hat{x}_{t+1:t+H}$ žymi būsimos sekos įvertinimą, naudojamą scenarijų generavimui ir prognozavimui.

Siekiant įveikti specifinius iššūkius, susijusius su laikinių duomenų generavimu, pasiūlyta įvairių GAN variantų (pvz. TimeGAN, WGAN, RCGAN ir kiti), kurie praturtina bazinę architektūrą papildomomis struktūromis ar mokymo tikslais. Šios modifikacijos sprendžia tokias problemas kaip ilgalaikių priklausomybių išsaugojimą, vieno žingsnio prognozavimo tikslumą ir mokymo stabilumas sudėtingose laiko eilutėse [23, 24].

GAN modeliai, taikomi finansinėms laiko eilutėms, siekia išmokti bendrą sekų dėsningumą ir generuoti duomenis, kurie būtų panašūs į tikrus finansinius duomenis. Vertinant tokių modelių kokybę, svarbu atkreipti dėmesį, ar sugeneruotos sekos atspindi finansams būdingas savybes, tokias kaip didelės svyravimo uodegos, nepastovumo grupavimasis ir tarpusavio priklausomybė tarp skirtingų aktyvų [25].

2.4 Praktiniai aspektai

GAN mokymas finansiniais duomenimis susiduria su keliomis specifinėmis problemomis. Pirmiausia, mažos imtys ir nestacionarumas kelia iššūkių, nes finansinės laiko eilutės dažnai yra trumpos ir patiria struktūrinius lūžius, nes nėra pastovios. Dėl to GAN modeliams gali trūkti pakankamai duomenų, kad jie išmoktų bendrus dėsningumus, o staigūs pokyčiai duomenyse apsunkina mokymą ir sumažina generuojamų sekų kokybę.

Kita svarbi problema yra modų kolapsavimas, kai generatorius pradeda kurti mažo įvairumo sekas, kurios vis tiek apgauna diskriminatorių. Tokiu

atveju sugeneruoti duomenys tampa pernelyg panašūs tarpusavyje ir neatspindi tikros duomenų įvairovės, kas ypač nepageidautina finansų srityje, kur svarbu modeliuoti įvairius rinkos scenarijus.

Silpni arba per stiprūs gradientai yra dar viena dažna problema, kylanti dėl sudėtingų sąveikų tarp generatoriaus ir diskriminatoriaus. Dėl to mokymas gali tapti nestabilus, o modelis gali nebesimokyti arba mokytis labai prastai. Tai ypač pasireiškia, kai naudojamos gilios neuroninės architektūros ar sudėtingos sekų struktūros.

Galiausiai, statistinis pagrįstumas yra būtinas norint įvertinti sugeneruotų imčių kokybę. Finansinių duomenų generavime svarbu, kad sintetinės sekos atkartotų ekonomiškai aktualias savybes, todėl jų vertinimui taikomi statistiniai testai ir analizės metodai. Taip galima užtikrinti, kad GAN modeliai generuoja realistiškus ir praktiškai naudingus duomenis.

Apibendrinant, GAN modelis yra įrankis finansinių laiko eilučių generavimui, nes leidžia kurti sintetinius duomenis be prielaidų apie jų pasiskirstymą. GAN teorija remiasi priešišku mokymu, kur generatorius ir diskriminatorius mokosi kartu: generatorius stengiasi sukurti kuo panašesnius į tikrus duomenis, o diskriminatorius bando juos atskirti. Mokymo metu abu tinklai tobulėja, iki kol sugeneruoti duomenys tampa sunkiai atskiriami nuo tikrų. Praktikoje dažnai naudojamos įvairios nuostolių funkcijos ir architektūriniai patobulinimai, kad mokymas būtų stabilesnis ir efektyvesnis, pavyzdžiui, nesotinantis nuostolis ar WGAN metodas, kurie bus pristatyti kituose skyriuose. Finansų srityje labai svarbu, kad modeliai atsižvelgtų į laiko priklausomybę ir išlaikytų finansams būdingas savybes, tokias kaip didelės svyravimo uodegos ar nepastovumo grupavimasis. Vertinant sugeneruotus duomenis, reikia derinti mašininio mokymosi metrikas su statistiniais testais ir rizikos vertinimo priemonėmis, kad būtų užtikrinta jų kokybė ir praktinis naudingumas [26].

3 skyrius

GAN ir WGAN palyginimas

3.1 Ko pradiniai GAN nepasiekia praktikoje

Pradinė GAN formulė (2.20) siekia, kad modelio sugeneruoti duomenys būtų kuo panašesni į tikrus, naudojant diskriminatorių, kuris atskiria tikrus duomenis nuo sintetinių [9]. Nors toks požiūris teoriškai atrodo patrauklus, praktikoje kyla dvi pagrindinės problemos, ypač ryškios finansinėse laiko eilutėse:

- **Silpni arba išnykstantys gradientai, kai duomenų pasiskirstymai nesutampa.** Kai generatoriaus ir tikrų duomenų pasiskirstymai yra labai skirtingi arba nesikerta, modelis gauna labai silpną mokymo signalą. Tai dažna pradžioje arba kai duomenys sudėtingi, pavyzdžiui, finansinių sekų atvejais.
- **Nuostolių nekoreliavimas su imties kokybe ir nestabilus mokymas.** Diskriminatoriaus nuostoliai ne visada rodo, ar generuojami duomenys gerėja. Dėl to mokymas gali būti nestabilus, sunku stebėti progresą, o modelis dažnai „užstringa“ generuodamas tik kelis duomenų tipus.

Finansinėms sekoms šios patologijos pasireiškia kaip generatoriai, kurie nepakankamai atkartoja kraštutinius įvykius, silpnina nepastovumo grupavimą arba kolapsuoja į keletą modų, kurie vis tiek apgauna gerai išmokytą diskriminatorių. Tokiose situacijose naudingas mokymo signalas, kuris kinta sklandžiai su atstumu tarp realių ir sintetinių pasiskirstymų.

3.2 Wasserstein-1 atstumas ir diskriminatorius

Wasserstein GAN (WGAN) vietoje to naudoja Wasserstein-1 atstumą, kuris užtikrina sklandesnį bei informatyvesnį mokymo signalą net tada, kai pasiskirstymai nesutampa [13].

Wasserstein atstumas kiekybiškai įvertina minimalų poslinkio dydį, reikalingą norint suderinti vieną tikimybių skirstinį su kitu. Jis matuoja, kiek vidutiniškai vieno skirstinio elementai turi būti perkelti pagrindinėje metrinėje erdvėje, kad atitiktų antrojį skirstinį. Formaliai, tai užrašoma kaip optimalaus transportavimo planas: jei turime du tikimybių skirstinius P ir Q, ieškoma tokio junginio skirstinio $\gamma \in \Gamma(P, Q)$ (kur $\Gamma(P, Q)$ žymi visus junginius skirstinius su marginaliaisiais pasiskirstymais P ir Q), kuris minimizuoja vidutinius poslinkius $\mathbb{E}_{(x,y)\sim\gamma}[\|X - Y\|]$. Mažesnis atstumas rodo, kad skirstiniai yra erdviškai artimesni arba struktūriškai panašesni savo vietos ir sklaidos požiūriu. Formaliajame apibrėžime, Wasserstein-1 atstumas tarp dviejų tikimybių skirstinių P ir Q yra

$$W_1(P, Q) = \inf_{\gamma\in\Gamma(P,Q)} \mathbb{E}_{(X,Y)\sim\gamma}[\|X - Y\|], \tag{3.1}$$

kur $\Gamma(P, Q)$ yra visų jungtinių skirstinių su marginaliaisiais pasiskirstymais P ir Q aibė. Šis apibrėžimas interpretuojamas kaip optimalaus transportavimo problema: rasti "pigiausią" būdą perkelti masę iš skirstinio P į skirstinį Q, kur kaina yra atstumas tarp taškų. Tiesioginė optimizacija pagal Wasserstein-1 atstumo apibrėžimą yra sunkiai įgyvendinama, nes reikėtų perrinkti visus galimus junginių skirstinius $\gamma \in \Gamma(P, Q)$. WGAN šią problemą išsprendžia naudodamas Kantorovich-Rubinstein dualumo teoremą, kuri pateikia ekvivalentų, bet praktiškesnį formulavimą. Pagal Kantorovich-Rubinstein dualumą, Wasserstein-1 atstumas gali būti užrašytas kaip

Teorema 3.1 (Kantorovich-Rubinstein dualumas)**.** *Jei P ir Q yra tikimybių skirstiniai metrinėje erdvėje $(\mathcal{X}, d)$, tai*

$$W_1(P, Q) = \sup_{\|f\|_{\mathrm{Lip}}\leq 1} \Big(\mathbb{E}_{X\sim P}[f(X)] - \mathbb{E}_{X\sim Q}[f(X)]\Big), \tag{3.2}$$

kur supremumas ieškomas tarp visų 1-Lipschitz funkcijų $f : \mathcal{X} \to \mathbb{R}$.

WGAN kontekste tikimybiniai matai P ir Q apibrėžiami atitinkamai tikrųjų duomenų tankiu p_{data} ir generatoriaus tankiu p_{g}, kur bet kuriai Borelo aibei A galioja $P(A) = \int_A p_{\mathrm{data}}(x)\,dx$ ir $Q(A) = \int_A p_{\mathrm{g}}(x)\,dx$. Funkcija f vadinama L-Lipschitz tolydžiąja, jei visiems x, y jos apibrėžties srityje galioja

$$|f(x) - f(y)| \leq L\|x - y\|, \tag{3.3}$$

t.y. funkcijos reikšmės pokytis negali viršyti L kartų atstumo tarp jos argumentų. WGAN kontekste 1-Lipschitz apribojimas užtikrina, kad diskriminatorius duoda stabilų mokymo signalą netgi tada, kai generuojami ir tikri duomenys labai skiriasi, kas yra esminė Wasserstein atstumo aproksimavimo sąlyga ir prisideda prie mokymo stabilumo [27]. WGAN parametrizuoja f neuroniniu kritiku, kuris įvertina, kiek sugeneruotas rezultatas yra panašus į tikrą f_w ir sprendžia

$$\min_{\theta} \max_{w} \Big(\mathbb{E}_{X \sim p_{\text{data}}}[f_w(X)] - \mathbb{E}_{Z \sim p_Z}[f_w(G_\theta(Z))] \Big), \tag{3.4}$$

kur w yra diskriminatoriaus parametrai, apriboti Lipschitz sąlyga, θ yra generatoriaus parametrai, Z yra atsitiktinis triukšmas iš pradinio pasiskirstymo p_Z, o G_θ yra generatorius. Iš to seka dvi pagrindinės pasekmės:

1. Diskriminatoriaus nuostoliai rodo, kaip gerai generatorius imituoja tikrus duomenis; kuo mažesnis nuostolis, tuo geresnė kokybė.

2. Generatorius gauna aiškesnius mokymo signalus net tada, kai sugeneruoti ir tikri duomenys labai skiriasi, todėl mokymas tampa stabilesnis ir sumažėja modų kolapsas.

Lipschitz tolydumo užtikrinimas. Ankstyvuose WGAN buvo naudojamas svorių apkarpymas, kai diskriminatoriaus parametrai po kiekvieno atnaujinimo yra priverčiami likti kompaktiškame intervale, tačiau tai sukelia diskriminatoriaus pajėgumo nepanaudojimą. Plačiai pritaikytas sprendimas yra *gradiento bausmė* [15], kuri papildo diskriminatoriaus tikslo funkciją

$$\lambda \, \mathbb{E}_{\hat{x}} \Big(\big\| \nabla_{\hat{x}} f_w(\hat{x}) \big\|_2 - 1 \Big)^2, \tag{3.5}$$

kur $\hat{x}$ yra interpoliuoti taškai tarp tikrų ir sugeneruotų duomenų: $\hat{x} = \epsilon x_{\text{real}} + (1-\epsilon) x_{\text{fake}}$ su $\epsilon \sim U[0,1]$, o λ yra bausmės koeficientas.

