



**KAUNO TECHNOLOGIJOS UNIVERSITETAS
MATEMATIKOS IR GAMTOS MOKSLŲ FAKULTETAS**

Asta Kamarauskaitė

**Chaotinių atraktorių rekonstravimo
kokybės nustatymas laiko eilučių prognozės
uždaviniuose**

Baigiamasis magistro projektas

**Vadovas
doc. dr. K. Lukoševičiūtė**

KAUNAS, 2016



**KAUNO TECHNOLOGIJOS UNIVERSITETAS
MATEMATIKOS IR GAMTOS MOKSLŲ FAKULTETAS**

**Chaotinių atraktorių rekonstravimo
kokybės nustatymas laiko eilučių prognozės
uždaviniuose**

Baigiamasis magistro projektas
Taikomoji matematika (kodas 612G10003)

Vadovas

doc. dr. K. Lukoševičiūtė
2016.05.26

Recenzentas

prof. J. Valantinas
2016.05.26

Projektą atliko

Asta Kamarauskaitė
2016.05.26

KAUNAS, 2016



KAUNO TECHNOLOGIJOS UNIVERSITETAS

Matematikos ir gamtos mokslų fakultetas

Asta Kamarauskaitė

Taikomoji matematika (612G10003)

„Chaotinių atraktorių rekonstravimo kokybės nustatymas laiko eilučių prognozės uždaviniuose“

AKADEMINIO SAŽININGUMO DEKLARACIJA

2016 – 05 – 26 Kaunas

Patvirtinu, kad mano, **Astos Kamarauskaitės**, baigiamasis projektas tema „Chaotinių atraktorių rekonstravimo kokybės nustatymas laiko eilučių prognozės uždaviniuose“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka)

(parašas)

Turinys

Santrauka.....	6
Summary	7
Ižanga.....	9
1. Literatūros apžvalga.....	10
1.1 Chaotinių atraktorių rekonstravimas	10
1.2 Chaotinių atraktorių rekonstravimo parametrų nustatymas.....	11
2. Medžiagos ir tyrimų metodai	14
2.1 Artimiausio klaidingo kaimyno metodas.....	14
2.2 Koreliacinės dimensijos metodas	15
2.3 Tarpusavio informacijos metodas.....	16
2.3.1 Fraser algoritmas.....	18
2.4 Neuroninis NARX modelis	21
2.5 Statistiniai kriterijai	25
2.5.1 Stjudento t kriterijus.....	25
2.5.2 Šapiro – Wilk kriterijus.....	26
2.5.3 Jarque – Bera kriterijus	26
3. Tyrimų rezultatai ir jų aptarimas.....	27
3.1 Lorencio atraktorius.....	27
3.1.1 Lorencio atraktoriaus rekonstravimo dimensijos nustatymas	28
3.1.2 Lorencio atraktoriaus laiko vėlinimo nustatymas	30
3.1.3 Lorencio atraktoriaus prognozavimas su neuroniniu NARX tinklu	32
3.2 Sumodeliuota laiko eilutė	38
3.2.1 Sumodeliuotos laiko eilutės rekonstravimo dimensijos nustatymas.....	38
3.2.2 Sumodeliuotos laiko eilutės laiko vėlinimo nustatymas	40

3.2.3	Sumodeliuotos laiko eilutės prognozavimas su neuroniniu NARX tinklu	42
3.3	HP kompanijos akcijų kainų laiko eilutė	46
3.3.1	HP kompanijos akcijų kainų rekonstravimo dimensijos nustatymas	47
3.3.2	HP kompanijos akcijų kainų laiko vėlinimo nustatymas	47
3.3.3	HP kompanijos akcijų kainų prognozavimas suneuroniniu NARX tinklu	49
	Išvados	52
	Literatūros sąrašas	53
	Priedai	54
1	priedas. Sumodeliuotos laiko eilutės neuroninio NARX tinklo rezultatai	54
2	priedas. HP akcijos kainų laiko eilutės neuroninio NARX tinklo rezultatai	56
3	priedas. HP kompanijos akcijų kainų prognozė	58
4	priedas. Programos kodai	59
	Laiko eilučių generavimas:	59
	Rekonstravimo dimensijos nustatymas:	60
	Tarpusavio informacijos metodas:	61
	Neuroninio NARX modelio sudarymas ir laiko eilutės prognozė:	62

Kamrauskaitė, Asta. Chaotinių atraktorių rekonstravimo kokybės nustatymas laiko eilučių prognozavimo uždaviniuose. *Magistro baigiamasis projektas / vadovas doc. dr. Kristina Lukoševičiūtė*; Kauno technologijos universitetas, Matematikos ir gamtos mokslų fakultetas.

Mokslo kryptis ir sritis: Taikomoji matematika

Reikšminiai žodžiai: *chaotinis, rekonstravimas, dimensija, laiko vėlinimas, neuroninis tinklas.*

Kaunas, 2016. 63 p.

Santrauka

Tiriant bet kokią chaotinę laiko eilutę yra svarbu ją rekonstruoti fazinėje erdvėje, kad sėkmingai būtų atliktas rekonstravimas užtenka turėti vieno kintamojo laiko eilutę ir identifikuoti rekonstravimo parametrus: rekonstravimo dimensiją ir laiko vėlinimo parametą. Mažiausios rekonstravimo dimensijos nustatymas buvo atliktas dviem metodais: artimiausio klaidingo kaimyno ir koreliacinės dimensijos metodu. Laiko vėlinimo parametras buvo nustatomas tarpusavio informacijos metodu. Jeigu laiko eilutė yra prognozuojama tiksliai, tuomet ir rekonstravimas bus atliktas korektiškai, todėl panaudojus neuroninį NARX tinklą, kiekviena laiko eilutė buvo prognozuojama ir tikrinama jos prognozės kokybė.

Lorenco atraktoriaus mažiausia rekonstravimo dimensija buvo nustatyta tiksliau artimiausio klaidingo kaimyno metodu nei koreliacinės dimensijos metodu. Tarpusavio informacijos metodu nustatytu laiko vėlinimo parametru atlikta laiko eilutės prognozė neuroniniu tinklu pakankamai tiksliai tik pirmiems 16 taškų, toliau prognozuojant paklaidos yra didelės.

Sumodeliuotos laiko eilutės, kurios komponentės yra skirtingų fazių ir amplitudžių harmonikos, mažiausia rekonstravimo dimensija taip pat nustatyta geriau artimiausio klaidingo kaimyno metodu, nei koreliacinės dimensijos metodu. Su tarpusavio informacijos metodu nustatytu laiko vėlinimo parametru atlikta prognozė neuroniniu tinklu pakankamai gerai 50 taškų.

HP kompanijos akcijų kainų mažiausia rekonstravimo dimensija buvo nustatyta tiek artimiausio klaidingo kaimyno, tiek koreliacinės dimensijos metodu, tačiau tiksliau nustatyta koreliacinės dimensijos metodu. Laiko vėlinimo parametras nustatytas tarpusavio informacijos metodu pakankamai didelis, tačiau tokie rezultatai yra tikėtini, kai dirbama su finansiniais

duomenimis. Šios laiko eilutės prognozė atlikta pakankamai korektiškai, prognozuojant 25 taškus į priekį neuroniniu NARX tinklu.

Taigi, visoms laiko eilutėms buvo nustatyti rekonstravimo parametrai, o rekonstravimo kokybė patikrinta atlikus eilučių prognozes neuroniniu tinklu bei įvertinant statistines hipotezes apie gautas paklaidas.

Summary

Kamrauskaitė, Asta. Quality Determination of Chaotic Time Series Reconstruction in Time Series Forecasting. *Master's thesis in Applied Mathematics / supervisor doc. dr. Kristina Lukoševičiūtė. Faculty of Mathematics and Natural Sciences. Kaunas University of Technology.*

Research area and field: Physical sciences and Mathematics.

Key words: *chaotic, reconstruction, dimension, time delay, neural network.*

Kaunas, 2016. 63 p.

Analyzing chaotic time series, it is important to reconstruct it in a phase space for successful reconstruction. Also, it is enough to have time series of one-variable and reconstruction parameters: minimum reconstruction dimension and time delay. For identifying minimum reconstruction dimension there were used two methods: false nearest neighbor method and correlation dimension method. Time delay was identified by the method of mutual information. Moreover, if forecast of time series is made correctly, then reconstruction of time series is also adequate, so for this purpose all-time series were forecasting by neural NARX network and then quality of errors was checked by statistical hypothesis.

Minimum reconstruction dimension for Lorenz attractor was better identified by false nearest neighbor method then by correlation dimension method. Time lag was identified by mutual information method and then Lorenz attractor forecast was done correctly for next 16 points by trained neural NARX network.

Furthermore, for modeled time series with different phase and amplitude minimum embedding dimension was identified better by false nearest neighbor method. When time lag was identified by mutual information method, this time series forecast was made adequate for 50 points.

HP company action prices are minimum embedding dimension was identified better by correlation dimension method. Identified time lag by mutual information method was quite big, but these results are expected when we work with financial time series. For this time series the forecast was mode correct for 25 points ahead by neural NARX network.

All in all, all-time series were analyzed, the reconstruction parameters were identified and the reconstruction quality was checked by forecasting time series with neural NARX method and by estimating statistical hypothesis of forecasting errors.

Ižanga

Norint iširti bet kokią chaotinę eilutę yra svarbu šią laiko eilutę rekonstruoti fazinėje erdvėje. Visa tai reikalinga, kad būtų aiškus dinaminis išsidėstymas daugiamatėje erdvėje, be to rekonstruojant vieno kintamojo eilutę yra išlaikoma jo dinamika ir nėra būtina, atliekant prognozę ar kitus tyrimus, žinoti, kitų kintamųjų. Norint teisingai atlikti šią rekonstrukciją yra tikslinga žinoti analizuojamą eilutę bei rekonstravimo parametrus.

Šiame baigiamajame magistro darbe – pagrindinis tikslas buvo iširti skirtingas laiko eilutes, prognozuojant neuroniniu NARX tinklu, turint vieno kintamojo eilutę. Darbe analizuojamos trys skirtingos laiko eilutės – keistųjų atraktorių grupei priklausantis Lorenco atraktorius, sumodeliuota laiko eilutė, kurią sudaro skirtingų fazių ir amplitudžių harmoninių funkcijų suma, ir realūs kompanijos HP akcijos kainų duomenys. Uždaviniai iškelti vienodi visoms trimis analizuojamos eilutėms: nustatyti mažiausią rekonstravimo dimensiją, laiko vėlinimo parametą, atlikti laiko eilutės prognozę neuroniniu NARX tinklu bei iširti prognozavimo kokybę.

Tyrimai atlikti iš pradžių nustatant mažiausią laiko eilučių rekonstravimo dimensiją dviem metodais: artimiausio klaidingo kaimyno ir koreliacinės dimensijos metodu. Tarpusavio informacijos metodu nustatomas laiko vėlinimo parametras ir su nustatytoju parametru atliktas neuroninio tinklo modelio sudarymas bei laiko eilutės prognozė. Sudarytojo modelio korektiškumas vertinamas analizuojant modelio paklaidas, paklaidų koreliaciją, tikrinamos Stjudento, Šapiro – Wilk ir Jarque – Bera kriterijai.

Chaotinių atraktorių rekonstravimo tyrimas atliktasnaudojantis MATLAB programine įranga. Dėka šios programinės įrangos buvo galima lengvai atlikti skaičiavimus bei pakankamai aiškiaijuosgrafiškai realizuoti.

1. Literatūros apžvalga

Laiko eilutės – tai periodiškų stebėjimų visuma, kurių duomenys tai tam tikrais laiko momentais fiksuoti stebėjimų dydžiai, arba per periodą stebimų dydžių suma. Laiko eilutės gali būti suformuotos iš įvairaus dažnumo, tačiau vienodo periodiškumo duomenų: valandinių, kasdienių, savaitinių, mėnesinių, metinių ir pan.

Dinaminės sistemos, sudarytosios iš laiko eilučių, yra visur aplink natūralioje gamtoje [3], pavyzdžiui tornado judėjimai, oro sąlygos, sūkurių susikonstravimai arba finansuose – akcijų kainos. Kiekvienu atveju analizavimo funkcijos yra skirtingos, priklausomai nuo nagrinėjamo objekto, jo unikalių ypatybių. Tornado judėjimo trajektorijos tyrimas yra svarbus, kad sumažinti šios stichinės nelaimės nuostolius, akcijų kainų tendencijos yra svarbu žinoti investuotojams ir akcininkams, norint pasiekti geriausią pelną. Bet kuriuo atveju analizuojant chaotinę laiko eilutę yra tikslinga ją rekonstruoti, nes rekonstruojant vieno kintamojo laiko eilutę yra išlaikoma unikali dinamika. Tai labai svarbus faktas, nes nėra svarbūs visi veiksniai lemiantys eilutės judėjimo trajektorijas ir iš vieno eilutės duomenų yra atkuriamą dinamiką. Norint optimaliai atlikti rekonstravimą svarbu žinoti reikalingus rekonstravimo parametrus.

1.1 Chaotinių atraktorių rekonstravimas

Svarbiausias žingsnis, bet kokiai chaoso identifikavimo technikai, yra laiko eilutės rekonstravimas fazinėje erdvėje. Toks rekonstravimas naudojamas vieno kintamojo eilutę pavaizduojant daugiamatėje erdvėje, kad būtų aiškus jo dinaminis išsidėstymas. Packard [1] pasiūlė metodą, rekonstruojantį laiko eilutę į fazinę erdvę naudojant laiko vėlinimą, tai yra konstruojant laiko vėlinimo vektorių. Rekonstravimo trajektorija Y gali būti apibrėžta kaip matrica, kur kiekviena eilutė yra fazinės erdvės vektorius:

$$Y = [Y_1, Y_2, \dots, Y_N]^T, \quad (1)$$

čia Y_i sistemos būseną laiko momentu i .

Egzistuoja ryšys tarp laiko eilutės x_i ($i = \overline{1, n}$) elementų ir rekonstruoto vektoriaus elemento Y_i :

$$Y_i = [x_i, x_{i+\tau}, \dots, x_{i+(d-1)\tau}]. \quad (2)$$

Šis vektorius konstruoja d - dimensinę rekonstravimo fazinę erdvę, kurioje duoda geriausią artimiausių trajektorijų atskyrimą kiek įmanoma mažesnėje rekonstravimo erdvėje.

Whitney [1] taip pat buvo pateikęs rekonstravimo teoremą (1936), kurioje teigiama, jog d - dimensinė erdvė gali būti patalpinta į $2d + 1$ koordinačių sistemą. 1991m. Sauer pritaikė pastarąją teoremą fraktaliniams atraktoriams su dimensija D_F ir parodė, kad rekonstravimo erdvė turi turėti didesnę nei $2D_F$ dimensiją. Tai buvo naudojama skaičiuojant galingumo dimensiją, arba tiesiog nustatinėjant fraktalų dimensijas, bet ne fraktalines dimensijas laiko eilutėms.

Takens (1981) [2] įrodė ir patvirtino jau anksčiau Packard pasiūlymą, kad pakankama rekonstravimo dimensija, norint rekonstruoti fazinę erdvę, yra $2D + 1$, kur D yra atraktoriaus sistemos dimensija, ir šis atvaizdavimas yra pasiekiamas vėlinant stebėjimų koordinates. Takens teorema sako, kad pakankama rekonstravimo dimensijos sąlyga yra, jog daugeliui sistemų rekonstravimas gali būti pasiektas keliose dimensijose, be to norint atlikti rekonstrukciją užtenka turėti vieno kintamojo eilutę.

Tegu laiko vėlinimai būna pažymimi, kaip $\tau_1, \tau_2, \dots, \tau_{d-1}$, tuomet sistemos vektorius laiko momentu t yra apibrėžiamas:

$$\vec{x}(t) = [x(t), x(t - \tau_1), \dots, x(t - \tau_{d-1})], \quad (3)$$

čia d yra rekonstruotos erdvės dimensijos dydis.

1.2 Chaotinių atraktorių rekonstravimo parametrų nustatymas

Chaotinių atraktorių rekonstravimui reikia žinoti du parametrus – rekonstravimo dimensiją ir laiko vėlinimo parametą. Literatūroje galima aptikti daugybę metodų ir algoritmų šių parametrų vertinimui, kadangi šiame darbe analizuojamos buvo itin chaotinės laiko eilutės, apie kurių dinamiką nebuvo iš anksto žinoma, todėl parinkti metodai tyrimui, kurie toleruoja chaotines laiko eilutes ir geba apdoroti jų duomenis.

Dažniausiai literatūroje minimi du populiariausi metodai rekonstravimo dimensijai nustatyti – artimiausio klaidingo kaimyno ir koreliacinės dimensijos metodai [4, 5]. Abu šie metodai yra skirti dirbti su netiesinėmis dinaminėmis sistemomis, skirtumas, kad finansiniuose uždaviniuose daugiau dirbama su artimiausio klaidingo kaimyno metodu, o medicinoje – koreliacinės dimensijos metodu.

Taip pat, artimiausio klaidingo kaimyno metodas puikiai dirba su aukštesnės dimensijos laiko eilutėmis (didesnėmis nei 3), tuo tarpu koreliacinės dimensijos metodas atvirkščiai – geriau apdoroja duomenis su dimensija 3 ir mažesne.

Artimiausio klaidingo kaimyno metodu teoriškai yra sakoma, kai klaidingų kaimynų skaičius pasiekia nulį, tai atitinkama rekonstravimo dimensija yra mažiausia. Analizuojant finansinius duomenis dažnai pasitaiko, kad nulis nėra pasiekiamas [6], tuomet yra nustatoma viršutinė riba, kuomet analizuojama laiko eilutė yra išskleista, dažniausiai naudojama 1% riba (Kugiumtzis 1994m. rekomendacija).

Koreliacinės dimensijos metodas, kuris naudoja fraktalinę dimensiją, greitai atlieka skaičiavimus, lengvai vertinami gauti rezultatai. Šiuo metodu rekomenduojama atlikti skaičiavimus su mažesnėmis nei 32 000 taškų laiko eilutėmis.