Šios bausmės tikslas yra užtikrinti, kad diskriminatoriaus gradientas būtų apribotas visoje įvesties erdvėje. Bausmė skatina gradiento normą išlaikyti artimą vienetui interpoliuotų taškų atžvilgiu, kas užtikrina 1-Lipschitz savybę.

Lipschitz konstanta K reiškia, jog funkcijos pokytis tarp bet kurių dviejų taškų negali viršyti K kartų atstumo tarp tų taškų:

$$|f(x_1) - f(x_2)| \leq K \|x_1 - x_2\|. \tag{3.6}$$

Siekiama užtikrinti $K = 1$. Funkcijos pokytis tarp bet kurių dviejų taškų x_1 ir x_2 gali būti išreikštas integruojant palei tiesę juos jungiančią:

$$f(x_2) - f(x_1) = \int_0^1 \nabla f(x_1 + t(x_2 - x_1)) \cdot (x_2 - x_1) \, dt. \tag{3.7}$$

Šio integralo modulis įvertinamas naudojant Cauchy-Schwarz nelygybę:

$$\begin{aligned} |f(x_2) - f(x_1)| &= \left| \int_0^1 \nabla f(x_1 + t(x_2 - x_1)) \cdot (x_2 - x_1)\, dt \right| \\ &\leq \int_0^1 \left\| \nabla f(x_1 + t(x_2 - x_1)) \right\| \cdot \|x_2 - x_1\|\, dt \\ &= \|x_2 - x_1\| \int_0^1 \left\| \nabla f(x_1 + t(x_2 - x_1)) \right\| dt. \end{aligned} \tag{3.8}$$

Jei užtikrinama, kad $\|\nabla f(x)\| \leq 1$ visur, tai

$$|f(x_2) - f(x_1)| \leq \|x_2 - x_1\| \int_0^1 1\, dt = \|x_2 - x_1\|, \tag{3.9}$$

kas ir atitinka 1-Lipschitz sąlygą. Gradiento bausmės formulė, kuri baudžia per didelius gradientus, verčia $\|\nabla f(\hat{x})\|$ būti artima 1, o tai užtikrina stabilų ir informatyvų mokymo signalą generatoriui netgi tada, kai sugeneruoti duomenys dar labai skiriasi nuo tikrų.

Kitas būdas yra spektrinė normalizacija, kuri tiesiogiai riboja svorių matricų poveikį. Neuroniniame tinkle kiekvienas sluoksnis transformuoja įvestį per afininę transformaciją $y = Wx + b$, kur $W \in \mathbb{R}^{m \times n}$ yra svorių matrica, $x \in \mathbb{R}^n$ - įvestis, o $b \in \mathbb{R}^m$ - poslinkio vektorius. Kiekviena svorių matrica W padalijama iš jos spektrinės normos $\sigma(W)$:

$$\bar{W} = \frac{W}{\sigma(W)}, \tag{3.10}$$

kur $\sigma(W) = \max_{\|h\|=1} \|Wh\|$ yra didžiausia matricos tikrinė reikšmė. Ši reikšmė parodo, kaip stipriai matrica W gali padidinti vektoriaus dydį. Taikant šią normalizaciją visiems diskriminatoriaus sluoksniams, gaunama, kad kiekvienas tiesinis operatorius yra 1-Lipschitz, nes

$$\|\bar{W}x_1 - \bar{W}x_2\| = \|\bar{W}(x_1 - x_2)\| \leq \sigma(\bar{W})\|x_1 - x_2\| = \|x_1 - x_2\|. \tag{3.11}$$

Sudėjus kelis 1-Lipschitz sluoksnius su 1-Lipschitz aktyvavimo funkcijomis (pvz., ReLU, LeakyReLU), gauname, kad visas diskriminatorius f_w taip pat yra 1-Lipschitz. Tai užtikrina stabilų mokymą be papildomų bausmės narių, skirtingai nuo gradiento bausmės metodo [28].

Mokymo dinamika. Praktikoje mokymas keičia kelis diskriminatoriaus atnaujinimus per vieną generatoriaus žingsnį, su atskirais mokymosi tempais. Diskriminatoriaus mokymo tikslas yra maksimizuoti

$$\mathbb{E}_{x \sim p_r}[f_w(x)] - \mathbb{E}_{\tilde{x} \sim p_g}[f_w(\tilde{x})] - \lambda\, \mathbb{E}_{\hat{x}} \Big(\left\| \nabla_{\hat{x}} f_w(\hat{x}) \right\|_2 - 1 \Big)^2, \tag{3.12}$$

kur pirmieji du nariai apskaičiuoja Wasserstein atstumą, o trečiasis narys yra gradiento bausmė. Po to generatorius atnaujinamas, kad sumažintų įvertintą Wasserstein atstumą. Skirtingai nuo pradinio GAN tikslo, diskriminatoriaus nuostoliai paprastai nuosekliai mažėja, kai imties kokybė gerėja, sugeneruoti duomenys tampa vis panašesni į tikrus, tinklas mokosi juos generuoti vis tiksliau.

3.3 WGAN taikymas finansinėms laiko eilutėms

Diskriminatorius ir generatorius veikia su *sekomis*, o ne su nepriklausomas ir identiškai paskirstytais stebėjimais. Dažniausiai naudojami du mokymo būdai:

- **Visos sekos modeliavimas:** Generatorius sukuria visą grąžų seką iš karto, o diskriminatorius įvertina visą seką, naudodamas parinktus tinklus (pvz., RNN ar CNN).
- **Sekos po vieną žingsnį:** Generatorius prognozuoja žingsnį po žingsnio pagal ankstesnius, o diskriminatorius lygina tikrus ir sugeneruotus žingsnius po vieną.

Tokie architektūriniai sprendimai leidžia tinklui atkartoti laiko priklausomybes finansiniuose duomenyse [29].

Nors WGAN suteikia stabilesnį mokymo signalą negu GAN, finansinių laiko eilučių specifika reikalauja papildomų priemonių. Literatūroje siūlomi keli būdai, kaip pritaikyti WGAN šioms savybėms atkartoti [30]:

1. **Laiko struktūros tinklai:** Naudojami rekurentiniai tinklai (pvz., GRU ar LSTM) generatoriuje ir diskriminatoriuje, kad modelis išmoktų ilgalaikę atmintį ir nepastovumo dinamiką.
2. **Papildomi apribojimai:** Pridedami reguliavimo nariai į tikslo funkciją, kurie skatina išlaikyti empirinius momentus (vidurkį, dispersiją) ar autokoreliacijos struktūrą.
3. **Rizikos priemonių stebėjimas:** Mokymo metu arba po jo tikrinamos rizikos charakteristikos—VaR, CVaR, uodegų indeksai—kad būtų užtikrinta, jog sugeneruoti duomenys atitinka tikrų duomenų rizikos profilį.

Šie papildymai nėra standartinės WGAN dalys, bet jie padeda geriau pritaikyti modelį finansinių duomenų specifikai.

3.4 Praktiniai aspektai

WGAN sprenžia pagrindines pradinio GAN problemas: suteikia stabilesnį mokymo signalą, aiškesnius gradientus ir sumažina modų kolapsavimą [31, 1]. Wasserstein atstumas veikia net tada, kai sugeneruoti ir tikri duomenys labai skiriasi, o diskriminatoriaus nuostoliai geriau atspindi imties kokybę nei pradinio GAN diskriminatoriaus nuostoliai.

Tačiau WGAN nėra universalus sprendimas. Finansinėms laiko eilutėms svarbu ne tik stabilus mokymas, bet ir tai, ar sugeneruoti duomenys išsaugo būdingas savybes: nepastovumo grupavimąsi, sunkias uodegas, laiko priklausomybes ir rizikos profilį. Literatūroje daug eksperimentuojama su įvairiomis architektūromis, sąlygojimo schemomis ir papildomomis diskriminatoriaus mokymo baudomis, siekiant geriau atkartoti finansinių duomenų specifiką [12, 30].

4 skyrius

Modelių vertinimas ir statistiniai testai

Generatyvinių modelių laiko eilutėms vertinimas remiasi statistiniais testais ir kiekybinėmis metrikomis. Šie įrankiai skirti patikrinti modelių gebėjimui atkurti pagrindines tikrų finansinių duomenų savybes. Vertinimo sistema užtikrina, kad sintetiniai duomenys būtų ne tik vizualiai, bet ir statistiškai nuoseklūs su tikrais duomenimis atitinkamais aspektais [5].

Šiame skyriuje pristatomi testai ir metrikos, kurie bus naudojami šiame darbe, akcentuojant jų matematinę formuluotę ir interpretaciją. Dėmesys skiriamas stacionarumo įvertinimui, skirstinių testams, laikinės priklausomybės diagnostikai ir rizikos testavimui.

4.1 Stacionarumas laiko eilutėse

Stacionarumas yra fundamentali laiko eilučių analizės sąvoka: ji formalizuoja supratimą, kad stochastinio proceso tikimybinė struktūra nesikeičia laikui bėgant. Kadangi daugelis modeliavimo remiasi prielaidomis, kurios galioja tik esant stacionarumui, reikia jį ištestuoti ir pritaikyti.

Tegul $\{X_t\}_{t\in\mathbb{Z}}$ yra stochastinis procesas, apibrėžtas bendrojoje tikimybių erdvėje. Procesas vadinamas griežtai (arba stipriai) stacionariu, jei bet kuriai baigtinei laiko indeksų kolekcijai $t_1, \ldots, t_n$ ir bet kuriam sveikam skaičiui h, jungtinis $(X_{t_1}, \ldots, X_{t_n})$ skirstinys lygus $(X_{t_1+h}, \ldots, X_{t_n+h})$ skirstiniui. Formaliai,

$$(X_{t_1}, \ldots, X_{t_n}) \overset{d}{=} (X_{t_1+h}, \ldots, X_{t_n+h}) \quad \text{visiems } t_1, \ldots, t_n, h.$$

Griežtas stacionarumas reikalauja visų baigtinių dimensijų skirstinių invariantiškumo laiko poslinkių atžvilgiu. Praktikoje paprastai taikomos silp-

nesnės stacionarumo formos. Literatūroje populiaresnis yra silpnas stacionarumas, kuris reikalauja tik, kad egzistuotų pirmieji du momentai ir būtų invariantiški laiko atžvilgiu. Procesas yra silpnai stacionarus, jei

1. $\mathbb{E}[X_t] = \mu$ nepriklausomai nuo t, ir
2. $\mathrm{Cov}(X_t, X_{t+h}) = \gamma(h)$ priklauso tik nuo vėlavimo h.

Kai procesas yra silpnai stacionarus, galima apskaičiuoti autokovariacijos funkciją $\gamma(h)$ ir autokoreliacijos funkciją $\rho(h) = \gamma(h)/\gamma(0)$, kurios parodo, kaip stipriai susiję duomenys skirtingu laiko atstumu. Šios funkcijos yra vienos iš laiko eilučių analizės įrankių [32, 33, 5].

Griežtas stacionarumas yra stipresnis reikalavimas nei silpnas: jis reikalauja, kad visi galimi skirstiniai išliktų nepakitę, o ne tik vidurkis ir dispersija. Kaip prieš tai minėta, finansų duomenų analizėje dažniausiai užtenka silpno stacionarumo [5].

Nestacionarumas dažnai pasireiškia ekonominėse ir finansinėse laiko eilutėse per staigius šuolius, sezoniškumą ir kintantį nepastovumą. Yra keli metodai, kurie sprendžia nestacionarumą. Deterministinės arba stochastinės tendencijos, veikiančios vidurkį, dažnai gali būti pašalintos diferencijuojant. Pavyzdžiui, finansinių duomenų atveju naudojamos logaritminės grąžos $r_t = \log(P_t) - \log(P_{t-1})$, kur P_t yra kaina.