Kitas rekonstravimo parametras, laiko vėlinimas τ , turi būti pasirenkamas labai atidžiai [3], nes per mažas τ daro įtaką rekonstravimui, rekonstruota laiko eilutė neperteikia dinamikos ir yra išsidėsčiusi įstrižainėje. Priešingu atveju, pasirinkus per didelį τ , rekonstruota laiko eilutė praranda dinamiką ir su originalia laiko eilute siejama tik atsitiktiniu tolygiu ryšiu. Vienas iš optimaliausių metodų identifikuoti korektišką laiko vėlinimą yra tarpusavio informacijos metodas, kurio tikslas apskaičiuoti tikimybinį pasiskirstymą bei skaičiuojant tarpusavio informacijos parametro vertę, stebėti kaip ši kinta, kai kinta laiko vėlinimas τ . Fraser ir Swinney pasiūlė, kad netiesinis tarpusavio informacijos parametras geriausiai nustato optimalų laiko vėlinimą, tai yra kai tarpusavio informacijos parametras pasiekia pirmąjį minimumą, atitinkamas laiko vėlinimas yra optimaliausias rekonstruojant laiko eilutę. Remiantis šiuo pastebėjimu buvo atliktas ir šio baigiamojo magistro darbo laiko vėlinimo tyrimas, visos trys analizuojamos laiko eilutės ištirtos tarpusavio informacijos metodu. Kiekvienu atveju buvo tikrinama kaip atrodo rekonstruotos laiko eilutės imant skirtingus laiko vėlinimus τ – geriausias rekonstravimas atliktas buvo imant visais atvejais tarpusavio informacijos pirmąjį minimumą, kitais atvejais laiko eilutės praranda dinamiką, tampa neinformatyvios.

Lorenco atraktoriaus rekonstravimas, nustatant laiko vėlinimo parametą, gerai iliustruoja kas būna kai pasirenkamas netinkamas parametras τ [3]. Pasirinkus per mažą τ atraktorius išsidėsto į tiesę, pasirenkant pagal tarpusavio informacijos metodą, naudojantis Fraser ir Swinney pastebėjimu, gautas optimaliausias laiko vėlinimas, o pasirinkus per didelį τ – gauta laiko eilutė yra praradusi

dinamiką. Kiek kitoks gautas rezultatas gali būti pasiektas sumodeliuotai laiko eilutei, kuri sudaryta iš skirtingų fazių ir amplitudžių harmoninės funkcijos. Šiuo atveju imant antrąjį tarpusavio informacijos parametro minimumą laiko eilutė gali atrodyti visiškai išskleista, toks nukrypimas gali atsirasti dėl to, kad analizuojama laiko eilutė yra periodinė. Dirbant su finansiniais duomenimis yra susiduriama su problema, kad tarpusavio informacijos metodas jautriai reaguoja į itin chaotinius duomenis ir nėra identifikuojamas laiko vėlinimo parametras – gautas rezultatas gali būti arba su bet kuriuo laiko vėlinimo parametru tarpusavio informacijos parametras lygus 0, arba tarpusavio informacijos parametras yra tolydžiai mažėjantis, to pasėkoje minimumas yra sunkiai identifikuojamas. Tokiu atveju, naudojami papildomi metodai, kurie padeda pašalinti sezoniškumą iš analizuojamų duomenų ir tuomet jau optimalūs laiko vėlinimai yra identifikuojami, taip pat reikia atkreipti dėmesį, jog dirbant su realiais duomenimis laiko vėlinimo parametras nustatomas ir pakankamai didelis (gali būti lygus ir didesnis už 40).

Nustačius rekonstravimo parametrus yra tikslinga patikrinti rekonstravimo kokybę, ypač kai dirbama su realiais duomenimis, kai nėra žinoma šių laiko eilučių dinamika. Vienas iš paprasčiausių būdų kokybę nustatyti, tai atlikti laiko eilutės prognozę. Jeigu laiko eilutės prognozė su rekonstravimo parametrais yra kiek norima tiksli, tai rekonstravimas bus atliktas teisingai. 1980 metais įvyko laiko eilučių prognozavimo atgimimas, nes pavyko išsiaiškinti, kad neuroninis tinklas gali būti naudojamas laiko eilučių prognozei. Neuroninis tinklas [7] prognozuojant yra labai patogus, kadangi šiam metodui nereikia žinoti visos informacijos apie analizuojamą laiko eilutę, tai yra nereikalinga informacija, kurios dėka ji yra sugeneruojama. Tačiau šis metodas yra labai sudėtingas naudoti, pavyzdžiui dirbant su finansiniais duomenimis (akcijų kainomis) gali reikėti panaikinti sezoniškumą, kad prognozuojant būtų išvengta keblumų. Pakankamai diskutuotinas klausimas yra kokio ilgio laiko eilutę reikia pasirinkti, nes ne visada turint daug duomenų neuroninis tinklas tikslingai atlieka prognozę. Netiesinis autoregresijos modelis su egzogenine išvestimi (NARX), kuris ir buvo naudojamas šiame darbe, labai gerai dirba ir su ilgomis laiko eilutėmis bei skirtingo tipo eilutėmis. Eugen Diaconescu [7] pademonstravo, kaip prognozuojamos yra Mackey – Glass eilutė, fraktalinė laiko eilutė bei akcijų kainų laiko eilutės. Vadinasi, ta pati taikoma metodika gali tikti skirtingo tipo laiko eilutėms analizuoti.

Taigi, literatūroje yra plačiai naudojami visi metodai, kurie buvo naudojami šiame baigiamajame darbe, bei naudojimo spektras yra platus. Nustatant rekonstravimo dimensiją naudojami arba artimiausio klaidingo kaimyno, arba koreliacinės dimensijos metodai. Laiko

vėlinimo parametą universaliausia nustatyti tarpusavio informacijos metodu. Pasirinktos 3 laiko eilutės analizuoti, iš kurių tik vienos yra žinomas dinaminis judėjimas erdvėje, todėl prognozavimui pasirinktas neuroninis tinklas NARX, kuriam nereikia žinoti visų laiko eilučių savybių.

2. Medžiagos ir tyrimų metodai

2.1 Artimiausio klaidingo kaimyno metodas

Artimiausio klaidingo kaimyno metodas [8, 9] yra geriausias metodas norint nustatyti mažiausią laiko eilutės rekonstravimo dimensiją. Svarbiausias šios metodo tikslas yra stebėti artimiausių kaimynų elgesį kintant dimensijai, kai $d \rightarrow d + 1$. Kai artimiausių kaimynų skaičius artėja į nulį, tai yra manoma, kad laiko eilutė yra visiškai išskleista šioje dimensijoje.

Pirmiausia, turint d dimensiją, pažymime r -tąjį $y(n)$ artimiausią kaimyną $y^r(n)$. Tuomet Euklidinio atstumo kvadratas taškui $y(n)$ ir jo kaimynas atrodo taip:

$$R_d^2(n, r) = \sum_{k=0}^{d-1} (x(n + kT) - x^r(n + kT))^2. \quad (4)$$

Pereinant iš d dimensijos į $d + 1$ dimensiją, naudojant laiko vėlinimą, yra pridama $(d + 1)$ -oji koordinatė kiekvienam $y(n)$. Ši naujoji koordinatė atrodo taip: $x(n + Td)$. Po koordinatės pridėjimo atstumas, tarp $y(n)$ ir to paties r -tojo artimiausio kaimyno, d dimensijoje yra:

$$R_{d+1}^2(n, r) = R_d^2(n, r) + (x(n + dT) - x^r(n + dT))^2. \quad (5)$$

Natūralu, kad ieškant rekonstravimo paklaidų, atstumo tarp $y(n)$ ir $y^r(n)$ padidėjimas, pereinant iš d dimensijos į $d + 1$, yra pakankamai didelis. Šiam tikslui įvertinti yra nustatytas pirmasis kriterijus:

$$\sqrt{\frac{(R_{d+1}^2(n, r) - R_d^2(n, r))}{(R_d^2(n, r))}} = \frac{|x(n + Td) - x^r(n + Td)|}{(R_d(n, r))} > R_{tol}, \quad (6)$$

Čia R_{tol} yra nustatyta reikšmė, dažniausiai $R_{tol} \geq 15$.

Jeigu $y(n)$ artimiausias kaimynas nėra arti $R_d(n) \approx R_A$ ir jis yra klaidingas, tuomet $R_{d+1}(n)$ atstumas, gautas pridant $(d + 1)$ -ąją komponentę prie duomenų vektoriaus, bus $R_d(n) \approx 2R_A$. Taip bus jei artimiausias kaimynas tęstųsi iki laiko eilutės ekstremumų, kai jis yra išskleistas, ir jei šis kaimynas bus klaidingas. Tokiu atveju yra sudarytas antrasis kriterijus:

$$\frac{(R_{d+1}(n))}{R_A} > A_{tol}, \quad (7)$$

čia A_{tol} nustatyta reikšmė, dažniausiai $A_{tol} \geq 2$.

Renkantis R_{tol} ir A_{tol} reikšmes, yra tikslinga remtis Abarbanel(1996) rekomendacijomis, tai yra $R_{tol} = 15$, $A_{tol} = 2$.

2.2 Koreliacinės dimensijos metodas

Koreliacijos dimensija [10] d - tai parametras apibūdinantis netiesinę dinaminę sistemą, kurio dėka galima pasakyti kiek nepriklausomų diferencialinių lygčių sistemai nusakyti. Šios dimensijos nusakymui naudojamas toks algoritmas:

- Tegu turime laiko eilutę x_1, x_2, \dots, x_n .
- Konstruojami vektoriai \vec{x}_i , kurio komponentės yra išrenkamos iš laiko eilutės duomenų, kai yra žinoma rekonstravimo dimensija d ir laiko vėlinimas τ (laiko tarpas nuo i iki $i + \tau$):

$$\vec{x}_i = (x_i, x_{i+\tau}, x_{i+2\tau}, \dots, x_{i+(d-1)\tau}) \quad (8)$$

- Apskaičiuojame koreliacijos sumą:

$$C^d(R) = \frac{1}{N(N-1)} \sum_{i,j=1, i \neq j}^N \theta(R - |\vec{x}_i - \vec{x}_j|), \quad (9)$$

čia N - imamų laiko eilutės duomenų skaičius, d - dimensija

Heavisaido funkcija $\theta(x) = \begin{cases} 0, & \text{jei } x < 0 \\ 1, & \text{jei } x \geq 0 \end{cases}$

$|\vec{x}_i - \vec{x}_j|$ - vektorių skirtumo modulis, šiuo atveju nurodantis maksimalią koordinatės distanciją. Kitaip sakant randame didžiausią skirtumą tarp komponentių ir šį dydį naudojame kaip vektorių skirtumo ilgį: $|\vec{x}_i - \vec{x}_j| = \text{Max}_k |\overrightarrow{x_{i+k\tau}} - \overrightarrow{x_{j+k\tau}}|$

- Koreliacijos dimensija randama iš sąryšio:

$$C^d(R) = R^{D_c(d)} \quad (10)$$

- Tuomet koreliacijos dimensija:

$$D_c(d) = \frac{\log C^d(R)}{\log R} \quad (11)$$

2.3 Tarpusavio informacijos metodas

Tarpusavio informacijos (MI) algoritmas [11] yra plačiai naudojamas nustatant vienus svarbiausius parametrus – laiko vėlinimus, kurie naudojami rekonstruoti netiesines laiko eilutes fazinėje erdvėje, koreliaciją tarp dviejų laiko eilučių, parodant priežastinį ryšį bei nurodant kuri laiko eilutė daro įtaką kitai. Tegu $\{s(t_i)\}, i = \overline{(1, N)}$ yra netiesinė laiko eilutė, gaunama eksperimentiškai ir patenka į intervalą Δt . Turima laiko eilutė gali būti rekonstruojama tokiu būdu:

$$\begin{aligned} &(s(t_1), s(t_1 + \tau), \dots, s(t_1 + (d - 1)\tau)) \\ &(s(t_2), s(t_2 + \tau), \dots, s(t_2 + (d - 1)\tau)) \\ &\dots \dots \\ &(s(t_N), s(t_N + \tau), \dots, s(t_N + (d - 1)\tau)), \end{aligned} \tag{12}$$

čia τ parodo laiko vėlinimo parametro dydį.

Tarpusavio informacijos parametras tarp:

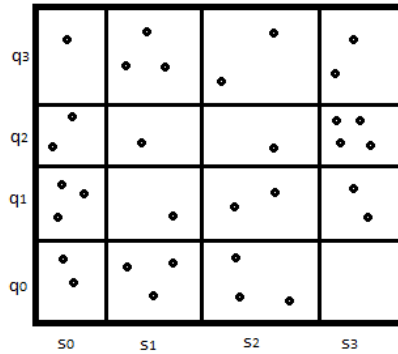
$$S = \{s(t_1), s(t_2), \dots, s(t_N)\} \text{ ir } Q = \{s(t_1 + \tau), s(t_2 + \tau), \dots, s(t_N + \tau)\}, \tag{13}$$

čia $I(S, Q)$ - tai yra vidutinis dažnis, kur S gali būti prognozuojamas pagal išmatavimą Q :

$$I(S, Q) = H(Q) + H(S) - H(S, Q), \tag{14}$$

čia $H(Q)$ ir $H(S)$ yra Q ir S atitinkama entropija, o $H(S, Q)$ yra bendroji entropija tarp S ir Q . Kuomet MI įgauna pirmąjį minimumą, yra laikoma, kad tuo laiko momentu laiko vėlinimo gautasis parametras yra optimaliausias, norint atlikti laiko eilutės rekonstravimą.

Norint pavaizduoti MI parametą tarp dviejų laiko eilučių, imamos skaitinės vertės tarp S ir Q , kurios atitinka taškų skaičių tarp dviejų statmenų ašių. Plokštuma SQ taip pat įtraukiama į šį procesą ir duomenų poros iš S ir Q atitinka skaičius $q_i, i = 1, 2, \dots$. Žemiau pateikiamas eskizinis šio proceso planas:



2.1 pav. Dvimatės erdvės dalijimas, pagal vienodas tikimybes

Taškų tikimybinis pasiskirstymas, atitinkantis duomenų poras q_i , žymimas $P(q_i)$ ir nustatomas dalinant turimą laiko eilutę iš turimų taškų skaičiaus N . Taip pat ašis S yra tokio pat ilgio kaip ir ašis Q , o pozicija tarp s_i ir q_i išreiškiama (s_i, q_i) ir $P(s_i, q_i)$ yra proporcija tarp taškų (s_i, q_i) ir taškų kiekio N . Be to MI parametras gali būti skaičiuojamas remiantis aukščiau pateiktu planu bei naudojanti šiomis formulėmis:

$$H(Q) = - \sum P(q_i) \log_2 P(q_i) \quad (15)$$

$$H(S) = - \sum P(s_i) \log_2 P(s_i) \quad (16)$$

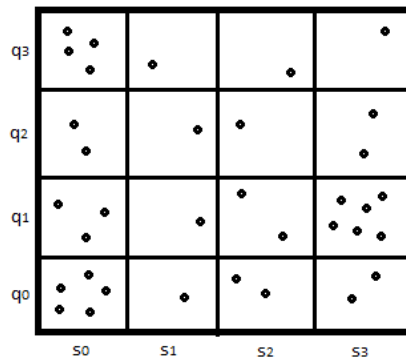
$$H(Q, S) = H(S, Q) = - \sum_{i,j} P(s_i, q_i) \log_2 P(s_i, q_i), \quad (17)$$

čia $P(s_i, q_i)$ yra sekcijos s_i tikimybinis pasiskirstymas.

Tuomet MI parametras yra apskaičiuojamas:

$$I(Q, S) = \sum_i \sum_j P_{sq}(s_i, q_i) \log_2 \frac{P_{sq}(s_i, q_i)}{P(s_i)P(q_i)}. \quad (18)$$

Dvimatė erdvė gali būti dalinama ne tik į vienodos tikimybės kvadratėlius, bet ir dalinant į vienodo dydžio kvadratėlius, neapibrėžiant tikslaus taškų, papuolančių į atitinkamus kvadratėlius, skaičiaus:



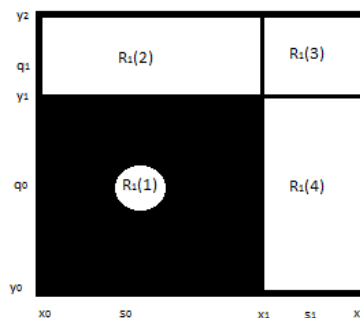
2.2 pav. Dvimatės erdvės dalijimas į vienodo dydžio kvadratėlius

2.3.1 Fraser algoritmas

Panaudojant Fraser algoritmą [12] galima lengvai vertinti tarpusavio informacijos parametą, tad šiame skyriuje trumpai bus pademonstruota kaip pastarasis algoritmas pritaikomas dvimatėje erdvėje, analizuojant laiko eilutės taškus, jų tikimybinus pasiskirstymus.

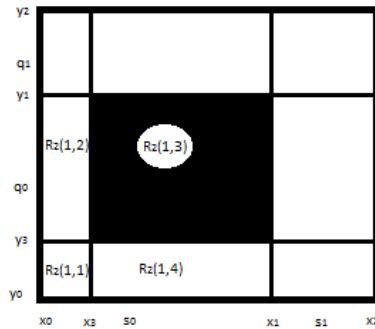
Duotos dvi laiko eilutės $\{X\} = \{x_1, x_2, \dots, a_n\}$ ir $\{Y\} = \{y_1, y_2, \dots, y_{en}\}$, jų tarpusavio informacijos parametras $I(X, Y)$, tai vidutinis dažnio skaičius, kai x_i iš $\{X\}$ gali būti prognozuojamas pagal y_{es} iš $\{Y\}$ tikimybinio dydžio. Be to tarp turimų laiko eilučių ryšys yra simetrinis, t.y. $I(X, Y) = I(Y, X)$ ir taškai (a_n, y_{en}) priklauso erdvei XY .

Pirmasis Fraser algoritmo žingsnis, parodytas žemiau 3 – jame paveikslėlyje, kuomet XY dvimatė erdvė dalinama į keturias dalis, su vienodomis tikimybėmis pagal X ir Y ašis (padalintos dalys žymimos atitinkamai $R_1(1)$, $R_1(2)$, $R_1(3)$ ir $R_1(4)$):



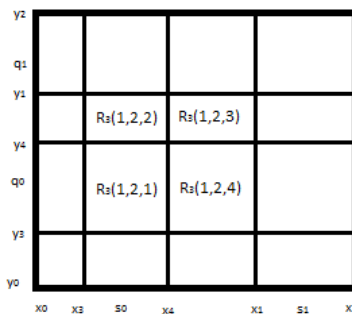
1.3 pav. Pirmasis XY dalijimas į vienodo tikimybinio dydžio 4 dalis

Toliau atliekamas didžiausios dalies $R_1(1)$ dalinimas, tokiu pat būdu – į keturias dalis su vienodomis tikimybėmis.