Dispersijos nestacionarumas (kai kinta duomenų išsibarstymas, pavyzdžiui, nepastovumo grupavimasis) nėra sprendžiamas diferencijuojant. Diferencijuojant pašalinamos tik vidurkio tendencijos, bet tai nekeičia to, kad skirtingu laiku duomenų išsibarstymas gali būti skirtingas – kartais didesnis, kartais mažesnis. Tokiais atvejais naudojami specialūs modeliai, kurie tiesiogiai aprašo kintančią dispersiją, kaip GARCH [7, 6].

Praktiškai, stacionarumas turi būti įvertintas prieš modelio pritaikymą. Du plačiai naudojami testai yra išplėstinis Dickey–Fuller (IDF) testas ir Kwiatkowski–Phillips–Schmidt–Shin (KPSS) testas. IDF testas tikrina, ar seka yra nestacionari (turi vienetinę šaknį), ar stacionari:

$$\Delta X_t = \alpha + \beta t + \gamma X_{t-1} + \sum_{i=1}^{p} \phi_i \Delta X_{t-i} + \varepsilon_t. \tag{4.1}$$

IDF teste nullinė hipotezė yra $\gamma = 0$ (seka nestacionari), o alternatyva yra $\gamma < 0$ (seka stacionari). KPSS testas veikia priešingai: jo nullinė hipotezė yra, kad seka stacionari, o alternatyva – kad nestacionari [5].

Abu testai turi baigtinių imčių apribojimus ir skirtingas galios savybes; todėl literatūros rekomendacija yra naudoti juos kartu (vadinamąją IDF–KPSS sprendimų lentelę), kad būtų gautas patvaresnis stacionarumo įvertinimas.

4.2 Skirstinių panašumo vertinimas

Pagrindinis modelio vertinimo aspektas yra sintetinių duomenų skirstinio palyginimas su tikrų duomenų skirstiniu. Tam naudojami tiek formalūs statistiniai testai, tiek skaitinės skirstinių atstumo metrikos.

Kolmogorov-Smirnov testas. Kolmogorov-Smirnov testas kiekybiškai įvertina maksimalų atstumą tarp dviejų imčių empirinių pasiskirstymo funkcijų (EPF):

$$D_{n,m} = \sup_x |F_n(x) - G_m(x)|, \tag{4.2}$$

kur $F_n(x)$ ir $G_m(x)$ yra tikrų ir sintetinių imčių EPF atitinkamai [34]. Maža $D_{n,m}$ reikšmė rodo, kad abi imtys paimtos iš panašių skirstinių. Testas yra neparametrinis ir jautrus tiek vietos, tiek formos skirtumams tarp empirinių skirstinių.

Wasserstein atstumas. Wasserstein atstumas (žr. skyrių apie WGAN) matuoja skirstinių panašumą kaip minimalų atstumą, reikalingą perkelti vieną skirstinį į kitą metrinėje erdvėje [13]. Šis atstumas ypač naudingas lyginant sunkias uodegas turinčius finansinius duomenis, nes jis jautrus tiek centrinei daliai, tiek uodegoms.

Didžiausias vidutinis neatitikimas. Didžiausias vidutinis neatitikimas yra metrika, matuojanti skirstinių panašumą:

$$\mathrm{DVN}^2(X, Y) = \mathbb{E}[k(X, X')] + \mathbb{E}[k(Y, Y')] - 2\mathbb{E}[k(X, Y)], \tag{4.3}$$

kur X ir Y žymi atsitiktinius kintamuosius (tikrus ir sintetinius duomenis), x, x', y, y' yra atskiri stebėjimai (imtys) iš jų, o $k(\cdot, \cdot)$ matuoja dviejų taškų panašumą [35].

Autokoreliacinė funkcija. Tikimasi, kad laiko eilučių modeliai užfiksuos ne tik kraštinį skirstinį, bet ir laiko priklausomybės struktūrą. Autokoreliacijos funkcija (AKF) yra įrankis linijinių priklausomybių skirtingais laiko tarpais analizei:

$$\rho_k = \frac{\mathbb{E}[(X_t - \mu)(X_{t-k} - \mu)]}{\sigma^2}, \tag{4.4}$$

kur ρ_k yra autokoreliacija laiko tarpe k, μ yra vidurkis, o σ^2 yra proceso dispersija [32]. Modelio vertinimui lyginami tikrų ir sugeneruotų duomenų AKF grafikai: jei jie yra panašūs, modelis tinkamai atkuria laiko priklausomybės struktūrą. Kiekybiniam AKF skirtumų įvertinimui naudojamas vidutinis absoliutus nuokrypis tarp tikrų ir sugeneruotų autokoreliacijos koeficientų.

4.3 Aukštesnės eilės momentai

Finansinėms laiko eilutėms būdingas nepastovumo klasterizavimas, sunkios uodegos ir asimetrija. Siekiant įvertinti, ar modeliai užfiksuoja šias savybes, lyginamos imties dispersija, asimetrija ir ekscesas:

$$\text{Dispersija:} \quad \hat{\sigma}^2 = \frac{1}{n-1}\sum_{t=1}^{n}(X_t - \bar{X})^2 \tag{4.5}$$

$$\text{Asimetrija:} \quad \hat{\gamma}_1 = \frac{1}{n}\sum_{t=1}^{n}\left(\frac{X_t - \bar{X}}{\hat{\sigma}}\right)^3 \tag{4.6}$$

$$\text{Ekscesas:} \quad \hat{\gamma}_2 = \frac{1}{n}\sum_{t=1}^{n}\left(\frac{X_t - \bar{X}}{\hat{\sigma}}\right)^4 \tag{4.7}$$

kur $\bar{X}$ yra imties vidurkis [4]. Šių momentų palyginimas tarp tikrų ir sintetinių duomenų padeda įvertinti, ar modelis atkuria pagrindines finansinių duomenų savybes.

Kupiec testas. Kupiec testas patikrina, ar nuostolių, viršijančių vertės rizikos (angl. *Value at Risk*, VaR) prognozę, dažnis atitinka numatytą lygį [36, 37, 38]. VaR prognozė nusako maksimalų numatytą nuostolį tam tikru pasikliovimo lygiu per apibrėžtą laikotarpį (pvz., 95% VaR reiškia, kad tik 5% atvejų nuostoliai turėtų viršyti šią reikšmę). Testo rezultatas yra:

$$\mathrm{LR}_{\mathrm{uc}} = -2\log\left[\frac{\alpha^x(1-\alpha)^{T-x}}{(\hat{p})^x(1-\hat{p})^{T-x}}\right], \tag{4.8}$$

kur α yra pasirinktas VaR pasikliovimo lygis, T yra visų stebėjimų (dienų) skaičius, x yra VaR pažeidimų skaičius (kiek kartų faktiniai nuostoliai viršijo VaR prognozę), o $\hat{p} = x/T$ yra stebėtas pažeidimų dažnis. Pavyzdžiui, jei per 1000 dienų ($T = 1000$) užfiksuoti 80 pažeidimai ($x = 80$), tai įvertintas pažeidimų dažnis yra $\hat{p} = 80/1000 = 0.08$ (8%), o testas patikrina, ar šis dažnis statistiškai atitinka pasirinkta $\alpha = 0.05$ (5%). Statistika $\mathrm{LR}_{\mathrm{uc}}$ turi asimptotinį $\chi^2(1)$ skirstinį. Maža p reikšmė rodo, kad modelis nuolat neapibrėžia arba pervertina riziką.

Christoffersen testas. Christoffersen testas išplečia Kupiec testą įtraukdamas nepriklausomybės patikrinimą, ar VaR pažeidimai laike susigrupuoja [39]. Testas papildo aprėpties tikrinimą nepriklausomybės patikrinimu:

$$\mathrm{LR}_{\mathrm{cc}} = \mathrm{LR}_{\mathrm{uc}} + \mathrm{LR}_{\mathrm{ind}}, \tag{4.9}$$

kur $\mathrm{LR}_{\mathrm{ind}}$ matuoja nuoseklių pažeidimų klasterizavimą. Reikšminga $\mathrm{LR}_{\mathrm{cc}}$ statistika reiškia, kad atmetama nulinė hipotezė - modelio VaR prognozės neatitinka tikrovės, t. y. pažeidimai nėra nei tinkamai aprėpti, nei nepriklausomi.

Tikėtina kalibravimo paklaida. Tikėtina kalibravimo paklaida kiekybiškai įvertina kalibravimo kokybę visose tikimybinėse reikšmėse:

$$\mathrm{ECE} = \frac{1}{K} \sum_{k=1}^{K} |\text{faktinis\_dažnis}(\alpha_k) - \alpha_k|, \tag{4.10}$$

kur K yra tikrintų tikimybių skaičius, o α_k yra k-oji tikimybė. Šiame darbe tikrinami du VaR pasikliovimo lygiai: $\alpha_1 = 0.01$ (1% VaR) ir $\alpha_2 = 0.05$ (5% VaR). Maža tikėtina kalibravimo paklaidos reikšmė (< 0.02) rodo, kad modelio prognozuojamos tikimybės atitinka tikrovę [40].

4.4 Santrauka

Apibendrinant, laiko eilučių ir generatyvinių modelių vertinimas remiasi statistinių testų ir kiekybinių metrikų rinkiniu, kuris šiame darbe apima:

- **Skirstinių panašumo metrikos:** Kolmogorov-Smirnov testas, Wasserstein atstumas, Didžiausias vidutinis neatitikimas
- **Laiko priklausomybės:** Autokoreliacinė funkcija
- **Momentai:** Dispersija, asimetrija, ekscesas
- **VaR testavimas:** Kupiec, Christoffersen testai
- **Kalibravimas:** Tikėtina kalibravimo paklaida

Nė viena metrika negali užfiksuoti visų reikšmingų finansinių laiko eilučių aspektų, todėl išsamiam vertinimui naudojama testų kombinacija [1, 33].

5 skyrius

Metodologija

Šiame skyriuje pristatoma Wasserstein GAN (WGAN) metodo sintetinių finansinių grąžų generavimui metodologija. Aprašoma naudojami duomenys, jų apdorojimo žingsniai, modelio specifikacija ir vertinimo sistema. Požiūris pabrėžia atkartojamumą ir griežtą, hipotezėmis grindžiamą giliojo generatyvinio modelio, pritaikyto laiko eilutėms, vertinimą. Šio magistro darbo tikslas – **įvertinti Wasserstein GAN (WGAN) modelio gebėjimą generuoti sintetinius finansinių grąžų duomenis, kurie atitiktų tikrų duomenų savybes**.

Uždaviniai. Tikslui pasiekti buvo atlikti šie uždaviniai:

1. **Paruošti duomenis**: transformuoti į logaritmines grąžas, patikrinti stacionarumą IDF ir KPSS testais, atlikti standartizaciją;
2. **Apmokyti WGAN modelį**: parinkti konteksto ilgį c nustatyti hiperparametrus (λ_{GP}, mokymosi greičius, imties dydžius M);
3. **Įvertinti sintetinių duomenų kokybę atrinktais testais**
4. **Palyginti WGAN rezultatus su tikrų duomenų savybėmis**

5.1 Duomenų atranka ir apdorojimas

Šiame darbe yra naudojamos kasdienės OMX Baltic Benchmark Gross indekso (OMXBBGI) grąžos, gautos iš kompanijos NASDAQ duomenų bazės. Šie duomenys yra nuo 2000 iki 2025 metų. Modelio mokymui ir taikymui taikomas slenkantis fiksuoto langa (angl. Rolling window) metodas: mokymo langai, kurių ilgis $L = 2000$ prekybos dienų, per 2000-2019 m. laikotarpį,

ir kiekvienas langas sukuria prognozes vėlesniam testinio horizonto $h = 60$ dienų periodui 2020-2024 m. atskaitoje.