2.4 pav. $R_1(1)$ dalijimas į vienodo tikimybinio dydžio 4 dalis

Tuomet taškų skaičius $N_{x_0-x_1}$ tarp x_0 ir x_3 yra toks pats kaip ir skaičius $N_{x_3-x_1}$ tarp x_3 ir x_1 , bet per pusę mažesnis negu skaičius $N_{x_1-x_2}$. Taigi, tikimybinis pasiskirstymas tarp taškų x_1 ir x_3 gali būti apibrėžiamas kaip $N_{x_3-x_1}/N_0$, čia N_0 bendrasis visų XY taškų skaičius. Taip pat dalių santykiai atitinkamai yra tokie patys ir y ašiai. Iš 4 paveikslo toliau renkamosi dalinti $R_2(1,3)$ tokiu pat būdu, kaip ir ankstesniuose žingsniuose:



2.5 pav. $R_2(1, 3)$ dalijimas į vienodo tikimybinio dydžio 4 dalis

Iš aukščiau pateikto paveikslėlio yra matoma, kad $R_2(1,3)$ dalis padalinta į keturias naujasias – $R_3(1,2,1)$, $R_3(1,2,2)$, $R_3(1,2,3)$ ir $R_3(1,2,4)$. Toliau tęsiant šį dalijimo procesą mažesniųjų dalių medis išauga ir galima MI parametro realizacija.

Panaudojus dvimatės erdvės XY dalijimo procesą Tarpusavio informacijos parametras tarp $\{X\}$ ir $\{Y\}$ gali būti apskaičiuojamas remiantis šia formule:

$$I(X, Y) = \left(\frac{1}{N_0}\right) F(R_0(Z_0)) - \log N_0, \tag{19}$$

čia N_0 pažymi bendrą XY taškų skaičių, $F(R_0(Z_0))$ - tai kvadratėlio $R_0(Z_0)$ MI parametras. Jeigu nebeturime priklausomybių turimame kvadratėlyje $A_{rk}(S_{ki})$, tai $F(A_{rk}(S_{ki}))$ galima apskaičiuoti:

$$F(R_k(Z_k)) = N(R_k(Z_k)) \log_2 \left(N(R_k(Z_k)) \right), \quad (20)$$

čia $N(A_{rk}(S_{ki}))$ yra kvadratėlio $A_{rk}(S_{ki})$ taškų skaičius, k – erdvės XY sluoksnis. Jeigu vis dėl to yra matomos priklausomybės kvadratėlyje $A_{rk}(S_{ki})$, tuomet tolimesnis dalijimas yra modifikuojant ankstesniąją lygtį:

$$F(R_k(Z_k)) = N(R_k(Z_k)) \log_2 4 + \sum_{i=1}^4 F(R_{k+1}(Z_k, i)), \quad (21)$$

čia $R_{k+1}(Z_k, i)$ yra $R_k(Z_k)$ pokvadratis. Tuomet belieka nuspręsti kuomet reikia nustoti atlikti dalijimus ir ištirti $R_k(Z_k)$ struktūrą. Remiantis Fraser ir Swinney kriterijumi, χ^2 panaudojamas ištirti $k+1$ ir $k+2$ sluoksnius iš $R_k(Z_k)$. Manoma, kad $R_k(Z_k)$ taškai turi tenkinti sąlygas:

$$\chi_3^2 < 1,547, \chi_{15}^2 < 1,287, \quad (22)$$

$$(23)$$

$$\chi_3^2 = \frac{16}{9} \left(\frac{1}{N(R_k(Z_k))} \right) \sum_{i=1}^4 \left(N(R_{k+1}(Z_k, i)) - \frac{N(R_k(Z_k))}{4} \right)^2$$

$$(24)$$

$$\chi_{15}^2 = \frac{256}{225} \left(\frac{1}{N(R_k(Z_k))} \right) \sum_{i=1}^4 \sum_{j=1}^4 \left(N(R_{k+2}(Z_k, i, j)) - \frac{N(R_k(Z_k))}{16} \right)^2,$$

čia $N(R_{k+1}(Z_k, i))$ ir $N(R_{k+2}(Z_k, i, j))$ tai atitinkamos $R_{k+1}(Z_k, i)$ bei $R_{k+2}(Z_k, i, j)$ sumos. Taigi, tokiu būdu MI parametro skaičiavimas tarp X ir Y tampa apribotas ir baigtinis.

2.4 Neuroninis NARX modelis

Netiesinis autoregresinis tinklas su egzogenine įvestimi (NARX modelis) – tai rekurentinis dinaminis tinklas sudarytas iš kelių sluoksnių tinklo [13]. NARX modelis yra sudarytas iš tiesinio ARX modelio, kuris yra naudojamas modeliuojant laiko eilutes. Bendruoju atveju NARX modelis atrodo taip:

$$y(t) = f\left(d(t-1), d(t-2), \dots, d(t-n_y), u(t-1), u(t-2), \dots, d(t-n_u)\right) + e(t), \quad (25)$$

čia $u(t)$ ir $d(t)$ parodo įvesties ir išvesties modelius tam tikru laiko momentu t ; n_u ir n_y – tai įvesties ir išvesties dinaminio modelio eilė, kur $n_u \geq 0, n_y \geq 1$. Funkcija f , tai netiesinė funkcija, aproksimuojanti analizuojamą neuroninį tinklą. $\bar{X} = (d(t-1) \dots d(t-n_y) u(t-1) \dots u(t-n_u))^T$ pažymimas analizuojamos sistemos įvesties vektorius su žinoma dimensija $n = n_y + n_u$. Kuomet funkcija f yra nežinoma, tai ji yra aproksimuojama regresinio modelio forma:

$$y(t) = \sum_{i=0}^{n_u} a(i)u(t-i) + \sum_{j=1}^{n_y} b(j)y(t-j) + \sum_{i=0}^{n_u} \sum_{j=i}^{n_u} a(i,j)u(t-i)u(t-j) + \sum_{i=1}^{n_y} \sum_{j=i}^{n_y} b(i,j)d(t-i)d(t-j) + \sum_{i=0}^{n_u} \sum_{j=1}^{n_y} c(i,j)u(t-i)y(t-j) + e(t), \quad (26)$$

čia $a(i)$ ir $a(i,j)$ yra tiesiniai ir netiesiniai koeficientai, atsirandantys dėl egzogerinės sąlygos; $b(i)$ ir $b(i,j)$ – tai tiesiniai ir netiesiniai koeficientai, atsirandantys dėl autoregresijos sąlygos; $c(i,j)$ – tai netiesiniai koeficientai, esantys dėl kertamosios sąlygos.

Netiesinis autoregresinis tinklas su egzogenine įvestimi, kaip rekurentinis neuroninis tinklas gali būti treniruojamas vienu iš žemiau išvardintų algoritmų:

- Sklaidos atgal laike algoritmu (angl. Backpropagation through time; BPTT). Vienas iš plačiausiai naudojamų praktikoje.
- Rekurentinio mokymosi dabartiniame laike algoritmas (angl. Real-time recurrent learning; RTRL). Tai tiesioginis matematinis algoritmas, tačiau reikalaujantis daug skaičiavimų resursų ir laiko.

- Išplėstinis Kalman algoritmas (angl. Extended Kalman filters; EKF). Šiuo metodu prognozės atliekamos pakankamai sėkmingai, tačiau dinaminės sistemos tampa suvaržytos.

Šiame darbe buvo naudojamas sklaidos atgal laike algoritmas, nes tai vienas patogiausių ir lanksčiausių algoritmų bei išėjties duomenys mokymosi etape nėra tiesiogiai įtraukiami. Neuroninio tinklo mokymasis yra skirstomas į tris žingsnius:

- Aktyvacijos funkcijos $x(t)$ visoms įvestiems duomenims $u(t)$ apskaičiuoti ir nustatyti;
- Sklaidos atgal paklaidų skaičiavimas visiems laiko momentams, atstumų tarp $x(n)$ ir $y(n)$ skaičiavimas esant laiko momentui n :

$$\delta_j(T) = (d_j(T) - y_j(T)) (\partial f(u)) / \partial u |_{u = z_j(T)} \quad (27)$$

$$\delta_i(T) = (\sum_{j=1}^L \delta_j(T) w_{ji}^{out}) (\partial f(u)) / \partial u |_{u = u_i(n)} \quad (28)$$

$$\delta_j(n) = ((d_j(n) - y_j(n)) \sum_{i=1}^N \delta_i(n+1) w_{ji}^{back}) (\partial f(u)) / \partial u |_{u = z_j(n)} \quad (29)$$

$$\delta_i(n) = (\sum_{i=1}^N \delta_i(n+1) w_{ji} + \sum_{j=1}^L \delta_j(n) w_{ji}^{out}) (f(u)) / \partial u |_{u = z_i(n)}, \quad (30)$$

čia $\delta_j(T)$ - sklaidos atgal paklaida gauta iš išvesties duomenų perceptronų; $\delta_i(T)$ - sklaidos atgal paklaida, gauta dėl perceptronų tarp paslėptųjų sluoksnių ir aktyvatoriaus $x_i(T)$; $\delta_j(n)$ ir $\delta_i(n)$ – sklaidos atgal paklaidos iš išvesties perceptronų, esančių paslėptajame sluoksnyje anksčiausiame sluoksnyje laike T ir potencialiame perceptrone $z_i(n)$.

- Suradus sklaidos atgal paklaidas, skaičiuojami svoriai, jungiantys skirtingus perceptronus:

$$w_{ij} = w_{ij} + \gamma \sum_{n=1}^T \delta_i(n) x_j(n-1) \quad (31)$$

$$(32)$$

$$w_{ij}^{in} = w_{ij}^{in} + \gamma \sum_{n=1}^T \delta_i(n) u_j(n-1) \quad (33)$$

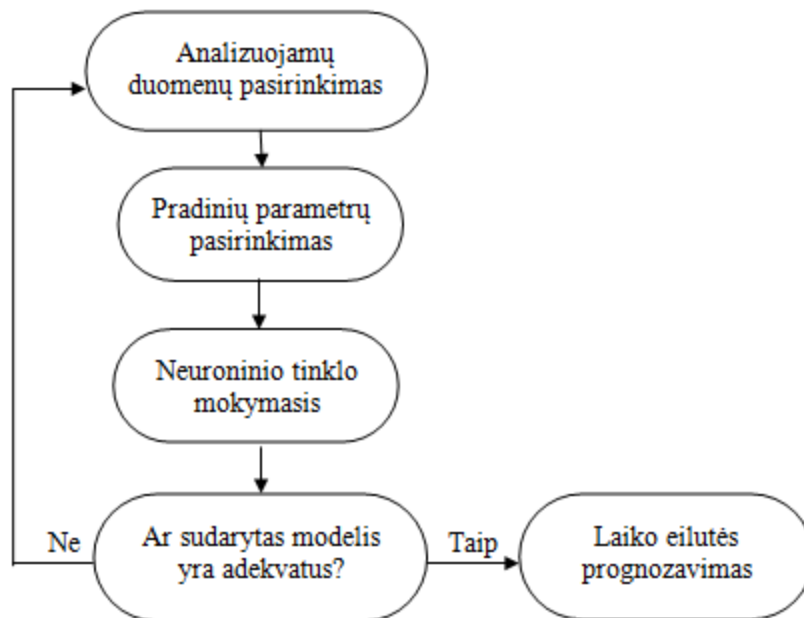
$$w_{ij}^{out} = w_{ij}^{out} \gamma \begin{cases} \sum_{n=1}^T \delta_i(n) u_j(n), & \text{jei } j - \text{išvesties perceptronas} \\ \sum_{n=1}^T \delta_i(n) x_j(n), & \text{jei } j - \text{paslėptasis perceptronas} \end{cases} \quad (34)$$

$$w_{ij}^{back} = w_{ij}^{back} + \gamma \sum_{n=1}^T \delta_i(n) y_j(n-1),$$

čia w_{ij} – svoris, jungiantis paslėptuosius perceptronus; w_{ij}^{in} , w_{ij}^{out} ir w_{ij}^{back} – atitinkami įvesties, išvesties ir atsako svoriai; γ – labai maža vertė, naudojama minimizuojant kvadratinę paklaidą.

Šiame darbe naudojama Levenberg – Marquardta [14] aproksimuojanti funkcija, kuri atnaujina svorius ir poslinkių vertes, naudojant Levenberg – Marquardt optimizavimą. Ši funkcija yra rekomenduojama naudoti kartu su sklaidos atgal laike algoritmu, kadangi pakankamai greitai yra atliekami skaičiavimai, nors mokymosi metu yra sunaudojama daugiau atminties negu kitos aproksimuojančios funkcijos.

NARX modelio schema:



Analizuojamų duomenų pasirinkimas, tai vienas svarbiausių žingsnių sudarinėjant neuroninį tinklą, nuo jo priklauso sudaromojo modelio adekvatumas. Duomenys yra skirstomi į dvi lygias dalis: įvesties ir išvesties laiko eilutes. Įvesties duomenys naudojami neuroninio tinklo apmokymui, o išvesties duomenys naudojami neuroninio tinklo validavimui bei tikrinimui atlikti.

Pradinių sąlygų pasirinkimas, tai laiko vėlinimo parametro, paslėptųjų sluoksnių skaičiaus identifikavimas, procentinis duomenų suskirstymas, kai pasirenkama kiek duomenų bus naudojama mokymuisi, validavimui ir testavimui.

Atlikus visus aukščiau įvardintus žingsnius yra pradedamas neuroninio tinklo mokymosi etapas. Gautąjo modelio adekvatumo vertinimas susideda iš paklaidų stebėjimo, jų dydžio bei pasiskirstymo. Taip pat vertinama paklaidų autokoreliacija, koreliacija tarp įvesties duomenų ir paklaidų. Kuomet tenkinamos visos adekvatumo sąlygos, neuroninis tinklas yra paruoštas atlikti analizuojamos laiko eilutės prognozei. Priešingu atveju yra grįžtama į pirmąjį žingsnį – analizuojama ar parinkti duomenys yra adekvatūs tyrimui, jeigu taip, tuomet analizuojama ar parinkti pirminiai parametrai yra optimalūs. Šis modeliavimo procesas yra tęsiamas, kol NARX modelis tampa adekvatus prognozavimui.

2.5 Statistiniai kriterijai

Atliekant laiko eilutės analizę yra susiduriama su problema, kaip įvertinti gautųjų rezultatų adekvatumą, tada yra patogiau vertinti statistines hipotezes. Šiame darbe statistinių hipotezių vertinimas buvo atliktas, kai laiko eilutės buvo analizuojamos neuroniniu NARX tinklu, tiksliau buvo panaudota paklaidų korektiškumo nustatymui. Tikrinamos dviejų tipų hipotezės: hipotezė apie paklaidų vidurkio lygybę nuliui, panaudojus Stjudento kriterijų, ir hipotezė apie paklaidų normalųjį skirstinį, panaudojus Šapiro – Wilk kriterijų bei Jarque – Bera testą.

2.5.1 Stjudento t kriterijus

Statistikoje Stjudento t testas [15], tai hipotezių apie mažos imties, pasiskirsčiusios pagal normalųjį skirstinį, vidurkių lygybės metodas, kai standartinis nuokrypis yra nežinomas. 1908 m. William Sealy Gosset išplėtojo t testą ir t pasiskirstymą. Pasiskirstymas t – tai kreivė, kurios laisvės laipsnių skaičius, apskaičiuojamas iš analizuojamų duomenų dydžio atėmus 1, pažymi tam tikrą kreivę. Kuomet analizuojami duomenys didėja, tai t pasiskirstymas įgauna varpo formą. Praktikoje dažniausiai turint duomenų dydį 30 ir didesnį, normalusis pasiskirstymas yra priimtinas. Šiame darbe kadangi buvo analizuojamos paklaidos, tai buvo tikrinama nulinė hipotezė, ar paklaidų vidurkis yra lygus nuliui:

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}, \quad (35)$$

čia \bar{x} analizuojamų duomenų vidurkis, μ_0 nustatytas vidurkis, šiuo atveju 0, s – tai standartinis nuokrypis ir n – analizuojamų duomenų dydis.

Taip pat nustatomas reikšmingumo lygmuo α , dažniausiai priimtinas $\alpha = 0,05$, tuomet hipotezė neatmetama arba atmetama su 95% reikšmingumu. Jeigu apskaičiuota t reikšmė patenka į intervalą tarp kritinių reikšmių, kurie vertinami pagal laisvės laipsnių skaičių, tai hipotezė neatmetama.

2.5.2 Šapiro – Wilk kriterijus

Statistikoje Šapiro – Wilk testas [15] plačiai naudojamas nustatyti, ar duomenys yra pasiskirstę pagal normalųjį skirstinį. Statistika yra apibrėžiama:

$$W = \frac{(\sum_{i=1}^n a_i x_{(i)})^2}{\sum_{i=1}^n (x_i - \bar{x})^2}, \quad (36)$$

čia x – tikrinama duomenų eilutė; $x_{(i)}$ – i -tosios eilės statistika arba kitaip i -tasis duomenų mažiausias skaičius; \bar{x} – duomenų vidurkis; a_i konstanta apskaičiuojama:

$$(a_1, \dots, a_n) = \frac{m^T V^{-1}}{(m^T V^{-1} V^{-1} m)^{1/2}}, \quad (37)$$

čia $m = (m_1, \dots, m_n)^T$ – tai nepriklausomųjų kintamųjų tikimosios reikšmės apskaičiuotos pagal standartinį normalųjį skirstinį, o V – tai kovariacinė matrica.

Nulinė hipotezė šiame darbe tikrinama, ar paklaidos yra pasiskirsčiusios pagal normalųjį skirstinį. Jeigu apskaičiuotą reikšmę yra didesnė už nustatytą reikšmingumo lygmenį, dažniausiai $\alpha = 0,05$, tai hipotezė apie normalųjį skirstinį yra neatmetama.