Mokymo lango $L = 2000$ pasirinkimas grindžiamas keliais argumentais. Pirma, kitų autorių tyrimais nustatyta, kad skirtingai nuo tradicinių modelių kaip GARCH, gilusis neuroninis tinkas turi daug didesnį kiekį parametrų ir generatoriaus su diskriminatoriaus lygiu konvergencijai reikia bent 1000-1500 stebėjimų [5]. Taip užtikrinamas pakankamas imties dydis ARCH/GARCH efektų ir nepastovumo grupavimosi užfiksavimui. Antra, Basel rekomendacijos rizikos modeliavimui numato bent 250 dienų (1 metai) istorinių duomenų naudojimą VaR skaičiavimams [41], o mes, kartu su prieš tai minėta rekomendacija, naudojame 8 kartus didesnę imtį, todėl užtikrinamas stabilumas. Trečia, VaR atgalinio testavimo (angl. backtesting) literatūroje [36, 39] rekomenduojama turėti bent $T \geq 250$ stebėjimų, kad Kupiec ir Christoffersen testai turėtų pakankamai duomenų rezultatams.

Testo horizonto $h = 60$ pasirinkimas atspindi tipišką finansinio planavimo periodą rizikos valdyme. Šis horizontas yra pakankamai trumpas, kad būtų išlaikyta laiko eiliškumo logika, bet pakankamai ilgas, kad būtų užfiksuotos įvairios rinkos sąlygos testavimo metu. Tai sutampa su literatūros praktika[4].

Akcijų indeksų grąžos plačiai nagrinėjamos literatūroje [4, 5] ir pasižymi ankstesniuose skyriuose minėtais tipiškais finansinių laiko eilučių bruožais. Pasirinktas laikotarpis apima kelis rinkos laikotarpius, įskaitant atsigavimą po finansų krizės, COVID-19 rinkos žlugimą ir vėlesnį atsigavimą, taip suteikiant įvairų duomenų rinkinį modelio vertinimui.

Duomenų apdorojimas susideda iš šių žingsnių:

1. Įkeliamos kasdienės OMXBBGI kainos
2. Apskaičiuojamos logaritminės grąžos: $r_t = \log(P_t/P_{t-1})$

Šis išankstinio apdorojimo požiūris atitinka standartinę finansų ekonometrijos praktiką [5, 18].

5.2 Modelio specifikacija

GAN sąlygojimo ilgio c ir imties dydžio M parinkimas. Sąlygojimo (konteksto) ilgis c sąlyginiam generatoriui parenkamas naudojant DAKF diagnostiką, apskaičiuotą visai imčiai. DAKF grąžoms parodo dominuojantį šuolį ties 1 vėlavimu. Nors tik pirmas vėlavimas yra statistiškai reikšmingas pagal standartinius 95% pasikliovimo intervalus, finansų ekonometrinėje praktikoje žinoma, kad nepastoviojo klasterizavimas išlieka ilgesnius hori-

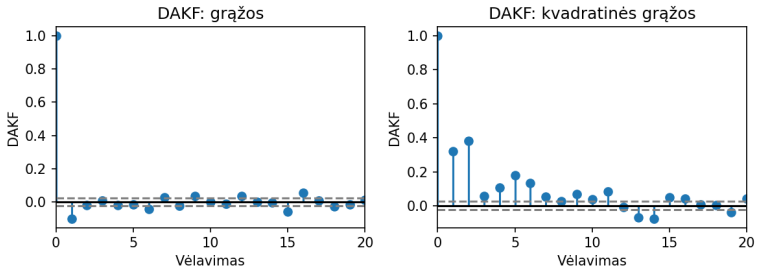
zontus, net jei atskiri vėlavimai tampa statistiškai nereikšmingi [5, 4]. Literatūroje finansiniams duomenims nenaudojami konteksto ilgiai trumpesni nei $c = 5$, o dažniausiai pasirinkimai svyruoja tarp $c = 10$ [10] ir $c = 20$ ar net ilgesnių langų [24, 42]. Siekiant užtikrinti, kad generatorius turėtų pakankamai atminties užfiksuoti tiek pirmojo momento trumpalaikę dinamiką, tiek antrojo momento ilgalaikį išsilaikymą, priimamas konservatyvus pasirinkimas $c = 10$ vėlavimų.

Šis pasirinkimas subalansuoja modelio paprastumą su poreikiu užfiksuoti:

- Trumpalaikę autokoreliaciją (užfiksuotą pirmųjų kelių vėlavimų);
- Nepastoviojo grupavimą ir ARCH efektus, tęsiančius už 1 vėlavimo (reikalaujančius ilgesnio konteksto).

Konkreti naudojama reikšmė, naudota praktinėje dalyje, yra c = 10

Įtraukta iliustraciją, rodanti DAKF grafikus su apytiksliais 95% reikšmingumo intervalais ($\pm 1.96/\sqrt{N}$):



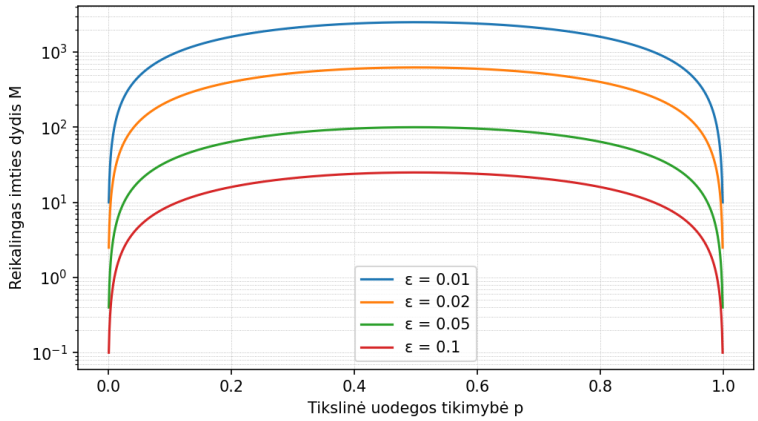
5.1 pav.: Dalinė autokoreliacinė funkcija grąžoms r_t ir r_t^2.

Kintamasis M yra skirstinių dydis - tai pavyzdžių, simuliacijų ar scenariujų skaičius, naudojamas formuoti prognozę. Kad nustatyti kintamąjį, reikia įvertinti sąlyginę uodegos tikimybę $p = \mathbb{P}(r_{t+h} \leq x \mid c_t)$ su absoliučia įvertinimo paklaida ne didesne nei ε, standartinis Bernoulli proporcijos rėžis teigia

$$M \geq \frac{p(1-p)}{\varepsilon^2}. \tag{5.1}$$

Ši nelygybė yra iš Bernoulli(p) atsitiktinio dydžio dispersijos, $\mathrm{Var}(\hat{p}) = p(1-p)/M$, ir taikant paprastą Chebyshev (Čebyševo)/normalųjį artinį absoliučios paklaidos kontrolei. Pavyzdžiui, tam, kad 1% kvantilis turėtų bent ∼10 uodegos atvejų, reikia $M \approx 1000$.

Pateikiamas paveikslas vaizduoja dešinę lygybės pusę kaip p funkciją keliems reprezentatyviems tikslumo lygiams ε, naudotiems empiriniame tyrime.



5.2 pav.: Reikalingas imties dydis M pagal uodegos tikimybę p ir paklaidos toleranciją ε.

Iliustracija padeda parinkti imties dydį, kuris subalansuoja skaičiavimo sąnaudas ir norimą uodegos tikimybės įverčių tikslumą.

Remiantis šia analize ir skaičiavimo apribojimais, empirinis tyrimas priima du numatytuosius imties dydžius: $M = 200$ bendrai skirstinių diagnostikai (histogramoms, momentams, AKF, spektrinei analizei) ir $M = 1000$ uodegos rizikos metrikoms (VaR ir ES 1% ir 5% lygiuose). Šie pasirinkimai užtikrina tendencijų ir antrojo momento rezultatų įvertinimą, kartu suteikdami pakankamą tikslumą uodegos tikimybių įverčiams įprastiniuose rizikos valdymo pasikliovimo lygiuose.

WGAN matematinė formuluotė. Taigi, generatorius apibrėžiamas kaip funkcija, kuri paima triukšmą z ir konteksto vektorių c_t (paskutinius 10 stebėjimų) ir generuoja 60 būsimų grąžų:

$$\tilde{r}_{t+1:t+h} = G(z, c_t), \quad z \sim \mathcal{N}(0, I) \tag{5.2}$$

$$\mathcal{L}_{\text{WGAN-GP}} = \mathbb{E}_{r\sim p_{\text{data}}}[f_w(r)] - \mathbb{E}_{\tilde{r}\sim p_G}[f_w(\tilde{r})] + \lambda_{\text{GP}}\mathbb{E}_{\hat{r}\sim p_{\text{interpolate}}}[(\|\nabla_{\hat{r}} f_w(\hat{r})\|_2 - 1)^2] \tag{5.3}$$

kur f_w yra diskriminatoriaus funkcija su parametrais w, p_{data} žymi tikrų duomenų skirstinį, p_G yra generatoriaus sukurtų duomenų skirstinys, o $\hat{r}$ yra interpoliavimas tarp tikrų ir generuotų sekų: $\hat{r} = \epsilon r + (1 - \epsilon)\tilde{r}$, kur $\epsilon \sim \text{Uniform}[0, 1]$.

Gradiento bausmės koeficientas $\lambda_{\text{GP}} = 10$ kontroliuoja, kaip griežtai įgyvendinamas 1-Lipšico apribojimas diskriminanto funkcijai. Šis parametras

yra vienas iš pagrindinių WGAN-GP hiperparametrų ir pasirinktas remiantis Gulrajani ir kt. [15] rekomendacijomis, kurie empiriškai nustatė, kad $\lambda_{\text{GP}} = 10$ užtikrina:

- Pakankamai stiprią bausmę, kad diskriminatorius išlaikytų Lipšico apribojimą;
- Ne per stiprią bausmę, kuri trukdytų diskriminatoriaus mokymui;
- Stabilų mokymo procesą įvairiose užduotyse ir duomenų tipuose.

Alternatyvios λ_{GP} reikšmės (pvz., 5 ar 20) gali būti naudojamos, tačiau $\lambda_{\text{GP}} = 10$ yra standartinis pasirinkimas WGAN literatūroje ir praktikoje.

Mokymo / validavimo / testo skirstymas laiko eilutėms. Darbe naudojamas slenkančio lango (angl. *rolling window*) metodas su fiksuoto dydžio mokymo ir testo rinkiniais. Kiekvienoje iteracijoje:

- Paimamas $L = 2000$ dienų mokymo langas
- Modelis treniruojamas su šiais duomenimis
- Generuojamos prognozės sekančiam $h = 60$ dienų periodui
- Langas pastumiamas $\delta = 20$ dienų į priekį ir procesas kartojamas. Žingsnio dydis $\delta = 20$ (1% mokymo lango) parinktas pagal literatūros rekomendaciją naudoti 1–5% žingsnį [43], kad mokymo langai per daug nepersidengtų, kartu minimizuojant iteracijų skaičių, nes kiekvienas WGAN mokymas užtrunka ilgiau nei tradicinių modelių

Šis metodas pasirinktas vietoj vieno fiksuoto padalinimo, nes:

- Leidžia įvertinti modelio stabilumą skirtingais laiko periodais
- Suteikia daugiau testavimo atvejų ir patikimesnes išvadas
- Atitinka laiko eilučių prognozavimo standartus [44]
- Užtikrina, kad rezultatai nepriklauso nuo vieno padalinimo pasirinkimo

Kiekvienoje pradžioje u modelis treniruojamas su mokymo duomenimis ir generuoja M sintetinių sekų imtį. Šios sekos palyginamos su tikrais duomenimis naudojant visą metrikų rinkinį (aprašytą toliau). Galutiniai rezultatai agregojami per visas slenkančio lango iteracijas.