2.5.3 Jarque – Bera kriterijus

Statistikoje Jarque – Bera testas yra naudojamas patikrinti, ar analizuojami duomenys yra pasiskirstę pagal normalųjį skirstinį, kai nėra žinomas vidurkis ir standartinis nuokrypis. Testo statistika apibrėžiama:

$$JB = \frac{n}{6} \left(s^2 + \frac{(k-3)^2}{4} \right),$$

čia n – analizuojamos laiko eilutės dydis; s – eilutės asimetrija; k – eilutės ekscesas.

Šis testas yra pranašesnis už Šapiro – Wilk testą, nes gali apdoroti ir dideles laiko eilutes, tuomet statistika įgauna chi – kvadrato pasiskirstymą. Šiame darbe Jarque – Bera testas buvo naudojamas tikrinti, ar sudarytojo NARX modelio paklaidos apsimokymo metu yra pasiskirsčiusios pagal normalųjį skirstinį su reikšmingumo lygmeniu 0,05.

3. Tyrimų rezultatai ir jų aptarimas

3.1 Lorencio atraktorius

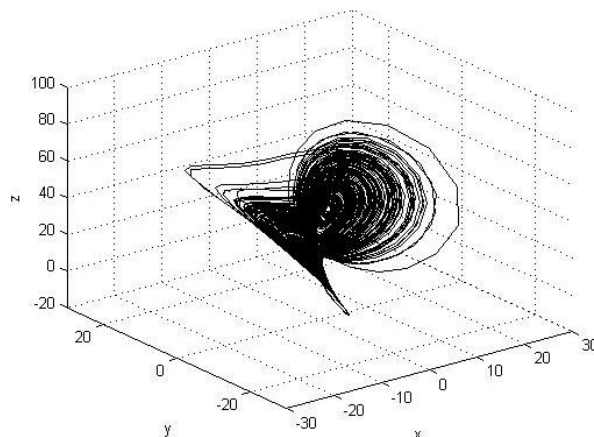
Atraktorius yra žinomas kaip aibė taškų, kuriais juda dinaminė sistema praėjus pakankamai daug laiko. Kitaip tariant, taškai, atsidūrę arti atraktoriaus ten ir išliks, net ir tuo atveju, kai būtų bandoma trikdyti judėjimą. Geometrine prasme atraktorius gali būti taškų / kreivių aibė su fraktaline struktūra, dar vadinama keistuoju atraktoriumi.

Plačiausiai paplitęs keistojo atraktoriaus pavyzdys – Lorencio atraktorius [11]. Jis yra žinomas nuo 1963m., kuomet Edvardas Lorencas šį keistąjį atraktorių išgavo suprastindamas konvekciją aprašančias lygtis. Bedimensinės formos Lorencio laiko eilutės lygtys atrodo taip:

$$(38) \quad \begin{cases} \frac{dx}{dt} = \sigma(y - x) \\ \frac{dy}{dt} = x(\rho - z) - y \\ \frac{dz}{dt} = xy - \beta z \end{cases}$$

čia $\sigma = 10$, $\beta = \frac{8}{3}$, $\rho = 28$.

Pasinaudojus aukščiau pateikta lygčių sistema buvo sugeneruotas 11 000 – šių taškų Lorencio atraktorius, su pradinėmis sąlygomis (1, 1, 1):



3.1 pav. Lorencio atraktorius

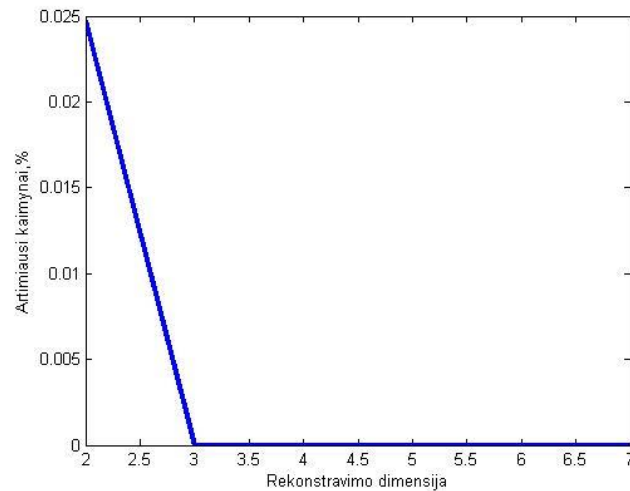
Taigi, turima laiko eilutė toliau buvo analizuojama, nustatant kokioje dimensijoje reikia atraktorių rekonstruoti bei imant atraktoriaus taškus nuo 8000 iki 11000, kuomet jau yra aiški ir

nusistovėjusi dinamika. Taip pat vertinamas laiko vėlinimo parametrastarpusavio informacijos metodu bei su nustatytu parametru atliekama prognozė pasinaudojus neuroninį NARX tinklą.

3.1.1 Lorencio atraktoriaus rekonstravimo dimensijos nustatymas

Lorenco laiko eilutės rekonstravimo dimensija buvo nustatyta dviem metodais: artimiausio klaidingo kaimyno ir koreliacinės dimensijos metodu. Pirmiausia, analizuojamas buvo artimiausio klaidingo kaimyno metodas, kuris yra geriausias metodas, norint nustatyti mažiausią chaotinės laiko eilutės rekonstravimo dimensiją. Svarbiausias šio metodo tikslas yra stebėti artimiausių kaimynų elgesį kintant dimensijai, kai $d \rightarrow d + 1$. Kai artimiausių kaimynų skaičius artėja prie nulio, tai yra laikoma, kad laiko eilutė yra pilnai išskleista šioje dimensijoje.

Rekonstravimo dimensijos nustatymas 11 000 – šių Lorenco atraktoriaus taškų, kuomet laiko vėlinimo parametras kinta nuo 1 iki 10:



3.2 pav. Lorenco atraktoriaus dimensija artimiausio klaidingo kaimyno metodu

Iš aukščiau pateikto paveikslėlio matyti, kad artimiausių kaimynų skaičius priartėjo prie nulio, kai rekonstravimo dimensija yra 3 ir didesnė, vadinasi mažiausia rekonstravimo dimensija yra lygi 3.

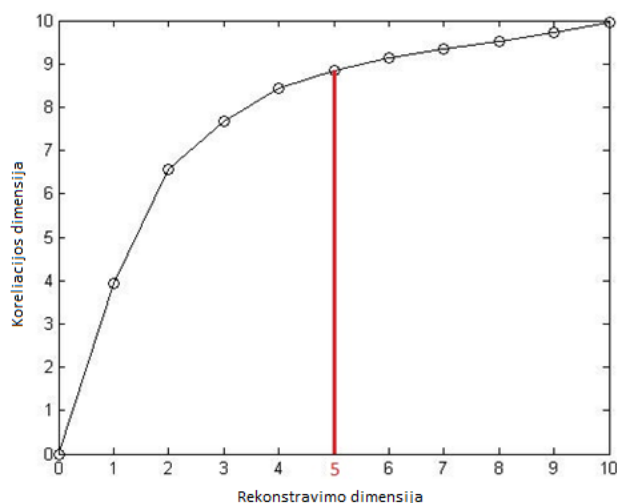
Toliau stebime, kaip kinta dimensijos vertė, kuomet laiko vėlinimo parametras τ yra pakankamai didelis:

1 lentelė. Lorencio atraktoriaus rekonstravimo dimensija, kintant laiko vėlinimui

τ	15	20	25	30	40	50	75	100
Dimensija	15	14,5	12	12,2	10,2	8	7,5	7,5

Iš lentelės matoma, kad didėjant laiko vėlinimo parametru, didėja ir rekonstravimo dimensija, todėl toliau tyrimas bus tęsiamas nustatant būtent parametro τ vertę.

Tuomet tos pačios laiko eilutės nustatoma koreliacinės dimensijos metodu:



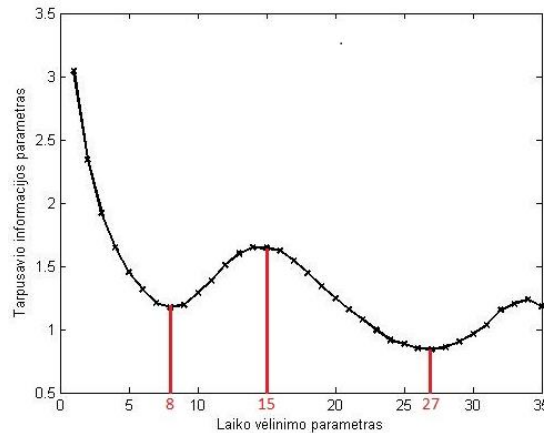
3.3 pav. Lorencio atraktoriaus dimensija koreliacinės dimensijos metodu

Iš aukščiau pateikto 3.3 paveikslo yra matoma, kad nusistovėjimas yra matomas, kai dimensija yra lygi 5. Nors koreliacinės dimensijos metodas yra skirtas dirbti su netiesinėmis sistemomis, tačiau Lorencio atraktoriaus laiko eilutė yra pakankamai chaotinė, todėl šiuo metodu nustatoma dimensija yra didesnė negu artimiausio klaidingo kaimyno metodu.

3.1.2 Lorencio atraktoriaus laiko vėlinimo nustatymas

Pasinaudojus Tarpusavio informacijos metodu buvo atliktas laiko vėlinimo parametro vertės nustatymas. Šiuo metodu yra lyginama, kaip kinta Tarpusavio informacijos parametras kintant laiko vėlinimui τ . Pasiekus pirmąjį minimumą, yra laikoma, kad su atitinkamu parametru τ reikia atlikti analizuojamos laiko eilutės rekonstravimą, nes tuomet laiko eilutė praranda mažiausiai savosios informacijos.

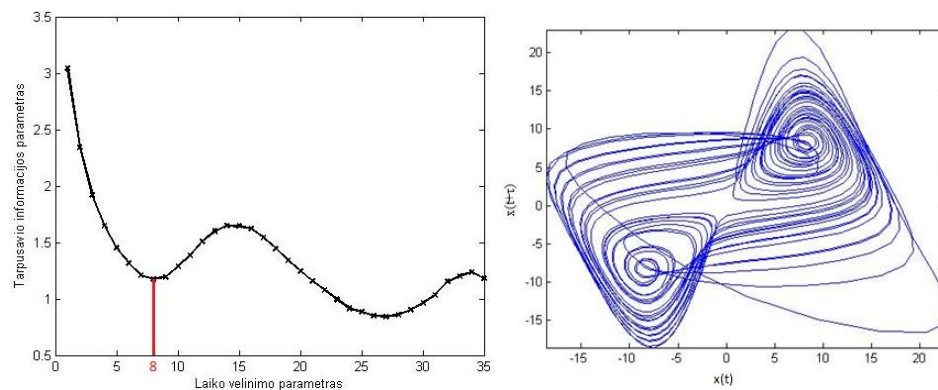
11 000 – šių Lorencio atraktoriaus taškų rezultatai:



3.4 pav. Lorencio atraktoriaus tarpusavio informacijos metodu

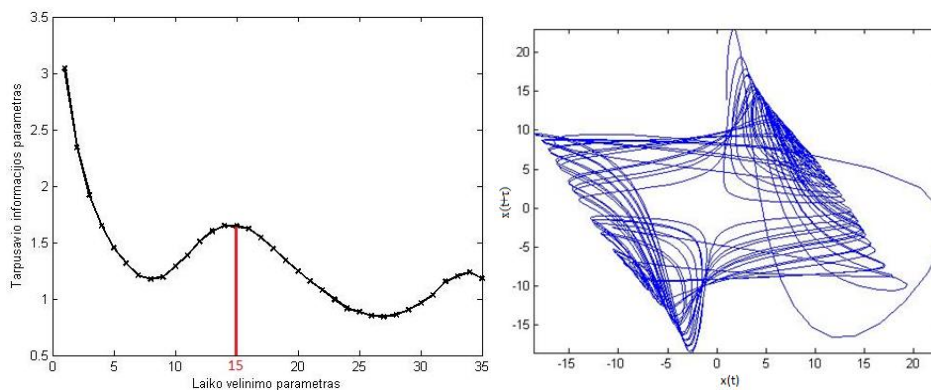
Iš aukščiau pateiktojo paveikslėlio matoma, kad pirmasis minimumas pasiekiamas, kai $\tau = 8$, antrasis – kai $\tau = 27$, o pirmasis maksimumas, kai $\tau = 15$. Taigi, toliau rekonstruojame Lorencio atraktorių su skirtingais laiko vėlinimais.

Lorencio atraktoriaus rekonstravimas, kai laiko vėlinimas $\tau = 8$:



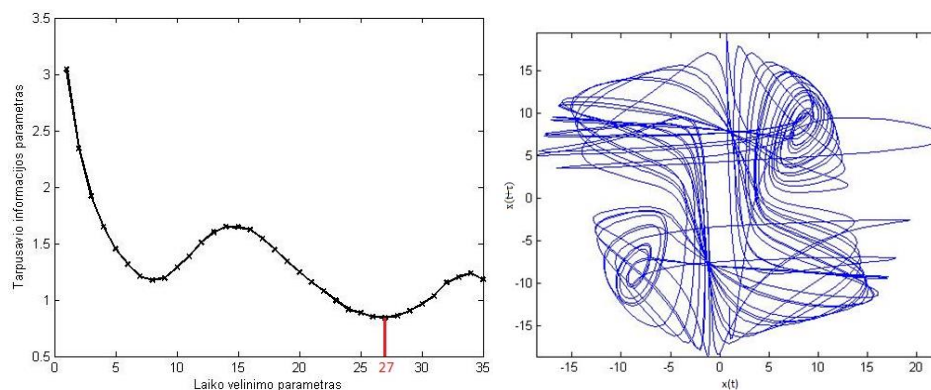
3.5 pav. Rekonstruotas Lorencio atraktoriaus, kai $\tau = 8$

Lorenco atraktoriaus rekonstravimas, kai laiko vėlinimas $\tau = 15$:



3.6 pav. Rekonstruotas Lorenco atraktorius, kai $\tau = 15$

Lorenco atraktoriaus rekonstravimas, kai laiko vėlinimas $\tau = 27$:



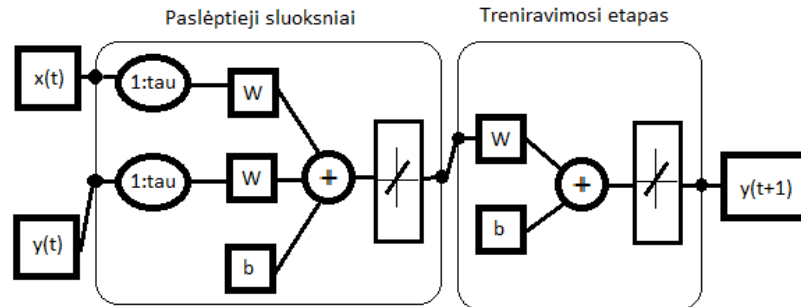
3.7 pav. Rekonstruotas Lorenco atraktorius, kai $\tau = 27$

Korektiškiausias Lorenco atraktoriaus 11 000 – šių taškų rekonstravimas buvo atliktas renkantis laiko vėlinimo parametą, kai tarpusavio informacijos reikšmė įgyja pirmąjį minimumą, tuomet analizuojama laiko eilutė praranda mažiausią dalį savosios informacijos. Rekonstruojant laiko eilutę su pasirinktomis didesnės laiko vėlinimo τ vertėm, gauti vaizdai dvimatėje erdvėje yra nepanašūs į analizuojamą eilutę, vadinasi buvo prarasta ne mažieilutės unikalios informacijos.

Taigi, Tarpusavio informacijos metodu buvo nustatyta Lorenco atraktoriaus rekonstravimo parametro τ optimaliausia reikšmė, kai $\tau = 8$.

3.1.3 Lorencio atraktoriaus prognozavimas su neuroniniu NARX tinklu

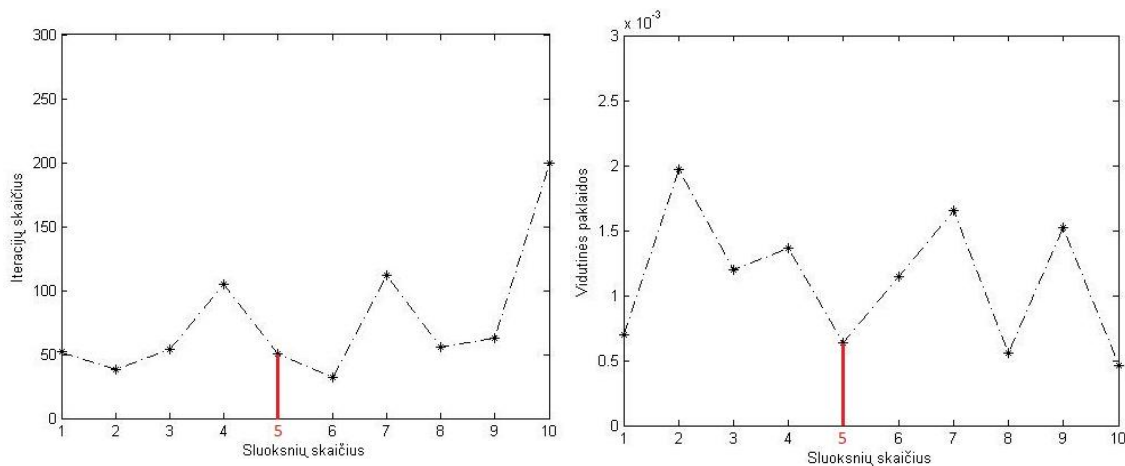
Lorencio atraktoriaus prognozavimas buvo atliktas panaudojus NARX neuroninį tinklą [13].



3.8 pav. Neuroninio NARX tinklo mokymosi schema

Pirmiausia, buvo paimti Lorencio laiko eilutės taškai $x(t)$ nuo 8 000 iki 9 000, kad būtų atliktas neuroninio tinklo apsimokymo etapas. Apsimokymo etapas buvo atliktas, kuomet 70 proc. taškų panaudota treniravimuisi ir po 15 proc. taškų panaudota validavimui ir testavimui. Tiksliniai taškai $y(t)$ buvo paimti iš Lorencio atraktoriaus eilutės imant taškus nuo 9 001 iki 10 000. Pasinaudojus laiko vėlinimo nustatymo tyrimu buvo parinktas parametras $\tau = 8$.