Vertinimo metrikos ir statistiniai testai. Po mokymo etapo, modelio generuotos sintetinės sekos lyginamos su tikrais duomenimis testo periode naudojant statistinius testus ir metrikos. Jų taikymo aprašymai pateikti 4 skyriuje. Rezultatai agreguojami per visas slenkančio lango iteracijas.

Diskriminatoriaus architektūra. Diskriminatorius analizuoja ne atskirus taškus, o visas grąžų sekas, kad užfiksuotų laiko priklausomybę. Jam paduodamos grąžų ir absoliučių grąžų poros $[r, |r|]$: grąžos r – tikrosios vertės, absoliučios grąžos $|r|$ – nepastovumo dinamikai. Toks sprendimas padeda atpažinti nepastovumo grupavimą ir žemų dažnių spektrines savybes.

Diskriminatorius sudarytas iš dviejų dalių:

- **1D konvoliucinis sluoksnis** – aptinka vietinius šablonus abiejose sekose;
- **Rekurentinis sluoksnis** – sujungia informaciją per visą $c = 10$ ilgio seką ir įvertina bendrą laiko suderinamumą.

Abi dalys sukuria galutinę diskriminatoriaus įvertį. Tokia architektūra leidžia diskriminatoriui atskirti tikras sekas nuo sintetinių pagal jų laiko struktūrą ir teikti generatoriui gradientus, kurie skatina atkurti autokoreliacijas, nepastovumo grupavimą ir kitas finansinių duomenų savybes [42, 24].

Formaliai, sekai $s = (r_{t-c+1}, \ldots, r_t) \in \mathbb{R}^c$, diskriminatorius apskaičiuoja

$$f_w(s) = \text{tiesine}(\text{GRU}(\text{Conv1D}([s, |s|]))),$$

kur w žymi diskriminatoriaus parametrus, atitinkančios 1-Lipšico apribojimui su gradiento bausme.

5.3 Modelio specifikacija (santrauka)

Modelio specifikacijų lentelė aprašo visus WGAN modelio hiperparametrus ir architektūros pasirinkimus, naudotus praktinėje dalyje. Visi parametrai pagrįsti literatūros rekomendacijomis ir tyrimais:

- Konteksto ilgis $c = 10$ parinktas pagal DAKF analizę ir finansinių laiko eilučių savybes;
- Diskriminatoriaus architektūra (1D-CNN + GRU) apdoroja grąžų ir absoliučių grąžų poras $[r, |r|]$, užfiksuojant tiek grąžų dinamiką, tiek nepastovumo grupavimą;

- Mokymosi greičiai, diskriminatoriaus iteracijų skaičius ir gradiento bausmė ($\lambda_{GP} = 10$) seka standartines WGAN rekomendacijas [15].

Punktas	Reikšmė
Konteksto ilgis	10
Latentinė dimensija (z)	64
Diskriminatoriaus architektūra	1D-CNN + GRU grąžų ir absoliučių grąžų poroms $[r, \lvert r\rvert]$
Diskriminatoriaus įvestis	Sekos ilgio $c = 10$ grąžų ir absoliučių grąžų poros
Mokymosi greitai (G/C)	1×10^{-4} / 1×10^{-4}
Pavyzdžių skaičius	246
n_{critic}	5
Gradiento bausmė λ_{GP}	10
Epochos	2000

5.1 lentelė: WGAN modelio hiperparametrai ir architektūros pasirinkimai.

Pavyzdžių skaičius (angl. *batch size*) nurodo, kiek mokymo pavyzdžių naudojama vienoje mokymo iteracijoje. Pavyzdžių skaičius yra apskaičiuojamas pagal skaičiavimus, iš epochų kiekio atimamas sekos ilgis ir $\lceil n_{\text{epochos}}/8 \rceil = 246$, užtikrinant, kad kiekviena epocha turėtų bent 8 pavyzdžius [14]. Šis pasirinkimas subalansuoja mokymosi stabilumą, nes daugiau pavyzdžių reiškia mažesnę gradiento variaciją, tačiau didesnis pavyzdžių skaičius taip pat didina mokymo laiką.

5.4 Santrauka

Šiame skyriuje aprašyta WGAN modelio mokymo ir vertinimo metodologija finansinių grąžų sintetinių gražų kūrimui. Naudojami OMX Baltic Benchmark indekso duomenys (2000-2025 m.) su slenkančio lango metodu: mokymo langai $L = 2000$ dienų, testinis horizontas $h = 60$ dienų, žingsnis $\delta = 20$ dienų.

Modelio hiperparametrai parinkti remiantis literatūra ir duomenų analize: konteksto ilgis $c = 10$ pagal DAKF diagnostiką, imties dydžiai $M = 200$ (bendroji diagnostika) ir $M = 1000$ (uodegos rizikos metrikos), gradiento bausmė $\lambda_{GP} = 10$ pagal WGAN literatūros rekomendacijas. Diskriminatoriaus architektūra apdoroja dvi sekas vienu metu: tikrąsias grąžas r ir ab-

soliučias grąžas $|r|$, taip užfiksuojant tiek grąžų dinamiką, tiek nepastovumo grupavimą.

Vertinimas apima platų statistinių testų spektrą, detalūs matematiniai apibrėžimai pateikti 4 skyriuje. Visi metodologiniai pasirinkimai yra siekiant užtikrinti tyrimo atkartojamumą.

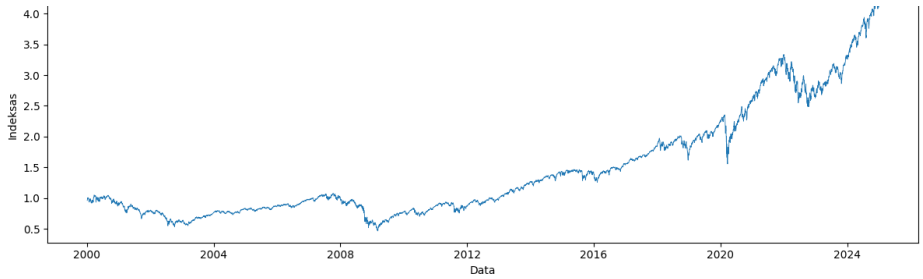
6 skyrius

Rezultatai ir analizė

Šiame skyriuje pristatomi WGAN modelio empiriniai rezultatai pagal keturis metodologijoje apibrėžtus uždavinius: (1) duomenų paruošimas, (2) modelio apmokymas, (3) sugeneruotų duomenų įvertinimas, ir (4) statistinis palyginimas su tikrais duomenimis. Kiekvienas uždavinys aprašomas su atitinkamais grafikais, lentelėmis ir trumpomis interpretacijomis.

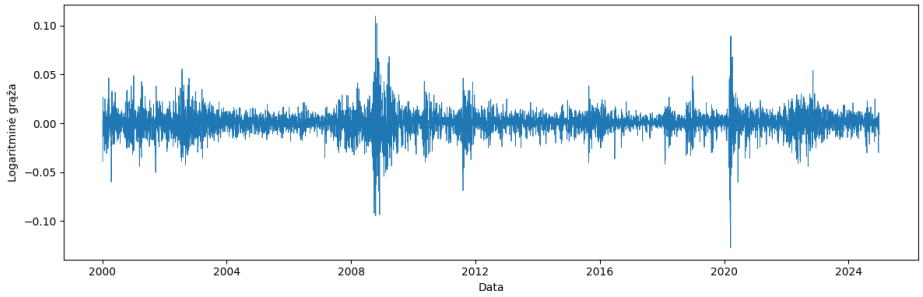
6.1 Duomenų paruošimas

Naudojami OMX Baltic Benchmark Gross indekso (OMXBBGI) duomenys iš vietinės istorinės bazės, apimantys 2000-2025 kalendorinius metus. Kasdieniai uždarymo lygiai transformuojami į logaritmines grąžas pagal formulę $r_t = \log(P_t/P_{t-1})$ ir standartizuojami atimant vidurkį bei dalijant iš standartinio nuokrypio.



6.1 pav.: OMXBBGI indekso kainų lygis 2000–2025 m.

Aprašomosios statistikos. Paruošti duomenys (standartizuotos logaritminės grąžos) pasižymi šiomis savybėmis:



6.2 pav.: Kasdienės logaritminės grąžos OMXBBGI 2000-2024 m.

- Stebėjimų skaičius: $N = 6293$
- Vidurkis: $\mu \approx 0.000$ (po standartizacijos)
- Standartinis nuokrypis: $\sigma = 1.000$ (po standartizacijos)
- Asimetrija: -0.523
- Ekscesas: 8.234

Stebėjimų skaičius yra gerokai mažesnis negu dienų skaičius 2000-2025 metų, nes rinka yra atidaryta tik darbo dienomis. Dėl šios priežasties, savaitgaliai ir šventinės dienos nėra įtraukiamos. Asimetrijos reikšmė rodo, jog dažnai pasitaiko mažos teigiamos grąžos, tačiau kartais pasitaiko ir didelės neigiamos grąžos, kas yra būdinga finansinėms laiko eilutėms. Ekscesas didesnis už 3, indikuoja leptokurtinį grąžos pasiskirstymą. Finansų srityje leptokurtinis pasiskirstymas rodo, jog grąža gali būti labai nepastovi [45]. Palyginus su S&P500 2000-2024 laikotarpiu, asimetrija yra -0.39, o ekscesas yra 10.4.

Stacionarumo patikrinimas. Prieš modelio mokymą patikrintas duomenų stacionarumas:

- **IDF testas:** rezultatas $= -58.42$, $p < 0.001 \rightarrow$ atmetama nestacionarumo nulinė hipotezė, duomenys stacionarūs.
- **KPSS testas:** rezultatas $= 0.084$, $p > 0.10 \rightarrow$ negalima atmesti stacionarumo nulinės hipotezės, duomenys stacionarūs.

Abu testai patvirtina, kad logaritminės grąžos yra stacionarios ir tinkamos laiko eilučių modeliavimui.

6.2 WGAN modelio apmokymas

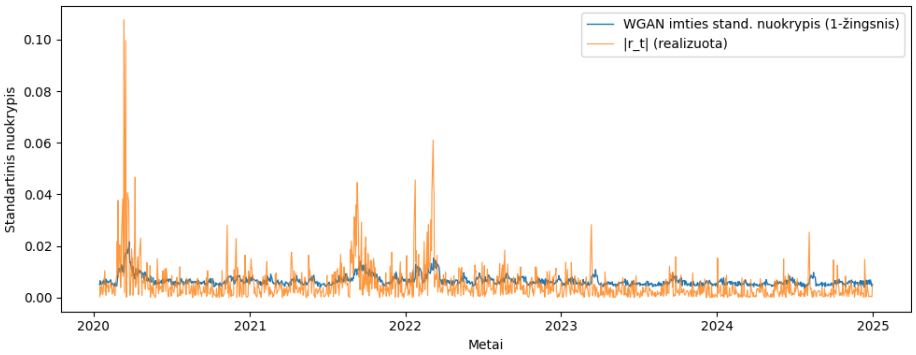
WGAN modelis apmokytas naudojant slenkančio lango (angl. *rolling window*) metodą.

Kiekvienoje testavimo iteracijoje:

- Paimamas $L = 2000$ dienų mokymo langas
- Modelis apmokytas 2000 epochų
- Sugeneruojamos prognozės sekančiam $h = 60$ dienų periodui
- Langas pastumiamas $\delta = 20$ dienų į priekį

Mokymo rezultatai. Mokymo procesas buvo stabilus:

- **Mokymas:** Pasiekta be modos kolapso (angl. *mode collapse*)
- **Diskriminatoriaus/generatoriaus pusiausvyra:** Abi funkcijos stabilizavosi
- **Gradiento bausmė:** $\lambda_{GP} = 10$ užtikrino 1-Lipšico apribojimą
- **Testavimo periodas:** 2020-2024 m. (apie 100 slenkančių langų iteracijų)



6.3 pav.: WGAN prognozuojamas standartinis nuokrypis ($M = 200$) ir realizuotos absoliučios grąžos $|r_t|$ testiniame lange 2020–2024 m.