Paslėptųjų sluoksnių skaičiui nustatyti buvo atlikta analizė, tai yra buvo tiriama, kaip kinta paklaidos ir apsimokymo iteracijų skaičius kintant sluoksnių skaičiui nuo 1 iki 10:

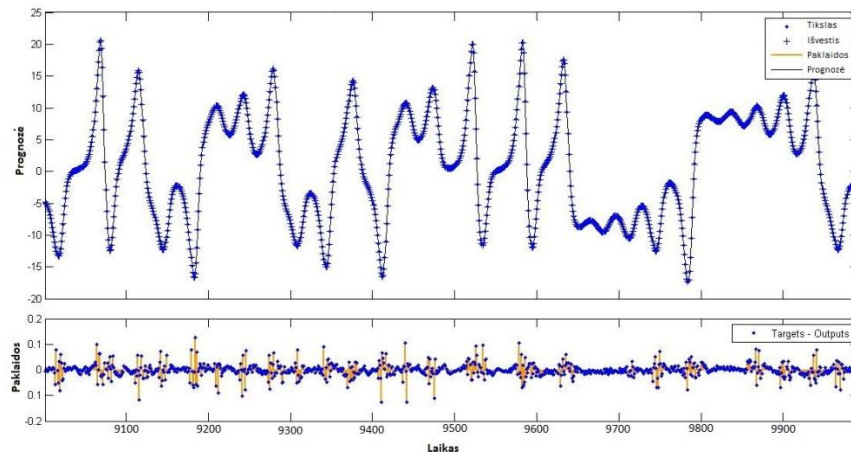


3.9 pav. Lorencio atraktoriaus neuroninio tinklo sluoksnių tyrimas

Iš aukščiau pateiktų paveikslėlių yra matoma, kad kintant sluoksnių skaičiui paklaidos kinta labai neženkliai, tačiau apsimokymo iteracijų skaičius didėja didėjant sluoksniams, pavyzdžiui esant 10 paslėptų sluoksnių, iteracijos padidėja net iki 200. Taigi, toliau buvo pasirinkta

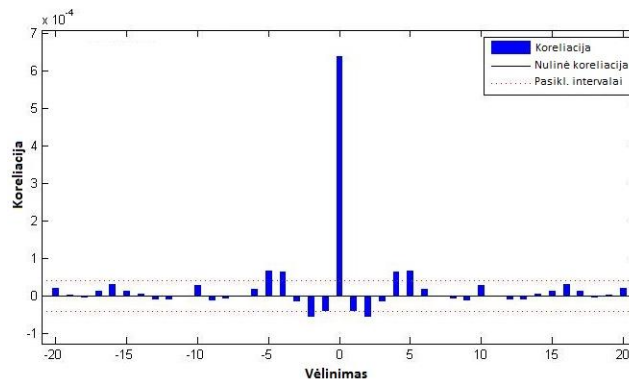
skaičiavimuose naudoti 5 paslėptus sluoksnius. Tuomet paklaidos yra mažiausios ir neuroninio tinklo apsimokymo iteracijų skaičius yra nedidelis – apie 50.

Lorenco atraktoriaus taškų nuo 8 000 iki 9 000 neuroninio tinklo apsimokymo etapo rezultatai atrodo taip:



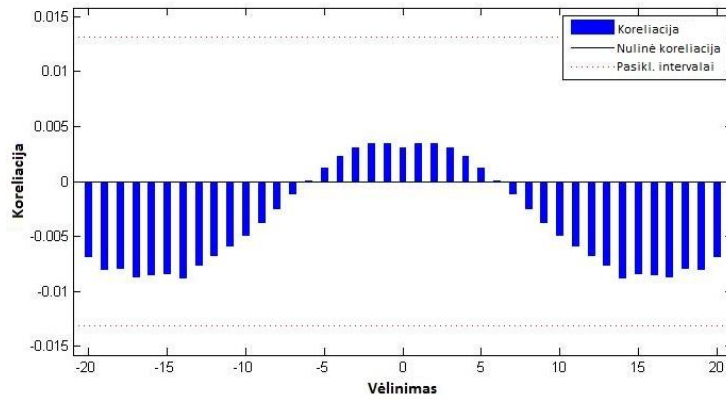
3.10 pav. Lorenco atraktoriaus neuroninio tinklo mokymosi rezultatai

Matoma, kad apsimokymo paklaidos yra pakankamai mažos, jos svyruoja apie nulį, nepriklausomai nuo pokyčio laike. Toliau analizuojama, ar paklaidos yra adekvačios prognozavimo modeliui.



3.11 pav. Lorenco atraktoriaus neuroninio tinklo paklaidų koreliacija

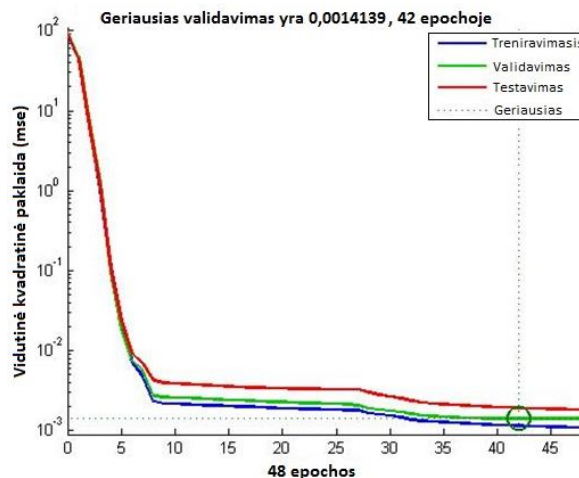
Aukščiau yra pateikta prognozavimo paklaidų autokoreliacijos funkcija, parodanti kaip paklaidos koreliuoja kintant laiko vėlinimui. Gauti rezultatai yra pakankamai geri, kadangi autokoreliacijos funkcijos vertės turi svyruoti apie nulį ir tik tuomet vėlinimas lygus 0, tuomet vertė turi staigiai išaugti. Vadinasi, prognozavimo paklaidos nekoreliuoja tarpusavyje.



3.12 pav. Lorencio atraktoriaus neuroninio tinklo koreliacija tarp duomenų ir paklaidų

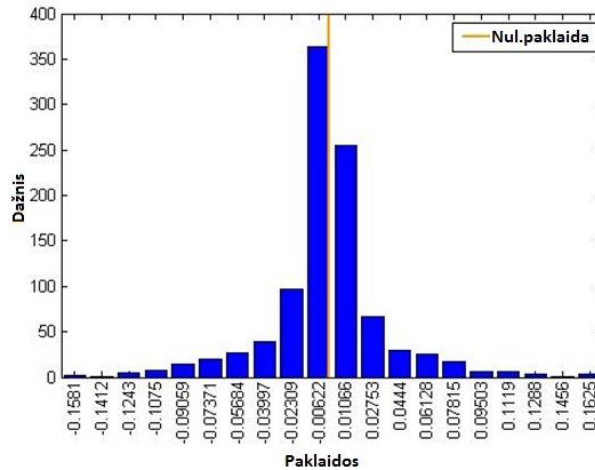
Papildomai tikrinama koreliacija tarp duotųjų Lorencio taškų ir prognozavimo paklaidų. Iš aukščiau pateikto grafiko matyti, kad prognozavimo korektiškumas yra patvirtinamas, nes koreliacijos rezultato reikšmės svyruoja apie nulį ir patenka į pasikliautiniųjų intervalų ribas.

Šie rezultatai pasiekti esant 42 – ai iteracijai, tolimesniuose iteracijose rezultatai nebe gerėja, vadinasi reikia sustoti atlikti testavimo darbus. Kadangi iki minimos iteracijos visų trijų etapų paklaidos tik mažėjo, vadinasi iki geriausio rezultato nebuvo įvykdytas duomenų perpildymas.



3.13 pav. Lorencio atraktoriaus neuroninio tinklo validavimo rezultatai

Taip pat buvo ištirta, kaip prognozavimo paklaidos tarpusavyje yra priklausomos, esant korektiškam modeliui paklaidos yra pasiskirsčiusios pagal normalųjį skirstinį.



3.14 pav. Lorencio atraktorinio neuroninio tinklo paklaidų pasiskirstymas

Kadangi iš aukščiau pateiktos histogramos yra matomas panašus pasiskirstymas į normalųjį, taip pat buvo patikrinta, pasinaudojus Jarque – Bera kriterijumi, hipotezė:

H_0 : paklaidos pasiskirsčiusios pagal normalųjį skirstinį

H_1 : paklaidos nėra pasiskirsčiusios pagal normalųjį skirstinį

$\alpha = 0.01$

$H = 0$ (hipotezė neatmetama)

$p = 0,0220$

Patikrinus hipotezę apie paklaidų normalųjį skirstinį, padarome išvadą, kad hipotezę neatmetame, nes $p > \alpha$ nors ir esant labai mažam reikšmingumo lygmeniui. Toliau tikriname pagal Stjudento kriterijų, ar paklaidų vidurkis yra lygus 0:

$H_0: \mu = 0$

$H_1: \mu \neq 0$

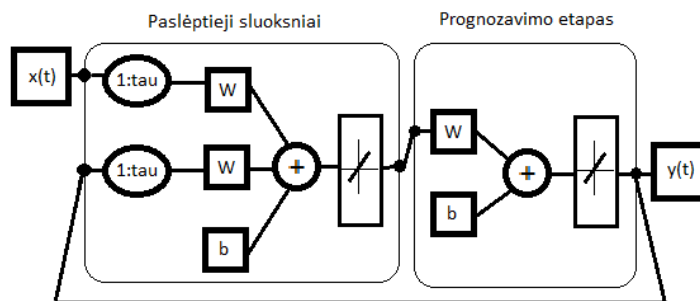
$\alpha = 0.05$

$H = 0$ (hipotezė neatmetama)

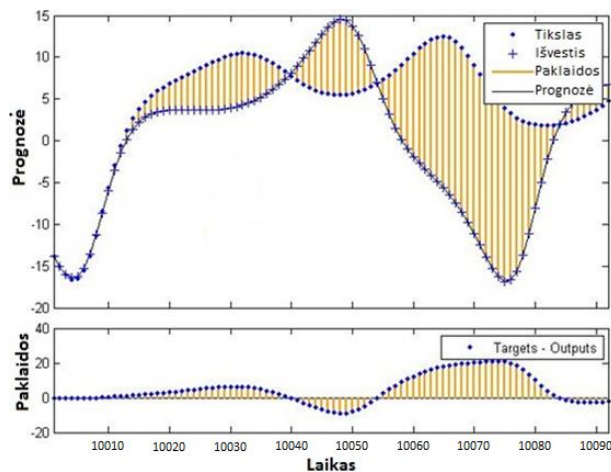
$p = 0.6159$

$ci = [-0.0029; 0.0017]$

Taigi, hipotezė apie vidurkio lygybę nuliui yra neatmetama su tikimybe 0,6159 ($p > \alpha$), vadinasi modelis yra tinkamas tolimesniems tyrimams, todėl toliau su treniruoti neuroniniu tinklu atliekamas Lorencio laiko eilutės prognozavimas:

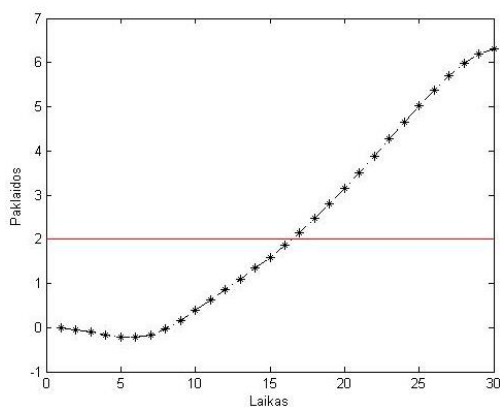


3.15 pav. Neuroninio NARX tinklo prognozavimo schema



3.16 pav. Lorencio atraktoriaus prognozė neuroniniu NARX tinklo metodu

Iš aukščiau pateikto Lorencio atraktoriaus prognozavimo yra matoma, kad negalima prognozuoti tolimų šio atraktoriaus taškų, nes iš prigimties ši laiko eilutė yra chaotiškai išsidėsčiusi laike, tad panagrinėkime kaip prognozuojami yra pirmieji 30 taškai:



3.17 pav. Lorencio atraktoriaus prognozavimo paklaidų vertinimas

Raudona horizontalia linija pažymėta nustatyta paklaidų riba, kadangi Lorencio atraktorius yra sunkiai prognozuojamas, buvo ištirta, kuomet procentiškai paklaidos lyginant su originaliais

duomenimis, neviršytų 3proc., tai yra prognozuojant pirmuosius 16 taškų paklaidos nenutolsta toliau nei 3proc, o prognozuojant tolimesnius taškus paklaidos staigiai didėja, pavyzdžiui prognozuojant 25 Lorencio atraktoriaus tašką paklaidos pasiekia net 50proc. vidutinių atraktoriaus verčių. Taip pat, patikrinta buvo statistinės hipotezės, kai imamos pirmosios 16 prognozuojamų taškų. Studento t hipotezė apie paklaidų vidurkį lygų 0 buvo neatmetama, kai reikšmingumo lygmuo yra 0,005, nes apskaičiuota tikimybė lygi 0,0098. Įvertinus Šapiro – Wilk kriterijų, daroma išvada, kad hipotezė apie paklaidų normalųjį skirstinį yra neatmetama su tikimybe 0,0875 esant reikšmingumo lygmeniui 0,05. Tad yra matoma, kad norimo rezultato galima pasiekti tik prognozuojant pirmuosius 16 taškų, apskritai, tiksliausia prognozė buvo atlikta pirmųjų 8 taškų, o po to yra staigus paklaidų padidėjimas.

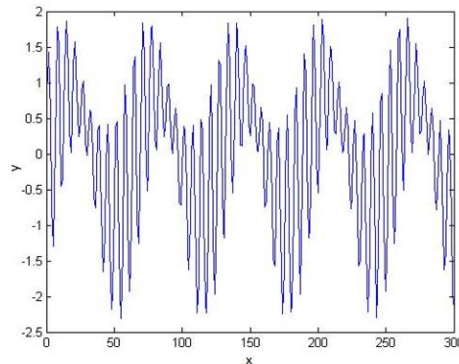
Išvados. Remiantis atliktais Lorencio atraktoriaus tyrimais galima teigti, kad artimiausio kaimyno metodu buvo nustatyta mažiausia rekonstravimo dimensija lygi 3, tarpusavio informacijos metodu nustatytas laiko vėlinimo parametras lygus 8. Pasinaudojus nustatytuoju laiko vėlinimo parametru atliktas laiko eilutės prognozavimas neuroniniu NARX tinklu pakankamai korektiškai, kadangi neuroninis tinklas buvo apmokytas su mažomis prognozavimo paklaidomis, o tolimesni taškai prognozuojami tiksliausiai iki 16 taško. Lorencio atraktoriaus negalima prognozuoti labai toli, nes ši eilutė chaotiškai juda kintant laikui.

3.2 Sumodeliuota laiko eilutė

Šiame darbe sumodeliuota laiko eilutė iš skirtingų fazių ir amplitudžių harmoninių funkcijos sumos:

$$x(t) = \sin(t) + A_1 \sin(\omega_1 t + \varphi_1) + A_2 \sin(\omega_2 t + \varphi_2), \quad (39)$$

čia $A_1 = 0.9$, $A_2 = 0.6$, $\omega_1 = 0.1$, $\omega_2 = 0.9$.

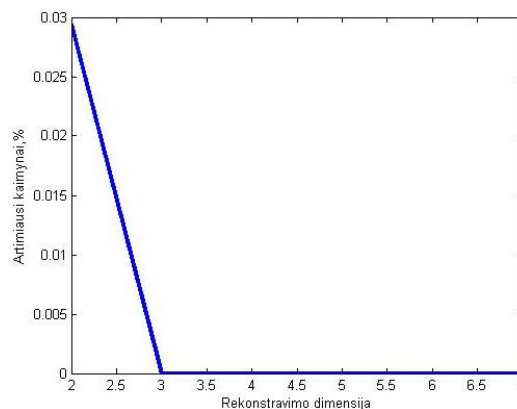


3.18 Sumodeliuota laiko eilutė

3.18 paveiksle yra pavaizduota, kai atrodo pirmieji 300 sumodeliuotos laiko eilutės taškų, toliau darbe buvo naudojama šios formos 3 000 taškų eilutė.

3.2.1 Sumodeliuotos laiko eilutės rekonstravimo dimensijos nustatymas

Rekonstravimo dimensijos nustatymas 3 000 – šių sumodeliuotos laiko eilutės taškų, kuomet laiko vėlinimo parametras kinta nuo 1 iki 10:



3.19 Sumodeliuotos laiko eilutės dimensija artimiausio klaidingo kaimyno metodu

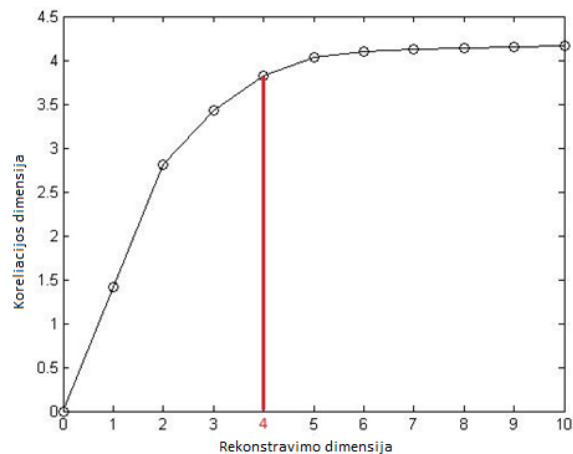
Iš aukščiau pateikto paveikslėlio matyti, kad artimiausių kaimynų skaičius priartėjo prie nulio, kuomet rekonstravimo dimensija yra 3 ir didesnė, vadinasi mažiausia rekonstravimo dimensija yra lygi 3.

Toliau stebime, kaip kinta dimensijos vertė, kai laiko vėlinimo parametras τ yra pakankamai didelis:

2 lentelė. Sumodeliuotos laiko eilutės rekonstravimo dimensija, kintant laiko vėlinimui

τ	15	20	25	30	40	50	75	100
Dimensija	3	3	3	3	3	3	3	3

Iš lentelės matoma, kad didėjant laiko vėlinimo parametrai, rekonstravimo dimensija nekinta, vadinasi sumodeliuotos laiko eilutės rekonstravimui vėlinimo parametras nėra svarbus. Taip pat atliekamas tyrimas koreliacinės dimensijos metodu:

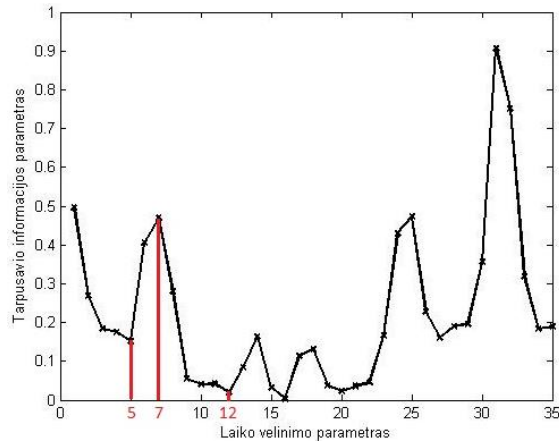


3.20 Sumodeliuotos laiko eilutės dimensija koreliacinės dimensijos metodu

Iš 3.20 paveikslo matoma, kad nusistovėjimas yra, kai dimensija yra lygi 4, toks rezultatas yra kuomet laiko vėlinimo parametras yra ne didesnis negu 8, kitu atveju nusistovėjo nėra ir grafikas tęsiasi iki begalybės. Taigi, toliau remiamės artimiausio klaidingo kaimyno rezultatais, nes naudojanti šiuo metu laiko vėlinimo parametrast įtakos neturi ir rekonstravimo dimensija yra gauta mažiausia.

3.2.2 Sumodeliuotos laiko eilutės laiko vėlinimo nustatymas

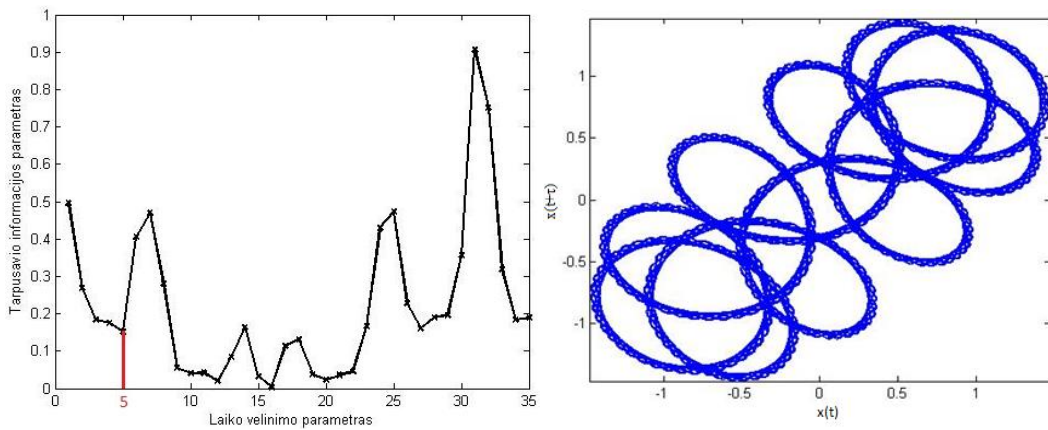
Pasinaudojus tarpusavio informacijos metodu buvo atliktas laiko vėlinimo parametro vertės nustatymas sumodeliuotai laiko eilutei. 3 000 – šių laiko eilutės rezultatai:



3.21 pav. Sumodeliuotos laiko eilutė tarpusavio informacijos metodu

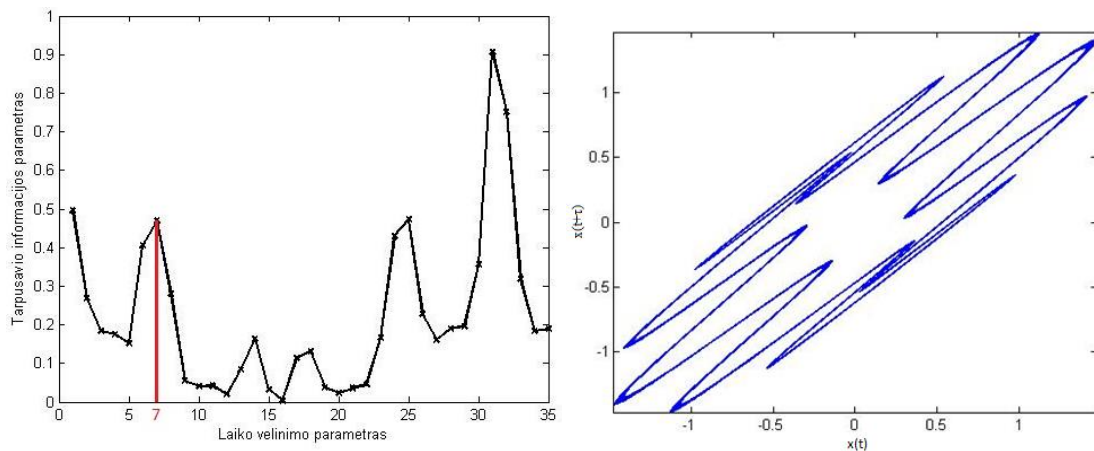
Iš aukščiau pateiktojo paveikslėlio matoma, kad pirmasis minimumas pasiekiamas, kai $\tau = 5$, antrasis – kai $\tau = 12$, o pirmasis maksimumas, kai $\tau = 7$. Taigi, toliau rekonstruojame sumodeliuotą laiko eilutę su skirtingais laiko vėlinimais.

Sumodeliuotos laiko eilutės rekonstravimas, kai laiko vėlinimas $\tau = 5$:



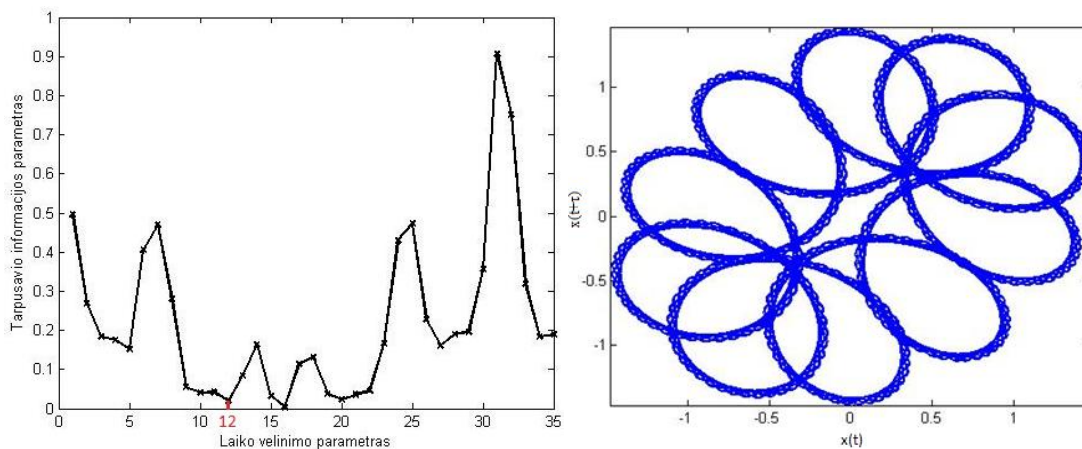
3.22 pav. Rekonstruota sumodeliuota laiko eilutė, kai $\tau = 5$

Sumodeliuotos laiko eilutės rekonstravimas, kai laiko vėlinimas $\tau = 7$:



3.23 pav. Rekonstruota sumodeliuota laiko eilutė, kai $\tau = 7$

Sumodeliuotos laiko eilutės rekonstravimas, kai laiko vėlinimas $\tau = 12$:



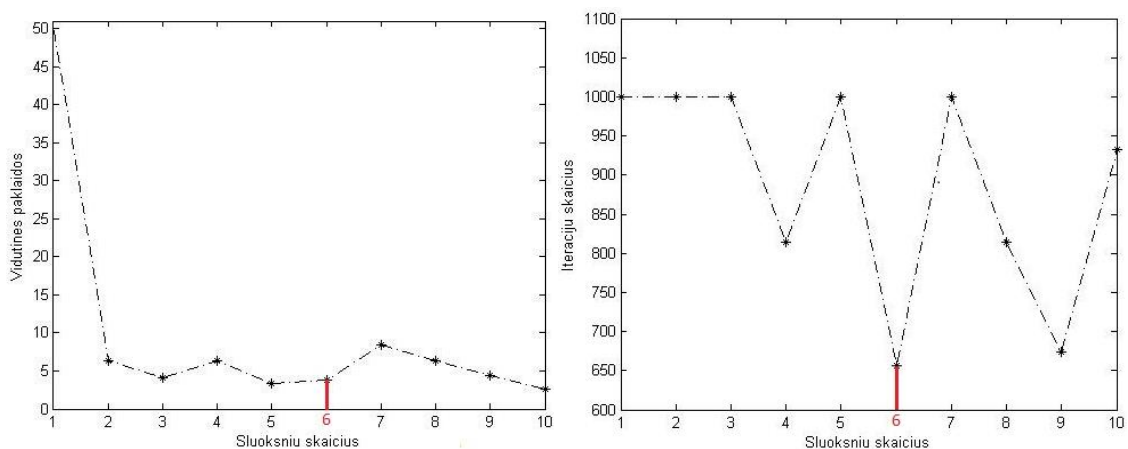
3.24 pav. Rekonstruota sumodeliuota laiko eilutė, kai $\tau = 12$

Korektiškiausias sumodeliuotos laiko eilutės 3 000 – čių taškų rekonstravimas buvo atliktas renkantis laiko vėlinimo parametą, taigi, tarpusavio informacijos metodu buvo nustatyta sumodeliuotos laiko eilutės rekonstravimo parametro τ optimaliausia reikšmė, kai $\tau = 5$.

3.2.3 Sumodeliuotos laiko eilutės prognozavimas su neuroniniu NARX tinklu

Sumodeliuotos laiko eilutės prognozavimas buvo atliktas panaudojus NARX neuroninį tinklą. Pirmiausia, buvo paimti pirmieji 2000 taškai (1000 įvesčiai ir 1000 išvesčiai), kad būtų atliktas neuroninio tinklo apsimokymo etapas. Apsimokymo etapas buvo atliktas, kuomet 70 proc. taškų panaudota mokymuisi ir po 15 proc. taškų panaudota validavimui ir testavimui. Pasinaudojus laiko vėlinimo nustatymo tyrimu buvo parinktas parametras $\tau = 6$.

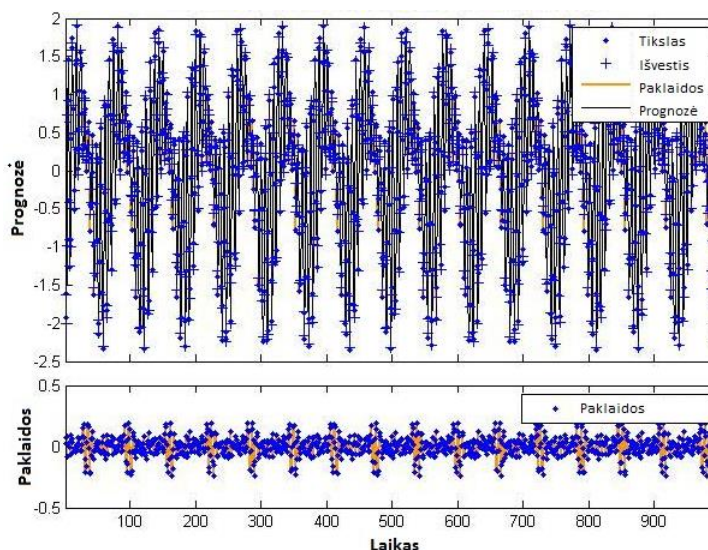
Paslėptųjų sluoksnių skaičiui nustatyti buvo atlikta analizė, kaip kinta paklaidos ir prognozavimo iteracijų skaičius kintant sluoksnių skaičiui nuo 1 iki 10. Kadangi analizuojama laiko eilutė yra periodinė, tai tyrimui pasirinkta ne apsimokymo, bet prognozavimo taškai nuo 1001 iki 1050.



3.25 pav. Sumodeliuotos laiko eilutės neuroninio tinklo sluoksnių tyrimas

Iš aukščiau pateiktų paveikslėlių yra matoma, kad kintant sluoksnių skaičiui paklaidos kinta labai neženkliai, kai paslėptųjų sluoksnių skaičius yra 2 ir daugiau. Prognozės iteracijų skaičius yra labai didelis, daugeliu atveju siekia maksimalų skaičių – 1000, nes kiekvienos iteracijos metu neuroninis tinklas pasiekia geresnį rezultatą. Mažiausias iteracijų skaičius buvo pasiektas esant 6 sluoksniams. Vadinas, tolimesnius skaičiavimus atliksime su 6 paslėptaisiais sluoksniais.

Sumodeliuotos laiko eilutės neuroninio tinklo mokymosi rezultatai:

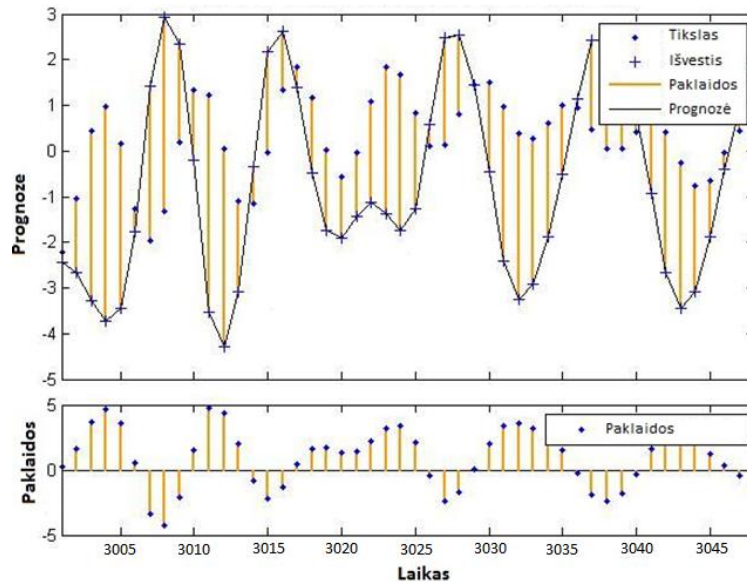


3.26 pav. Sumodeliuotos laiko eilutės neuroninio tinklo mokymosi etapo rezultatai

Neuroninio tinklo mokymosi etapas atliktas labai gerai, kadangi prognozavimo paklaidos svyruoja intervale $[-0,5; 0,5]$. Atlikus sudarytojo modelio adekvatumo vertinimą buvo tenkinamos visos sąlygos (žiūrėti 1 priedą): paklaidų autokoreliacijos funkcija svyruoja apie nulį, tik esant nuliniam vėlinimui, autokoreliacijos funkcija yra nelygi nuliui; koreliacija tarp duomenų ir paklaidų taip pat tenkinama, rezultatai patenka į pasikliautinuosius intervalus.

Taip pat buvo atliktas paklaidų vertinimas, kuris susidėjo iš trijų dalių: grafinis, Stjudento kriterijaus ir Šapiro – Wilk kriterijaus (žiūrėti 1 priedą). Pirmiausia, sudarytojo neuroninio tinklo paklaidų pasiskirstymas grafiškai atrodo tik dalinai panašus į normalųjį, todėl privaloma atlikti Šapiro – Wilk hipotezės tikrinimą: ar paklaidos pasiskirsčiusios pagal normalųjį skirstinį? Hipotezė yra neatmetama, esant reikšmingumo lygmeniui 0,05, gauta tikimybė p 0,0508 ($p > \alpha$), kai tikrinami yra tik 50 duomenų. Patikrinus skirtingus duomenų etapus gauta panaši tikimybė ir hipotezė neatmetama. Tada pagal Stjudento kriterijų tikrinama hipotezė, ar paklaidų vidurkis yra lygus nuliui. Atlikus visų paklaidų vertinimą padaroma išvada, kad hipotezė yra neatmetama su 0,5592 tikimybe, esant reikšmingumo lygmeniui 0,05 ($p > \alpha$).

Sudarytas neuroninio tinklo modelis yra adekvatus, todėl panaudojus šį apmokytą modelį toliau naudojame sumodeliuotos laiko eilutės tolimesnių 50 taškų prognozei:



3.27 pav. Sumodeliuotos laiko eilutės prognozė

Atlikta prognozė yra ne tokia gera koks buvo mokymosi etapas, nes paklaidos šiuo atveju svyruoja intervale: $[-5 ; 5]$. Atliktos prognozės paklaidoms taip pat patikrinamos hipotezės, apie paklaidų pasiskirstymą ir paklaidų vidurkį.

Šapiro – Wilk kriterijaus hipotezė:

H_0 : paklaidos pasiskirsčiusios pagal normalųjį skirstinį

H_1 : paklaidos nėra pasiskirsčiusios pagal normalųjį skirstinį

$\alpha = 0.01$

$H = 0$ (hipotezė neatmetama)

$p = 0,0145$

$SWstatistic = 0,9334$

Stjudento kriterijaus hipotezė:

$$H_0: \mu = 0$$

$$H_1: \mu \neq 0$$

$$\alpha = 0.05$$

$$H = 0 \text{ (hipotezė neatmetama)}$$

$$p = 0.0593$$

$$ci = [-0.0209; 1,0453]$$

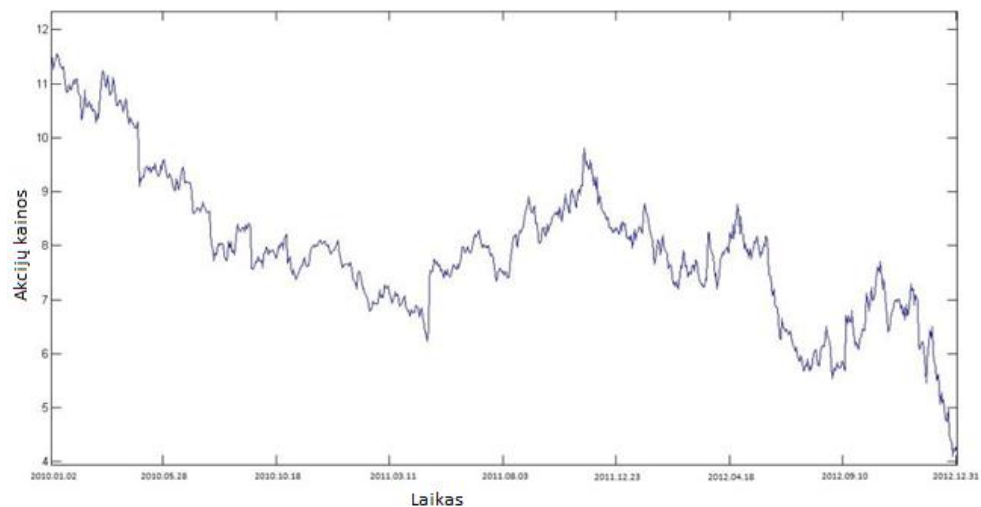
Taigi, sumodeliuotos laiko eilutės prognozės paklaidos yra pasiskirsčiusios pagal normalųjį skirstinį su tikimybe 0,0145, kai reikšmingumo lygmuo 0,01, ir paklaidų vidurkis yra nulis su tikimybe 0,0593, kai reikšmingumo lygmuo yra 0,05.