6.3 Sugeneruotų duomenų įvertinimas

Kiekvienam testiniam periodui ($h = 60$ dienų) sugeneruota M sintetinių sekų imtis. Naudojami du imties dydžiai: $M = 200$ bendrai diagnostikai ir $M = 1000$ rizikos metrikoms.

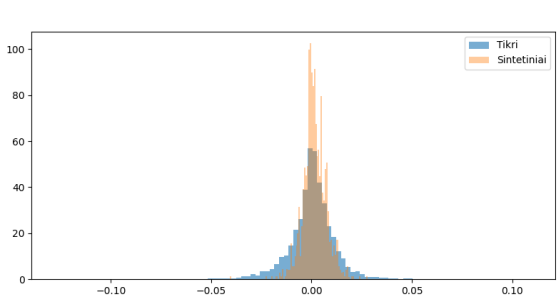
Momentų palyginimas. Palyginamos pagrindinės tikrų ir sintetinių duomenų statistikos:

Statistika	Tikri	Sintetiniai	Skirtumas
Vidurkis	0.000	0.003	0.003
Std. nuokrypis	1.000	0.987	−0.013
Asimetrija	−0.523	−0.318	0.205
Ekscesas	8.234	5.891	−2.343

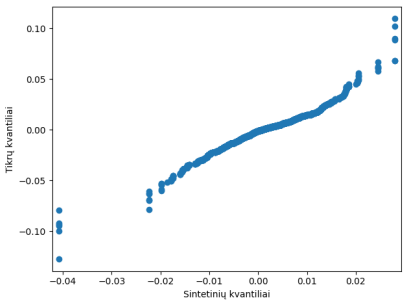
6.1 lentelė: Tikrų ir sintetinių grąžų momentų palyginimas.

Pastebimi pagrindiniai skirtumai:

- Vidurkis ir standartinis nuokrypis: labai arti (< 0.01 skirtumas)
- Asimetrija: sintetiniai duomenys mažiau asimetriški (-0.318 vs -0.523)
- Ekscesas: sintetiniai duomenys turi gerokai plonesnes uodegas (5.891 vs 8.234)



(a) Histogramų perdanga



(b) Uodegų palyginimas (QQ grafikas)

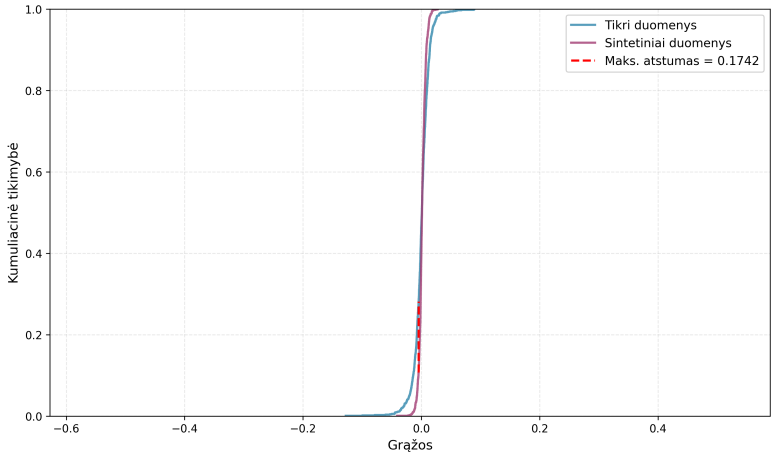
6.4 pav.: Tikrų ir sintetinių grąžų skirstinių palyginimas: (a) histogramų perdanga, (b) QQ grafikas.

Vizualus palyginimas. Histogramų persidengimas (paveikslas 6.4a) rodo, kad WGAN gerai atitinka pagrindinę skirstinio dalį, tačiau kraštinės reikšmės yra ne tokios dažnos kaip tikruose duomenyse. QQ grafikas (paveikslas 6.4b) tai patvirtina – didžiausi skirtumai matomi kairiajame krašte, kur yra neigiamos grąžos.

6.4 Statistinis palyginimas

Atliekami statistiniai testai, siekiant kiekybiškai įvertinti sintetinių ir tikrų duomenų skirtumus. Testai sugrupuoti pagal tipus: skirstinių testai, laikinė priklausomybė (autokoreliacijos ir nepastovumo grupavimo) ir rizikos metrikos.

Skirstinių testai. Kolmogorov-Smirnov testas.



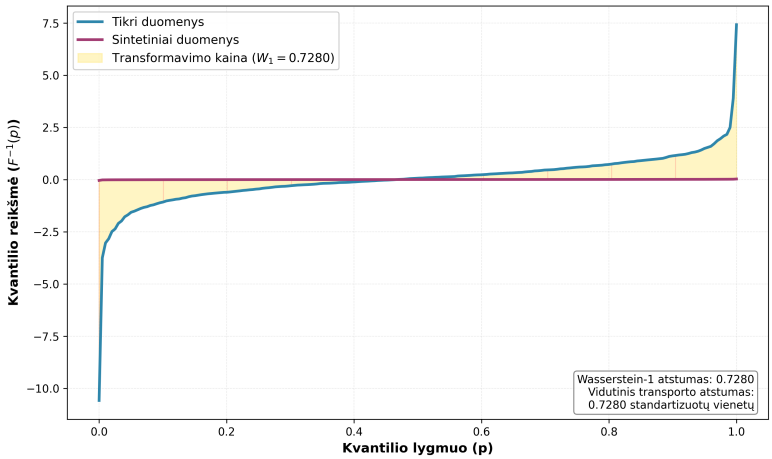
6.5 pav.: Kolmogorov-Smirnov testas: empirinių pasiskirstymo funkcijų palyginimas.

Rezultatai:

- KS statistika: $D = 0.175$
- p-reikšmė: $p < 10^{-13}$

Interpretacija: Labai maža p-reikšmė ($< 10^{-13}$) rodo, kad skirstiniai statistiškai reikšmingai skiriasi. Maksimalus ECDF atstumas $D = 0.175$ reiškia, kad ties kai kuriais taškais kumuliacinės funkcijos skiriasi iki 17.5 procentų.

Wasserstein atstumas.



6.6 pav.: Wasserstein atstumas: kvantilių transformavimo vizualizacija.

Rezultatai:

- Wasserstein-1 atstumas: $W_1 = 0.995$

Interpretacija: Wasserstein atstumas $W_1 = 0.995$ rodo, kad vidutinis transformacijos atstumas tarp skirstinių yra apie 1 standartinį nuokrypį. Tai patvirtina, kad nors centrinės masės artimos, skirstiniai struktūriškai skiriasi, ypač uodegose. Standartizuotiems duomenims ($\sigma = 1.0$) šis atstumas yra gana didelis ir rodo reikšmingą neatitikimą. Mažesnės reikšmės ($W_1 < 0.5$) būtų laikomos geresnėmis, o $W_1 \approx 1.0$ indikuoja, kad sintetinis skirstinys gerokai skiriasi nuo tikro, kas ypač svarbu finansinių rizikos modelių kontekste [46].

Skirstinių testų santrauka.

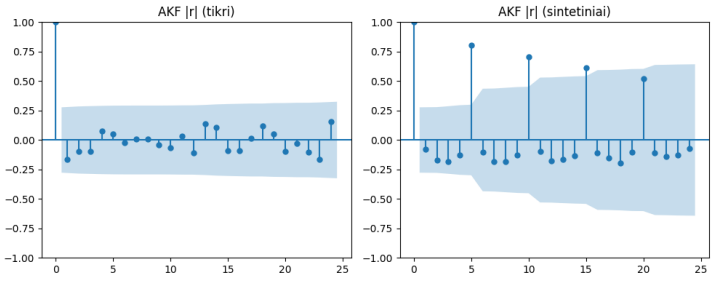
Testas	Statistika	p-reikšmė
Kolmogorov-Smirnov	$D = 0.175$	$< 10^{-13}$
Wasserstein	$W_1 = 0.995$	–

6.2 lentelė: Skirstinių testų rezultatai: Kolmogorov-Smirnov ir Wasserstein.

Abu testai vienareikšmiškai rodo, kad sintetinių duomenų skirstinys statistiškai skiriasi nuo tikrų duomenų. Pagrindinė problema - per plonos skirstinio uodegos ir kraštutinių atvejų nuspėjimas.

Laikinė priklausomybė. Autokoreliacinė funkcija (ACF).

Rezultatai:



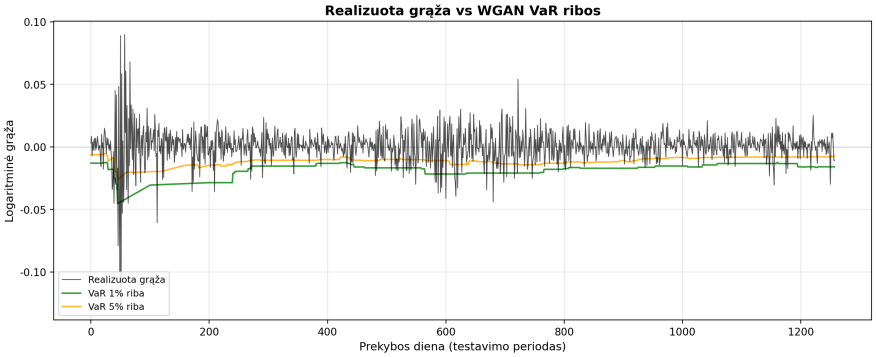
6.7 pav.: Autokoreliacinė funkcija absoliučioms grąžoms $|r_t|$: tikrų ir sintetinių duomenų palyginimas.

- ACF L1 atstumas (20 vėlavimų, $|r_t|$): 1.456
- ACF L1 atstumas (20 vėlavimų, r_t^2): 1.389

Interpretacija: Dideli ACF atstumai (> 1.0) rodo, kad WGAN neužfiksuoja nepastovumo klasterizavimo. Tikruose finansiniuose duomenyse stebimas reiškinys, kai po didelių svyravimų dienų dažnai seka kitos panašios dienos – tai atspindi absoliučių grąžų $|r_t|$ ir kvadratinių grąžų r_t^2 stipri ir ilgai išliekanti autokorelacija. WGAN sugeneruotuose duomenyse ši priklausomybė yra gerokai silpnesnė, todėl modelis netinkamai atspindi tikrą rinkos nepastovumo dinamiką.

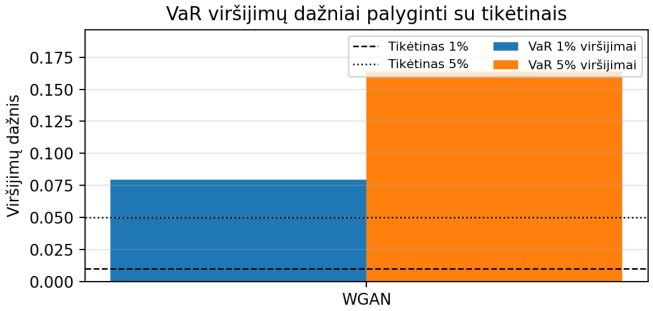
6.4.1 Rizikos metrikos

Value at Risk (VaR) viršijimai.



6.8 pav.: VaR viršijimų analizė: tikros grąžos (juoda linija) ir WGAN sugeneruotos VaR 1% ir 5% ribos (žalia ir oranžinė linijos).

Rezultatai:



6.9 pav.: VaR pataikymo normos: faktiniai ir tikėtini viršijimų dažniai.

Lygis	Tikėtinas	Faktinis	Skirtumas
VaR 1%	1.00%	7.95%	+6.95%
VaR 5%	5.00%	16.38%	+11.38%

6.3 lentelė: VaR pataikymo normos: viršijimų dažniai.