Išvados. Sudarytosios sumodeliuotos laiko eilutės mažiausia rekonstravimo dimensija nustatyta artimiausio klaidingo kaimyno metodu lygi 3. Tarpusavio informacijos metodu nustatytas laiko vėlinimo parametras τ lygus 6. Su nustatytaisiais parametrais atliktas neuroninio NARX tinklo tyrimas sėkmingai, analizuojamos laiko eilutės paklaidos buvo atitinkančios kriterijus. Apmokius neuroninį tinklą atlikta 50 taškų prognozė, kuomet neatmetamos hipotezės apie paklaidų vidurkį lygų nuliui ir normalųjį skirstinį.

3.3 HP kompanijos akcijų kainų laiko eilutė

Hewlett – Packard kompanija, daugiau žinoma HP trumpiniu, buvo įkurta 1939m. Bilo Havleto. Tai viena galingiausių pasaulio informacinių technologijų korporacijų, kurios pagrindinė veikla – monitorių, asmeninių kompiuterių, nešiojamų kompiuterių, spausdintuvų, serverių (Proliant) bei tinklinės įrangos (ProCurve) gamyba, jiems skirtos programinės įrangos kūrimas.

Šiame darbe toliau buvo nagrinėjama kaip tik šios kompanijos akcijos, imant duomenis nuo 2010.01.02 iki 2012.12.31. Duomenų eilutė buvo sudaryta tik iš fiksuotų duomenų darbo dienomis, kitaip sakant duomenys nebuvo renkami savaitgaliais ir šventinėmis dienomis. Šie duomenys buvo paimti iš prieigos internetu: <http://finance.yahoo.com/>.

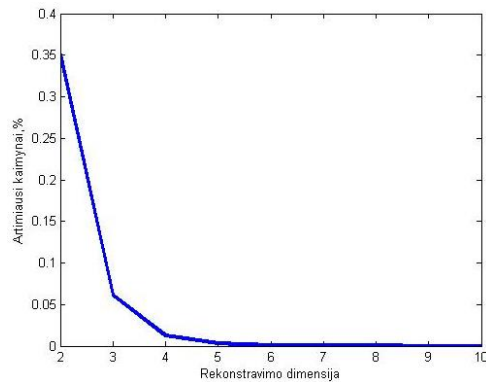


3.28 pav. HP kompanijos akcijų laiko eilutė

Iš aukščiau pateiktojo paveikslėlio matoma, kaip kito HP kompanijos akcijų kainos nuo 2010.01.01 iki 2012.12.31.

3.3.1 HP kompanijos akcijų kainų rekonstravimo dimensijos nustatymas

Artimiausio klaidingo kaimyno metodu analizuojami HP kompanijos akcijų kainos, kintanti nuo 2010 iki 2012 metų.



3.29 pav. HP kompanijos laiko eilutė artimiausio klaidingo kaimyno metodu

Gauti rezultatai rodo, kad analizuojamos laiko eilutės mažiausia rekonstravimo dimensija yra 6. Toliau atliekamas tyrimas kaip kinta rekonstravimo dimensija koreliacinės dimensijos metodu.

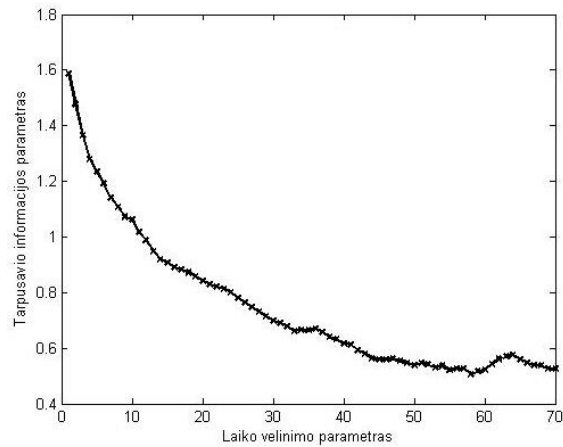
3 lentelė. HP kompanijos akcijos kainų rekonstravimo dimensija, koreliacinės dimensijos metodu

Rekonstravimo dimensija	1	2	3	4	5	6	7	8	9	10
Koreliacinė dimensija	3,1599	4,4054	5,4817	6,3415	6,9514	7,3593	7,6207	7,8220	7,9762	8,10253

Aukščiau pateiktoje 3 lentelėje yra pastebima, kai rekonstravimo dimensija yra 4 ir daugiau koreliacinės dimensijos pokytis, didėjant rekonstravimo dimensijai, yra nedidelis, apytiksliai didėja mažiau nei 1, vadinasi šiuo atveju nustatyta mažiausia rekonstravimo dimensija yra 4.

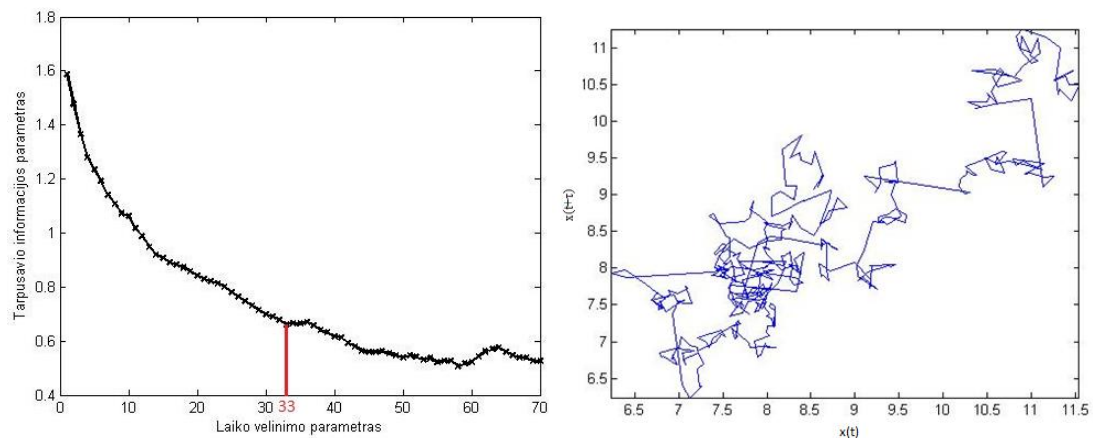
3.3.2 HP kompanijos akcijų kainų laiko vėlinimo nustatymas

Pasinaudojus Tarpusavio informacijos metodu buvo atliktas laiko vėlinimo parametro vertės nustatymas HP kompanijos akcijų kainų laiko eilutei. Pirmiausia analizuojama, kaip atrodo tarpusavio informacijos parametro vertė, kai kinta laiko vėlinimo parametras nuo 1 iki 70:



3.30 pav. HP kompanijos akcijų kainos tarpusavio informacijos metodu

Iš 3.30 paveikslo matoma, kad tarpusavio informacijos parametras tolygiai mažėja, kai didėja laiko vėlinimo parametras ir pirmasis minimumas pastebimas, esant dideliame laiko vėlinimui:



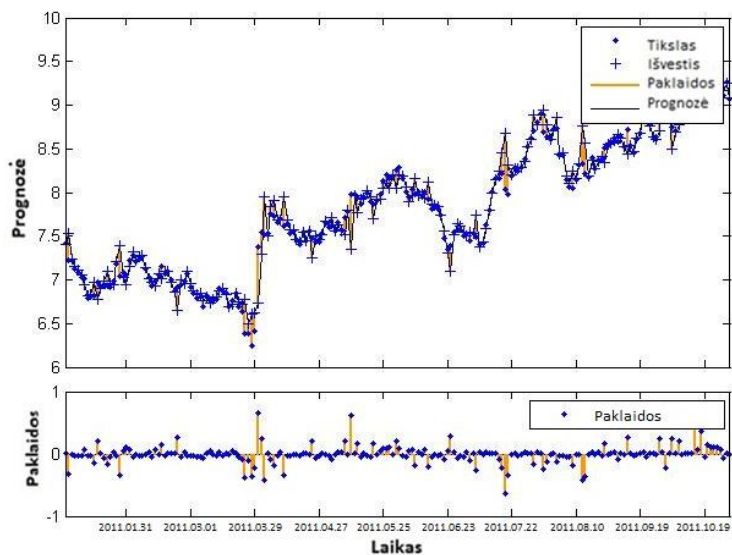
3.31 pav. Rekonstruota HP kompanijos laiko eilutė

Iš 3.31 paveikslo matomas ir pirmasis pasiektas minimumas tarpusavio informacijos metodu, ir kaip atrodo analizuojama laiko eilutė rekonstruota su nustatytu laiko vėlinimo parametru, kuris yra lygus 33. Iš dešiniau esančiojo paveikslo matoma, kad rekonstruota laiko eilutė nėra visiškai išsidėsčiusi į tiesę.

Nustačius laiko vėlinimo parametą toliau analizuojama eilutė neuroniniu NARX tinklo modeliu.

3.3.3 HP kompanijos akcijų kainų prognozavimas suneuroniniu NARX tinklu

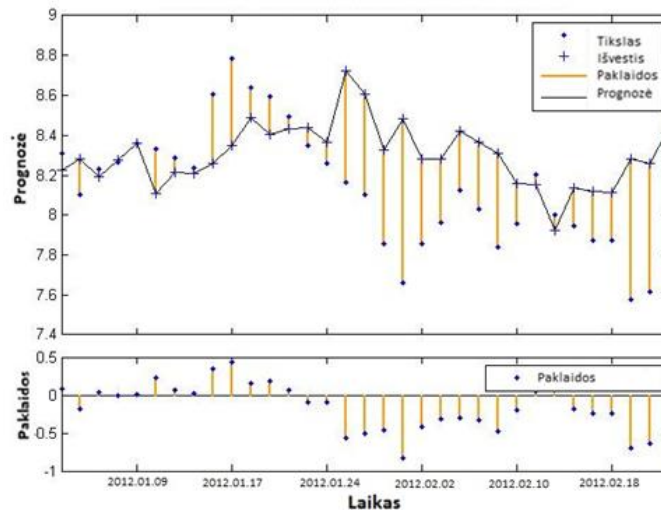
HP kompanijos akcijų kainų laiko eilutės prognozavimas atliktas panaudojus neuroninį NARX tinklą, apsimokymui paėmus 2010 – 2011 metų duomenis, kai 2010 metų duomenys naudojami įvesčiai, o 2011 metų – išvesčiai. Apsimokymo etapas buvo atliktas, kaip ir prieš tai aprašytųjų laiko eilučių, 70 proc. taškų panaudota treniravimuisi ir po 15 proc. – validavimui bei testavimui. Panaudojus tarpusavio informacijos metodo rezultatai, naudojamas laiko vėlinimo parametras, kai $\tau = 33$. Atlikus sluoksnių tyrimą buvo pastebėta, kad mokymosi etapo rezultatai nepriklauso nuo jų skaičiaus, todėl pasirinkta buvo mažiausia vertė, mokymosi etapas atliktas esant 2 sluoksniams:



3.35 pav. HP kompanijos laiko eilutės neuroninio tinklo mokymosi rezultatai

Neuroninio tinklo apmokymas atliktas korektiškai, paklaidos svyruoja tarp -1 ir 1, todėl yra atliekamas papildomas paklaidų tyrimas (žiūrėti 2 priedą). 2 priede yra matoma, kad paklaidų autokoreliacijos funkcija svyruoja apie nulį, bet duomenys su paklaidomis tarpusavyje koreliuoja. Patikrinus Šapiro – Wilk hipotezę, ar paklaidos pasiskirsčiusios pagal normalųjį skirstinį, hipotezė yra neatmetama esant 0,001 reikšmingumo lygmeniui. Taip pat Stjudento kriterijaus hipotezė, ar paklaidų vidurkis yra lygus nuliui, yra neatmetama su tikimybe 0,4638, esant reikšmingumo lygmeniui 0,05. Sudarytasis modelis yra iš dalies tinkamas, tačiau įvertinus, kad dirbama su realiais duomenimis ir, kad neuroninio tinklo analizė atlikta naudojantis originaliais duomenimis, kurie galėjo turėti nukrypimų dėl tam tikrų natūralių veiksnių, sudarytasis modelis yra priimamas ir toliau yra atliekama prognozė.

HP kompanijos akcijų kainų prognozė laikotarpiui nuo 2012.01.03 iki 2012.02.27 (33 laiko eilutės taškai):



3.36 pav. HP kompanijos laiko eilutės prognozė

Iš aukščiau pateikto paveikslo matomos paklaidos nėra didelės, tačiau ne visus taškus prognozuojant gautos paklaidos yra korektiškos, nors Šapiro – Wilk kriterijus ir tenkinamas, bet netenkinama Stjudento kriterijaus hipotezė (žiūrėti 3 priedą). Atlikus taškų statistinį tyrimą pastebėta, kad prognozuoti akcijų kainas galima į priekį 25 taškus, tuomet statistinės hipotezės:

Stjudento kriterijaus hipotezė:

$$H_0: \mu = 0$$

$$H_1: \mu \neq 0$$

$$\alpha = 0.05$$

$$H = 0 \text{ (hipotezė neatmetama)}$$

$$p = 0.0517$$

$$ci = [-0.2548 ; 0.0010]$$

Šapiro – Wilk kriterijaus hipotezė:

$$H_0: \text{paklaidos pasiskirsčiusios pagal normalųjį skirstinį}$$

$$H_1: \text{paklaidos nėra pasiskirsčiusios pagal normalųjį skirstinį}$$

$$\alpha = 0.05$$

$$H = 0 \text{ (hipotezė neatmetama)}$$

$$p = 0,8546$$

$$SW\text{statistic} = 0,9785$$

Taigi, hipotezė apie paklaidų vidurkį lygų nuliui yra priimtina esant 0,05 reikšmingumo lygmeniui su tikimybe 0,0517, o hipotezė apie paklaidų normalųjį skirstinį yra priimtina su tikimybe 0,8546, esant reikšmingumo lygmeniui 0,05. Vadinasi, sudarytasis neuroninis tinklas HP kompanijos akcijų kainų prognozei yra priimtinas prognozuojant ne daugiau kaip 25 taškus.

Išvados. Išnagrinėjus HP kompanijos akcijų kainas pastebėta, kad mažiausią rekonstravimo dimensiją artimiausio kaimyno metodu nustatyta 6, o koreliacinės dimensijos 4. Tarpusavio informacijos metodu nustatytas laiko vėlinimo parametras yra lygus 33. Atlikus prognozavimą panaudojus neuroninį NARX tinklą, modelis gautas adekvatus prognozuojant 25 taškus į priekį, nes paklaidos yra nedidelės, pasiskirsčiusios pagal normalųjį skirstinį, bei hipotezė apie paklaidų lygybę nuliui yra neatmetama.

Išvados

Mažiausia rekonstravimo dimensija Lorencio atraktoriui ir sumodeliuotai laiko eilutei, iš skirtingų fazių ir amplitudžių sinusų funkcijų sumos, geriau nustatyta artimiausio klaidingo kaimyno metodu nei koreliacinės dimensijos metodu, abejoms laiko eilutėms nustatyta dimensija 3. HP kompanijos akcijų kainų geresnę mažiausią rekonstravimo dimensiją nustatė koreliacinės dimensijos metodu, gauta dimensija lygi 4.

Tarpusavio informacijos metodu optimalus laiko vėlinimas nustatytas visoms laiko eilutėms imant pirmąjį tarpusavio informacijos parametro minimumą. Laiko vėlinimo parametras lengvai nustatytas Lorencio atraktoriui ($\tau = 8$) ir sumodeliuotai laiko eilutei ($\tau = 5$). Kiek sunkiau buvo nustatyta HP kompanijos akcijų kainoms, kadangi tarpusavio informacijos parametras tolydžiai mažėjo, kintant laiko vėlinimui. Šiai laiko eilutei nustatytas optimalus laiko vėlinimas 33.

Visoms trimis analizuojamoms laiko eilutėms buvo apmokytas neuroninis NARX tinklas. Lorencio atraktoriui apmokyti neuroninį tinklą prireikia iki 50 iteracijų. Sumodeliuotai laiko eilutei dažnu atveju prireikia maksimalaus skaičiaus iteracijų (1000), kadangi kiekvienos iteracijos metu apmokymas atliekamas vis geriau, visa tai yra todėl kad analizuojama laiko eilutė yra periodinė. HP kompanijos akcijų kainų apmokomas neuroninis tinklas apmokymui sunaudoja iki 15 iteracijų, kadangi duomenys yra chaotiški, originalūs.

Su nustatytais rekonstravimo parametrais, rekonstravimo dimensija ir laiko vėlinimu, bei apmokytais neuroniniais NARX tinklais atliktos visų laiko eilučių prognozės. Kiekvienai laiko eilutei korektiškai prognozuoti į priekį galima skirtingą taškų skaičių: Lorencio atraktoriui – 16, sumodeliuotai laiko eilutei – 50, o HP kompanijos akcijų kainų laiko eilutei – 25. Toliau prognozuoti negalima, nes nebėra tenkinamos statistinės Stjudento ir Šapiro – Wilk hipotezės apie paklaidas.