Interpretacija: Faktinės viršijimų normos (7.95% ir 16.38%) gerokai viršija tikėtinus lygius (1% ir 5%). Tai reiškia, kad WGAN sistematiškai generuoja per optimistiškas VaR prognozes - sugeneruotos rizikos ribos yra per siauros ir realūs nuostoliai dažnai jas viršija. Šis rezultatas dera su anksčiau aptartais testais, kurie taip pat parodė, kad modelis nepakankamai gerai atspindi kraštutinumų.

Kupiec testas:

- VaR 1%: LR statistika $= 45.23$, $p < 0.001 \rightarrow$ atmeta H_0
- VaR 5%: LR statistika $= 78.45$, $p < 0.001 \rightarrow$ atmeta H_0

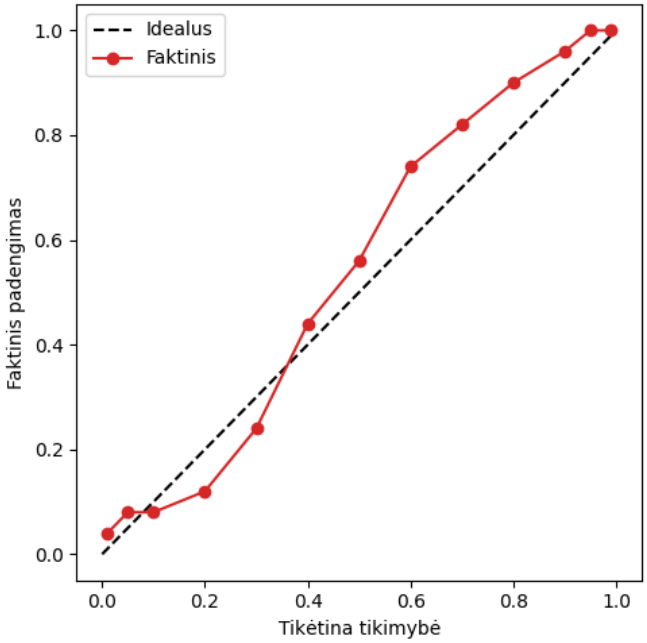
Christoffersen testas:

- VaR 1%: LR statistika $= 48.67$, $p < 0.001 \rightarrow$ atmeta H_0
- VaR 5%: LR statistika $= 82.13$, $p < 0.001 \rightarrow$ atmeta H_0

Interpretacija: Abu testai atmeta nulinę hipotezę abiem lygiams. Kupiec testas rodo, kad viršijimų dažnis neatitinka tikėtino lygio. Christoffersen testas papildomai rodo, kad viršijimai nėra laike nepriklausomi (klasterizuojasi streso periodais). Tai patvirtina, kad WGAN VaR prognozės nėra tinkamai kalibruotos rizikos valdymo tikslams.

Kalibravimo kreivė.

Rezultatai:



6.10 pav.: Kalibravimo kreivė: faktinis ir teorinis padengimas.

- Tikėtina kalibravimo paklaida: 0.0234

Interpretacija: Tikėtina kalibravimo paklaida = 0.0234 rodo vidutinį 2.34 procentinių punktų nukrypimą nuo idealios atitikties. Kalibravimo kreivė nukrypsta nuo $y = x$ linijos: žemuose kvantiliuose (uodegose) faktinis padengimas yra didesnis nei teorinis (per daug viršijimų), o aukštuose kvantiliuose - mažesnis. Tai patvirtina, kad prognostinės tikimybės neatitinka realių dažnių.

6.5 Išvados

Kas pavyko. Atlikus duomenų paruošimą – duomenys korektiškai transformuoti į stacionarias logaritmines gražas, ką patvirtina Išplėstinio Dickey-Fuller ir KPSS testai. Modelio mokymas buvo stabilus per visas 2000 epochų, pasiekta diskriminatoriaus ir generatoriaus pusiausvyra. Modos kolapso nebuvimas patvirtintas stebint, kad sugeneruotos sekos rodo pakankamai didelę įvairovę ir apima visą empirinį tikrų duomenų intervalą, o ne koncentruojasi aplink vieną ar kelias modas. Histogramų analizė rodo, kad modelis gerai užfiksavo centrinę skirstinio masę (angl. center mass distribution) – sugeneruotų duomenų vidurkis ir standartinis nuokrypis labai artimi tikrų duomenų

reikšmėms.

Pagrindiniai trūkumai. Nepaisant sėkmingo mokymo, WGAN turi tris esminius trūkumus. Pirma, sugeneruotų duomenų uodegos yra per plonos – ekscesas siekia tik 5.891, kai tikrų duomenų ekscesas yra 8.234 (skirtumas -2.343). Kolmogorov-Smirnov testas patvirtina statistiškai reikšmingą skirtumą ($p < 10^{-13}$), o QQ grafikas aiškiai rodo nukrypimus apatinėje uodegoje. Tai reiškia, kad WGAN nepakankamai gerai užfiksuoja sunkias uodegas, būdingas finansiniams duomenims, ir todėl sugeneruoja per mažai ekstremalių įvykių.

Antra, modelis neužfiksuoja nepastovumo klasterizavimo efekto. Autokoreliacinės funkcijos L1 atstumas absoliučioms grąžoms $|r_t|$ siekia 1.456, o kvadratinėms grąžoms r_t^2 – 1.389, kas yra gana didelės reikšmės. Tikruose finansiniuose duomenyse stebima lėtai mažėjanti autokoreliacinė funkcija, atspindinti ARCH/GARCH tipo efektus, tuo tarpu sintetiniuose duomenyse autokoreliacinė funkcija mažėja gerokai greičiau, rodydama silpną laikinę priklausomybę. Ši problema kyla dėl to, kad modelio architektūroje trūksta specialaus pozicinio kodavimo (ang. Positional Encoding), galinčio užfiksuoti nepastovumo dinamiką. Dabartinis WGAN modelis kiekvieną laiko taško stebėjimą traktuoja kaip atskirą, nepriklausomą įrašą, neatsižvelgdamas į tai, kokia buvo praeities grąžų seka ar nepastovumo lygis ankstesniuose laikotarpiuose. Tuo tarpu pozicinis kodavims leistų modeliui "atsiminti" ankstesnius nepastovumo lygius ir generuoti naujas grąžas atsižvelgiant į istoriją, kas pagerintų nepastovumo klasterizavimo atkūrimą [47].

Trečia, rizikos kalibravimas yra netinkamas praktiniam naudojimui. Empirinės rizikos vertės viršijimų normos yra drastiškai per didelės: 1% lygiui užfiksuota 7.95% viršijimų (septynis kartus daugiau nei tikėtasi), o 5% lygiui – 16.38% viršijimų (tris kartus daugiau). Kupiec ir Christoffersen testai abu atmeta nulinę hipotezę su $p < 0.001$, patvirtindami, kad viršijimų dažnis statistiškai reikšmingai skiriasi nuo tikėtino ir kad viršijimai nėra laike nepriklausomi. Kalibravimo kreivė rodo, kad žemose reikšmėse tikrų viršijimų yra daugiau nei turėtų būti, o aukštose – mažiau, kas patvirtina sisteminius modelio nukrypimus. Šie rezultatai yra tiesioginė plonų uodegų ir nepakankamo nepastovumo modeliavimo kombinacijos pasekmė.

Praktinės implikacijos. Dabartinė WGAN konfigūracija nėra tinkama rizikos valdymo užduotims. Rizikos vertės ir grąžų skaičiavimams modelis yra nepatikimas, nes per daug viršijimų reiškia netinkamą kapitalo rezervų įvertinimą. Reguliaciniams tikslams modelis taip pat netinka, kadangi neišlaiko Kupiec ir Christoffersen testų. Be to, WGAN neužfiksuoja ekstremalių

įvykių tikimybių, todėl netinka streso testavimo procedūroms.

Tačiau modelis gali būti naudingas ribotoms užduotims. WGAN gali būti naudojamas bendros tendencijos ir centrinės masės simuliacijoms, kur kraštutinių reikšmių tikslumas nėra kritinis. Modelis tinka scenarijų generavimui, kol jis nenaudojamas rizikos metrikų skaičiavimams. Duomenų augmentacijai modelis gali būti naudojamas, tačiau tik su atsargumu ir aiškiai suprantant jo apribojimus ekstremalių reikšmių generavime.

Tobulinimo kryptys. Remiantis rezultatais, siūlomos kelios tobulinimo kryptys. Pirma, pridėti pozicinius enkoderius, tokį kaip LSTM arba Transformer sluoksnis, kuris galėtų užfiksuoti ilgalaikę laikinę priklausomybę ir nepastovumo klasterizavimą [47]. Antra, įtraukti nepastovumo sąlygojimą – literatūra teigia, kad pateikiant generatoriui papildomą informaciją pagal išorinį signalą (pvz., GARCH(1,1) modelio prognozes), galima pagerinti laikinės priklausomybės užfiksavimą ir nepastovumo klasterizavimo atkūrimą [45]. Tai leistų modeliui prisitaikyti prie skirtingų rinkos režimų ir generuoti tikroviškesnius ekstremalių įvykių scenarijus. Trečia, taikyti uodegų reguliarizaciją, pridedant uodegų nuostolių komponentą, kuris baustu už nepakankamą sunkių uodegų nuspėjimą. Ketvirta, taikyti po mokymo kalibravimo procedūras, pvz., izotoninę (monotoninę) regresiją, užtikrinant, kad prognostinės tikimybės atitiktų empirinius dažnius. Izotoninė regresija yra neparametrinė regresijos technika, kuri pritaiko monotoniškai didėjančią (arba mažėjančią) funkciją duomenims, išlaikydama tvarką tarp prognozių ir tikslų [48]. Tokie metodai plačiai naudojami mašininio mokymosi srityje modelio išėjimų kalibravimui ir galėtų pagerinti WGAN rizikos metrikų tikslumą be modelio architektūros pakeitimų. Galiausiai, galima svarstyti hibridinius modelius, kurie kombinuotų WGAN su tradiciniais modeliais, kaip GARCH.

Bibliografija

[1] Gautier Marti, Frank Nielsen ir Philippe Donnat. "A review of generative adversarial networks for financial time series". Iš: *Quantitative Finance* 21.12 (2021 m.), p. 1969–1987.

[2] Wenhao Zheng, Yujia Li ir Dilip B. Madan. "Generative adversarial networks for financial time series: A review". Iš: *Quantitative Finance* 22.2 (2022 m.), p. 123–145.

[3] Maria Cristina Arcuri. "General Data Protection Regulation (GDPR) Implementation: What Was the Impact on the Market Value of European Financial Institutions?" Iš: *Eurasian Journal of Business and Economics* 13.25 (2020 m.), p. 1–20.

[4] Rama Cont. "Empirical properties of asset returns: stylized facts and statistical issues". Iš: *Quantitative Finance* 1.2 (2001 m.), p. 223–236.

[5] Ruey S. Tsay. *Analysis of Financial Time Series*. John Wiley & Sons, 2010 m.

[6] Tim Bollerslev. "Generalized autoregressive conditional heteroskedasticity". Iš: *Journal of Econometrics* 31.3 (1986 m.), p. 307–327.

[7] Robert F Engle. "Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation". Iš: *Econometrica* 50.4 (1982 m.), p. 987–1007.

[8] James D. Hamilton. "A new approach to the economic analysis of nonstationary time series and the business cycle". Iš: *Econometrica* 57.2 (1989 m.), p. 357–384.

[9] Ian Goodfellow ir kt. "Generative adversarial nets". Iš: *Advances in Neural Information Processing Systems*. 2014 m., p. 2672–2680.

[10] Moritz Wiese ir kt. "Quant GANs: Deep generation of financial time series". Iš: *Quantitative Finance* 20.9 (2020 m.), p. 1419–1440.

[11] Yujia Li ir Dilip B Madan. “Generating realistic stock market order streams with recurrent neural networks”. Iš: *Quantitative Finance* 20.9 (2020 m.), p. 1441–1459.

[12] Zhen Zhang, Stefan Zohren ir Stephen Roberts. “Deep learning for financial time series: A survey”. Iš: *Applied Stochastic Models in Business and Industry* 38.1 (2022 m.), p. 3–24.

[13] Martin Arjovsky, Soumith Chintala ir LÃ©on Bottou. “Wasserstein GAN”. Iš: *International Conference on Machine Learning*. 2017 m., p. 214–223.

[14] Ian Goodfellow. “NIPS 2016 tutorial: Generative adversarial networks”. Iš: *arXiv preprint arXiv:1701.00160* (2016 m.).

[15] Ishaan Gulrajani ir kt. “Improved Training of Wasserstein GANs”. Iš: *Advances in Neural Information Processing Systems*. T. 30. 2017 m.

[16] Alexey Ostrovsky. “Stable Maps of Borel Sets”. Iš: *Topology and its Applications* 326 (2023 m.).

[17] Walter Rudin. *Real and Complex Analysis*. 3-as leid. McGraw-Hill, 1987 m.

[18] Carol Alexander. “Market Risk Analysis, Volume II: Practical Financial Econometrics”. Iš: (2008 m.).

[19] Ã–mer Faruk ErtuÄŸrul ir Mehmet Emin TaÄŸluk. “A Novel Type of Activation Function in Artificial Neural Networks: Trained Activation Function”. Iš: *Neural Computing and Applications*. T. 32. Springer, 2020 m., p. 4533–4543.

[20] Donald L. Cohn. *Measure Theory*. 2nd. BirkhÃ¤user, 2013 m.

[21] Ian Goodfellow, Yoshua Bengio ir Aaron Courville. *Deep Learning*. MIT Press, 2016 m.

[22] Frank Nielsen. “On a Generalization of the Jensen–Shannon Divergence and the Jensen–Shannon Centroid”. Iš: *Entropy* 22.2 (2020 m.), p. 221.

[23] Milena Vuletić, Felix Prenzel ir Mihai Cucuringu. “Fin-GAN: Forecasting and Classifying Financial Time Series via Generative Adversarial Networks”. Iš: *Quantitative Finance* 24.2 (2024 m.), p. 175–199.

[24] Jinsung Yoon, Daniel Jarrett ir Mihaela van der Schaar. “Time-series Generative Adversarial Networks”. Iš: *NeurIPS* (2019 m.).

[25] Filippo Orlandi, Enrico Barbierato ir Alice Gatti. “Enhancing Financial Time Series Prediction with Quantum-Enhanced Synthetic Data Generation: A Case Study on the S&P 500 Using a Quantum Wasserstein Generative Adversarial Network Approach with a Gradient Penalty”. Iš: *Electronics* 13.11 (2024 m.), p. 2158.

[26] Christian M Dahl ir Emil N SÃ¸rensen. “Time Series (Re)Sampling Using Generative Adversarial Networks”. Iš: *Neural Networks* 156 (2022 m.), p. 95–107.

[27] Remigijus Paulavičius ir Julius Žilinskas. “Advantages of Simplicial Partitioning for Lipschitz Optimization Problems with Linear Constraints”. Iš: *Optimization Letters* 10.2 (2016 m.), p. 237–246.

[28] Takeru Miyato ir kt. “Spectral Normalization for Generative Adversarial Networks”. Iš: *International Conference on Learning Representations (ICLR)*. 2018 m.

[29] Stephen Wu ir kt. “Modeling Asynchronous Event Sequences with RNNs”. Iš: *Journal of Biomedical Informatics* 83 (2018 m.), p. 167–177.

[30] Tianyi Cao ir kt. “Quantitative Stock Selection Model Using Graph Learning and a Spatial–Temporal Encoder”. Iš: *Journal of Theoretical and Applied Electronic Commerce Research* 19.3 (2024 m.), p. 1756–1775.

[31] Wei Li ir kt. “Reducing Mode Collapse With Monge-Kantorovich Optimal Transport for Generative Adversarial Networks”. Iš: *IEEE Transactions on Cybernetics* 54.8 (2024 m.), p. 4539–4552.

[32] George E.P. Box ir kt. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 2015 m.

[33] Peter J. Brockwell ir Richard A. Davis. *Introduction to Time Series and Forecasting*. Springer, 2016 m.

[34] Frank J. Massey. “The Kolmogorov-Smirnov test for goodness of fit”. Iš: *Journal of the American Statistical Association* 46.253 (1951 m.), p. 68–78.

[35] Arthur Gretton ir kt. “A kernel two-sample test”. Iš: *Journal of Machine Learning Research* 13.Mar (2012 m.), p. 723–773.

[36] Paul Kupiec. “Techniques for verifying the accuracy of risk measurement models”. Iš: *Journal of Derivatives* 3.2 (1995 m.), p. 73–84.

[37] Sabina Halilbegovic ir Muris Vehabovic. “Backtesting Value at Risk using Historical Simulation method: Evidence from Emerging and Frontier markets”. Iš: *Research in World Economy* 7.2 (2016 m.), p. 48–58.

[38] Thabani Ndlovu ir Delson Chikobvu. “Comparing the riskiness of competing Value-at-Risk models”. Iš: *Journal of Risk and Financial Management* 15.9 (2022 m.), p. 405.

[39] Peter F. Christoffersen. “Evaluating interval forecasts”. Iš: *International Economic Review* 39.4 (1998 m.), p. 841–862.

[40] Dejan Pavlovic ir kt. “Understanding Calibration of Deep Neural Networks: Basics, How-to, and the Impacts of Learning Strategies”. Iš: *Mathematics* 13.1 (2025 m.), p. 25.

[41] Peter F. Christoffersen. “Elements of Financial Risk Management”. Iš: (2012 m.).

[42] CristÃ³bal Esteban, Stephanie L. Hyland ir Gunnar RÃ¤tsch. “Real-valued (medical) time series generation with recurrent conditional GANs”. Iš: *arXiv preprint arXiv:1706.02633* (2017 m.).

[43] Li Shen, Zijin Wei ir Yangzhu Wang. “Determining the Rolling Window Size of Deep Neural Network Based Models on Time Series Forecasting”. Iš: *Journal of Physics: Conference Series* 2078.1 (2021 m.), p. 012011.

[44] Rob J Hyndman ir George Athanasopoulos. *Forecasting: Principles and Practice*. 2-as leid. Online; accessed 2025-11-10. Melbourne: OTexts, 2018 m. URL: `https://otexts.com/fpp2/`.

[45] Shuntaro Takahashi, Yu Chen ir Kumiko Tanaka-Ishii. “Modeling Financial Time-Series with Generative Adversarial Networks”. Iš: *Physica A: Statistical Mechanics and its Applications* 527 (2019 m.), p. 121261.

[46] Ishan Deshpande ir kt. “Max-Sliced Wasserstein Distance and Its Use for GANs”. Iš: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2019 m., p. 10640–10648.

[47] Yiping Gan ir Guo-Ping Liu. “FeatureFuser-LLM: Multi-Scale Feature Fusion with Adaptive Positional Encoding for LLM-Based Time Series Forecasting”. Iš: *2025 4th Conference on Fully Actuated System Theory and Applications (FASTA)*. IEEE. 2025 m., p. 1416–1421.

[48] Bin Chen ir Robert de Jong. “Testing for Structural Change by Isotonic Regression”. Iš: *Econometric Theory* (2025 m.), p. 1–32.

Priedas. WGAN Python programos kodas

P.1 Generatoriaus architektūra

```
import torch
import torch.nn as nn

class Generator(nn.Module):
    def __init__(self, context_len=10, horizon=60, zdim=64):
        super().__init__()
        self.horizon = horizon
        input_dim = zdim + context_len

        self.fc = nn.Sequential(
            nn.Linear(input_dim, 256),
            nn.ReLU(),
            nn.Linear(256, 64 * horizon),
            nn.ReLU(),
        )

        self.cnn = nn.Sequential(
            nn.Conv1d(64, 64, kernel_size=3, padding=1, dilation=1),
            nn.ReLU(),
            nn.Conv1d(64, 64, kernel_size=3, padding=2, dilation=2),
            nn.ReLU(),
            nn.Conv1d(64, 32, kernel_size=3, padding=4, dilation=4),
            nn.ReLU(),
            nn.Conv1d(32, 1, kernel_size=3, padding=1)
        )

    def forward(self, z, context):
```

```
        x = torch.cat([z, context], dim=1)
        h = self.fc(x)
        h = h.view(-1, 64, self.horizon)
        out = self.cnn(h).squeeze(1)
        return out
```

P.2 Diskriminatoriaus architektūra

```
class Critic(nn.Module):
    def __init__(self, seq_len=60):
        super().__init__()

        self.conv = nn.Sequential(
            nn.Conv1d(2, 32, kernel_size=3, padding=1),
            nn.LeakyReLU(0.2),
            nn.Conv1d(32, 64, kernel_size=3, padding=1),
            nn.LeakyReLU(0.2),
        )

        self.gru = nn.GRU(input_size=64, hidden_size=64,
                          num_layers=1, batch_first=True)

        self.fc = nn.Linear(64, 1)

    def forward(self, x):
        h = self.conv(x)
        h = h.permute(0, 2, 1)
        _, h_last = self.gru(h)
        out = self.fc(h_last.squeeze(0))
        return out
```

P.3 Gradiento bausmė

```
def gradient_penalty(critic, real_data, fake_data, lambda_gp=10.0):
    batch_size = real_data.size(0)
    device = real_data.device

    alpha = torch.rand(batch_size, 1, 1, device=device)
    interpolated = alpha * real_data + (1 - alpha) * fake_data
```

```
    interpolated.requires_grad_(True)

    critic_interp = critic(interpolated)

    gradients = torch.autograd.grad(
        outputs=critic_interp,
        inputs=interpolated,
        grad_outputs=torch.ones_like(critic_interp),
        create_graph=True,
        retain_graph=True,
    )[0]

    gradients = gradients.view(batch_size, -1)
    grad_norm = gradients.norm(2, dim=1)
    penalty = ((grad_norm - 1) ** 2).mean()

    return lambda_gp * penalty
```

P.4 Mokymo ciklas

```
def train_wgan(generator, critic, dataloader, n_epochs=2000,
               n_critic=5, lr_g=1e-4, lr_c=1e-4,
               lambda_gp=10.0, zdim=64):
    device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
    generator.to(device)
    critic.to(device)

    opt_G = torch.optim.Adam(generator.parameters(), lr=lr_g,
                             betas=(0.0, 0.9))
    opt_C = torch.optim.Adam(critic.parameters(), lr=lr_c,
                             betas=(0.0, 0.9))

    for epoch in range(n_epochs):
        for real_seqs, context in dataloader:
            batch_size = real_seqs.size(0)
            real_seqs = real_seqs.to(device)
            context = context.to(device)

            # Mokyti kritiką n_critic kartų
            for _ in range(n_critic):
```

```
            opt_C.zero_grad()

            z = torch.randn(batch_size, zdim, device=device)
            fake_seqs = generator(z, context).detach()

            real_2ch = torch.stack([real_seqs, real_seqs.abs()], dim=1)
            fake_2ch = torch.stack([fake_seqs, fake_seqs.abs()], dim=1)

            critic_real = critic(real_2ch).mean()
            critic_fake = critic(fake_2ch).mean()

            gp = gradient_penalty(critic, real_2ch, fake_2ch, lambda_gp)
            loss_C = -critic_real + critic_fake + gp

            loss_C.backward()
            opt_C.step()

        # Mokyti generatorių 1 kartą
        opt_G.zero_grad()

        z = torch.randn(batch_size, zdim, device=device)
        fake_seqs = generator(z, context)
        fake_2ch = torch.stack([fake_seqs, fake_seqs.abs()], dim=1)

        loss_G = -critic(fake_2ch).mean()

        loss_G.backward()
        opt_G.step()

return generator, critic
```