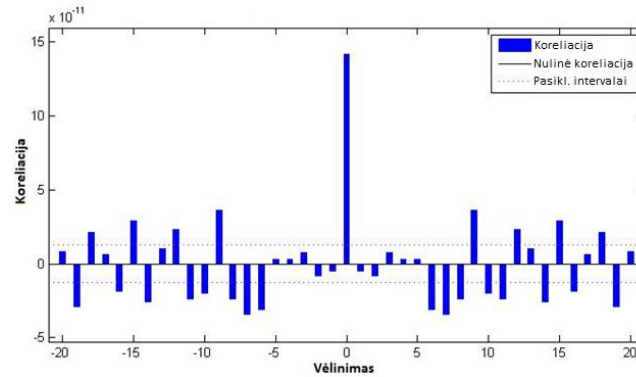
Literatūros sąrašas

1. FENG H., SHU Y., YANG O.W.W. Research on characterization of wireless LANs traffic. Communications in Nonlinear Science and Numerical Simulation. China: 2011, vol. 16 (8), pp. 3179-3187.
2. FRASER A. M. Reconstructing attractors from scalar time series: a comparison of singular system and redundancy criteria. Physica D 34. USA: University of Texas at Austin, 1989, pp. 391-404. 0167-2789/89/\$03.50.
3. ZONGHUA, L. Chaotic time series analysis. Hindawi Publishing Corporation Mathematical Problems in Engineering. China: East China Normal University, 2010, vol. 10(720190), pp. 31. 10.1155/2010/720190
4. ALBOSTAN A., ONOZ B. Implementation of Chaotic Analysis on River Discharge Time Series. Energy and Power Engineering. Turkey: Istanbul Technical University, 2015, vol. 7(73008), pp. 81-82.
5. KOSTELICH E. J., SWINNEY H. L. Practical Considerations in Estimating Dimension from Time Series Data. Physica Scripta. USA, Texas: University of Texas, 1989, vol. 40, pp. 436-441.
6. HANIAS M. P., PANAYIOTIS G. G., MYLONAKIS I. The use of chaos theory predicting the Euribor index. Investment Management and Financial Innovations. Greece: 2008, vol. 5(4).
7. DIACONESCU E. The use of NARX Neural Networks to predict Chaotic Time Series. Computer Research. Romania: Communications and Computer Science Faculty, University of Pitesti, 2008, vol. 3 (3). ISSN: 1991-8755.
8. MEZEIOVA K., KRAKOVSKA A. Choice of Measurement for Phase – Space Reconstruction: Decision Based on False Nearest Neighbors Method. Slovak Grant Agency for Science. Slovakia, Bratislava: Institute of Measurement Science SAS, 2011.
9. KENNEL M. B., BROWN R., ABARBANEL H.D.I. Determining embedding dimension for phase – space reconstruction using a geometrical construction. Physical Review A. California: University of California, 1992, vol. 45 (6), pp. 3403-3411.
10. NAVICKAS Z., STATKUS V., VAINORAS A., GARGASAS L. EKG derivacijų integralinės koreliacinės dimensijos vertinimas. Medicinos Technologija. Kaunas: Kauno Technologijos universitetas, 2005, nr. 5(61). ISSN 1392-9631.
11. MIZRACH B. Determining delay times for phase space reconstruction with application to the FF/DM exchange rate. Journal of Economic Behavior & Organization. USA: Rutgers university, 1996, vol. 30, pp. 369-381.
12. FRASER A. M., SWINNEY H. L. Independent coordinates for strange attractors from mutual information. Physical Review A. Texas: University of Texas at Austin, 1986, vol. 33 (2), pp. 1134-1140.
13. DEMUTH H., BEALE M., HAGAN M. Neural Network Toolbox User's Guide, 2009.
14. LI Z., BEST M. Optimization of the Input Layer Structure for Feed-Forward Narx Neural Networks. Engineering and Technology. USA: World Academy of Science, 2015, vol. 9 (7), pp. 646-651.
15. Čekanavičius V., Murauskas G. Statistika ir jos taikymas, 2006. ISBN: 9789986546931.

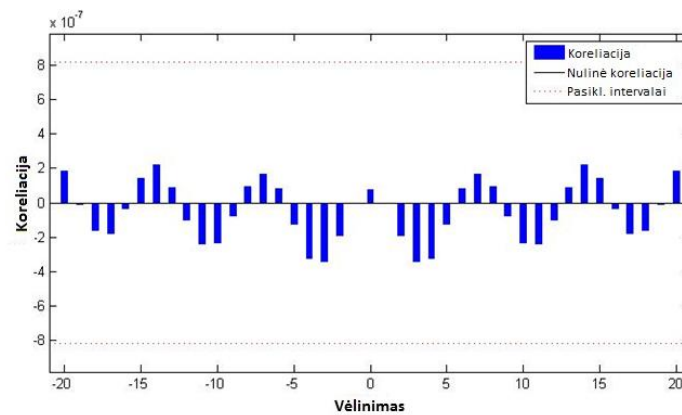
Priedai

1 priedas. Sumodeliuotos laiko eilutės neuroninio NARX tinklo rezultatai

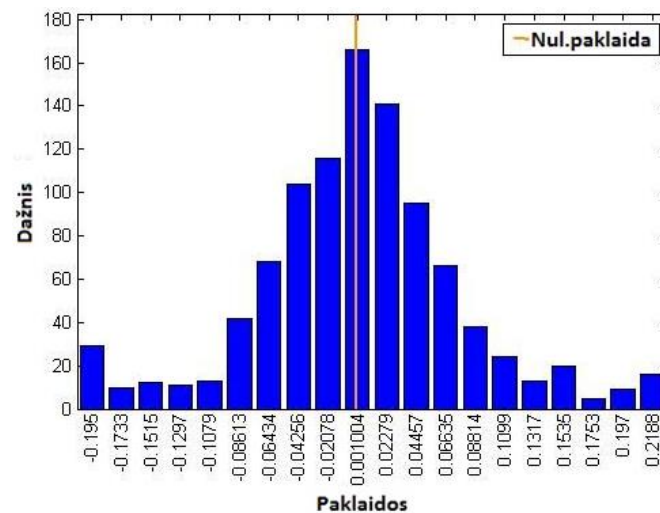
Paklaidų autokoreliacijos grafikas:



Koreliacija tarp duomenų ir paklaidų:



Paklaidų pasiskirstymas:



Šapiro – Wilk kriterijaus hipotezė, kai imami pirmieji 50 taškai:

H_0 : paklaidospasiskirsčiusiospagalnormalųjį skirstinį

H_1 : paklaidosnėrapasiskirsčiusiospagalnormalųjį skirstinį

$\alpha = 0.05$

$H = 0$ (hipotezė neatmetama)

$p = 0,0508$

$SWstatistic = 0,9544$

Stjudento kriterijaus hipotezė, visiems taškams:

$H_0: \mu = 0$

$H_1: \mu \neq 0$

$\alpha = 0.05$

$H = 0$ (hipotezė neatmetama)

$p = 0.5592$

$ci = 1.0e - 06 * [-0.3012; 0.5565]$

$stats =$

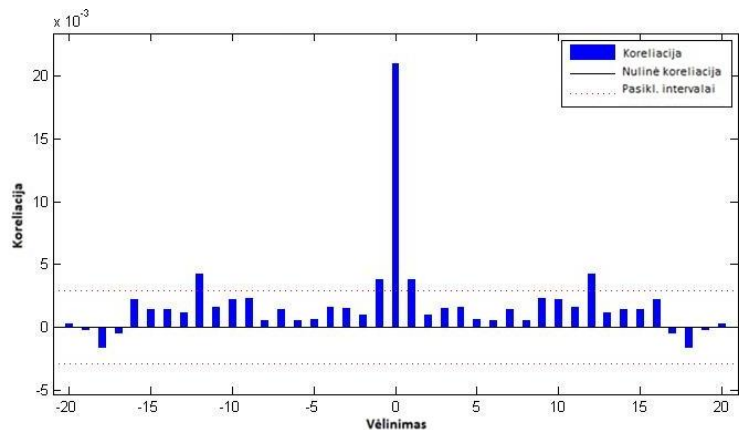
$tstat: 0,5842$

$df: 994$

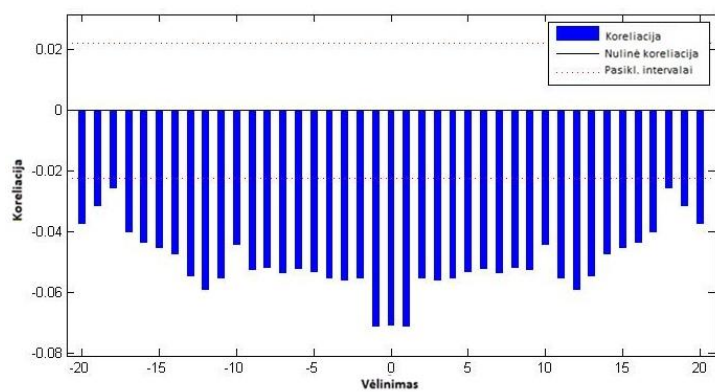
$sd: 6,8931e - 06$

2 priedas. HP akcijos kainų laiko eilutės neuroninio NARX tinklo rezultatai

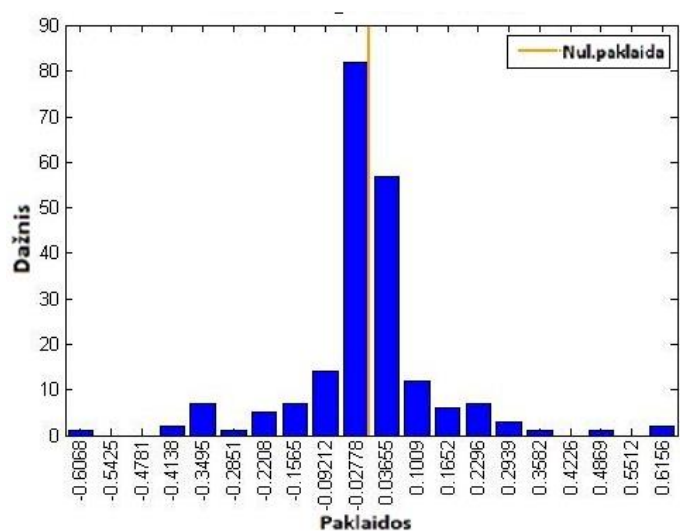
Paklaidų autokoreliacijos grafikas:



Koreliacija tarp duomenų ir paklaidų:



Paklaidų pasiskirstymas:



Šapiro – Wilk kriterijaus hipotezė, kai imami pirmieji 50 taškai:

H_0 : paklaidospasiskirsčiusiospagalnormalųjį skirstinį

H_1 : paklaidosnėrapasiskirsčiusiospagalnormalųjį skirstinį

$\alpha = 0.001$

$H = 1$ (hipotezė atmetama)

$p = 3.4100e - 05$

$SWstatistic = 0,8450$

Stjudento kriterijaus hipotezė, visiems taškams:

$H_0: \mu = 0$

$H_1: \mu \neq 0$

$\alpha = 0.05$

$H = 0$ (hipotezė neatmetama)

$p = 0.4638$

$ci = [-0,0272; 0,0124]$

$stats =$

$tstat: -0.7340$

$df: 207$

$sd: 0,1451$

3 priedas. HP kompanijos akcijų kainų prognozė

Stjudento kriterijaus hipotezė:

$$H_0: \mu = 0$$

$$H_1: \mu \neq 0$$

$$\alpha = 0.001$$

$$H = 0 \text{ (hipotezė neatmetama)}$$

$$p = 0.0036$$

$$ci = [-0.2918; -0.0624]$$

$$stats =$$

$$tstat: -3.1457$$

$$df: 32$$

$$sd: 0,3234$$

Šapiro – Wilk kriterijaus hipotezė:

$$H_0: \text{paklaidos pasiskirsčiusios pagal normalųjį skirstinį}$$

$$H_1: \text{paklaidos nėra pasiskirsčiusios pagal normalųjį skirstinį}$$

$$\alpha = 0.05$$

$$H = 0 \text{ (hipotezė neatmetama)}$$

$$p = 0,6123$$

$$SWstatistic = 0,9745$$

4 priedas. Programos kodai

Laiko eilučių generavimas:

Lorenco atratorius:

```
a=10;
b=28;
c=8/3;

t = 0;
dt = 0.02;

x=1;
y=1;
z=1;

for i = 1:11000

    dx=a*(y-x);
    dy=x*(b-z)-y;
    dz=x*y-c*z;

    xs=x+dx*dt;
    ys=y+dy*dt;
    zs=z+dz*dt;

    xxx = [x xs]; yyy = [y ys]; zzz=[z zs];
    eilute(i,1)=t; eilute(i,2)=x; eilute(i,3)=y; eilute(i,4)=z;
if i>20000
    set(h2,'xdata',xxx,'ydata',yyy,'zdata', zzz);
    drawnow
end
    figure(2);

    t = t + dt;
    x = xs; y = ys; z = zs;

end
disp('ciklas baigtas');

plot3(eilute(:,2),eilute(:,3),eilute(:,4), '-k');
grid on;
axis([-30 30, -30 30 -20 100]);
xlabel ('x');
ylabel ('y');
zlabel ('z');

save(Lorenco11000.txt', 'eilute', '-ASCII')
```

Sumodeliuota laiko eilutė:

```
A = 0.9;
B = 0.6;
om = 0.1;
omm = 0.9;

X = [];
T = [];

for t = 1:3000
for fi = 0:2*pi
    X(t) = A * sin(om*t + fi)+ B * sin(omm*t+ fi);
    T(t) = t;
end
end
```

Rekonstravimo dimensijos nustatymas:

Artimiausio klaidingo kaimyno metodas

Pagrindinis programos kodas yra prieigoje internetu:

<http://www.mathworks.com/matlabcentral/fileexchange/27561-measures-of-analysis-of-time-series-toolkit-mats/content/MATS/FalseNearestNeighbors.m>

FalseNearestNeighbors.m programos šaukinys FNNlorenco.m:

```
load Lorencol1000.dat;
XT = Lorencol1000(:,2);

figure(3)
plot(XT)

% TAU=[1 2 3 4 5 6 7 8 9 10];
TAU = 1;
DIM=[2 3 4 5 6 7 8 9 10];

fnnM=FalseNearestNeighbors(XT,TAU,DIM,10,0);

figure(2)
plot(DIM, fnnM(1,:), 'LineWidth',3);
xlabel('Rekonstravimo dimensija');
ylabel('Artimiausi kaimynai,%');
```

Koreliacinės dimensijos metodas:

```
load Lorencol1000.dat
XT = Lorencol1000;

time = XT(:,1);
duom = XT(:,2);
for i = 301:11000
    duomen(i - 300) = duom(i);
    times(i - 300) = time(i);
end
N = 1000;

figure(1);
plot(times, duomen)

DCR(1) = 0;
dimension(1) = 0;

T = 3;
lgR = -0.4;
R = 10^lgR;

for D = 1:10
    y = [];
    for i = 1:(N - (D - 1) * T)
        for j = 1:D
            y(i,j) = duom(i + (j - 1) * T);
        end
    end
    ilgis = N - (D - 1) * T;
    Nul = ilgis * (ilgis - 1);
    [NN m] = size(y);
    Rsuma = 0;
    norma = 0;
```

```

s = 0;
iksas = zeros(Nul, D);
%iksas = [];
RD = [];

for i = 1:NN
for j = 1:NN
if i ~= j
        s = s + 1;
        iksas(s,:) = abs(y(i,:) - y(j,:));
end
end
end
[K m] = size(iksas);
Maxn = zeros(K,1);
Teta = zeros(K,1);
for i = 1:K
    Maxn(i) = max(iksas(i,:));
    Teta(i) = R - Maxn(i);
if Teta(i) < 0
        Teta(i) = 0;
else Teta(i) = 1;
end
    Rsuma = Rsuma + Teta(i);
end

Rsuma = Rsuma / (ilgis * (ilgis - 1));
lgCDR = log10(Rsuma);

DCR(D + 1) = lgCDR / lgR;
dimension(D + 1) = D;
end

figure(2)
plot(dimension, DCR, 'o-k')
xlabel('d')
ylabel('Dc')

```

Tarpusavio informacijos metodas:

```

load Lorencol1000.dat
XT = Lorencol1000;

figure(1)
plot(XT)

for tau = 1:50;

n = length(XT);

for i = 1:n-tau;
    y = [];
    y1(i) = (XT(i + tau));
    y = y1;
end

YYT = y';
xxt=XT(1:n-tau);
XXT=xxt';

figure(2)
plot(XXT,YYT)
axis tight;

[MMI] = mutualinfo(XXT, YYT);

TAU(tau)=tau;
MI(tau)=MMI;

```

```

XXT=[];
xxt=[];
YYT=[];
y=[];
y1=[];

end

figure(3)
plot(TAU, MI, 'x-k', 'LineWidth',2)
xlabel('Laiko velinimo parametras');
ylabel('Tarpusavio informacijos parametras');

```

Neuroninio NARX modelio sudarymas ir laiko eilutės prognozė:

```

load Lorencol1000.dat;
x = Lorencol1000(8001:9000,2);
y = Lorencol1000(9001:10000,2);

inputSeries = tonndata(x,false,false);
targetSeries = tonndata(y,false,false);

feedbackDelays = 1:8;
hiddenLayerSize = 5;
net = narxnet(inputDelays,feedbackDelays,hiddenLayerSize);

net.inputs{1}.processFcns = {'removeconstantrows','mapminmax'};
net.inputs{2}.processFcns = {'removeconstantrows','mapminmax'};

[inputs,inputStates,layerStates,targets] =
preparets(net,inputSeries,{},targetSeries);

net.divideFcn = 'dividerand';
net.divideMode = 'value';
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

net.trainFcn = 'trainlm'; % Levenberg-Marquardt

net.performFcn = 'mse';

net.plotFcns = {'plotperform','plottrainstate','plotresponse', ...
'ploterrcorr', 'plotinerrcorr'};

[net,tr] = train(net,inputs,targets,inputStates,layerStates);

outputs = net(inputs,inputStates,layerStates);
errors = gsubtract(targets,outputs);
performance = perform(net,targets,outputs)

trainTargets = gmultiply(targets,tr.trainMask);
valTargets = gmultiply(targets,tr.valMask);
testTargets = gmultiply(targets,tr.testMask);
trainPerformance = perform(net,trainTargets,outputs)
valPerformance = perform(net,valTargets,outputs)
testPerformance = perform(net,testTargets,outputs)

```

```

figure, plotperform(tr)
figure, plottrainstate(tr)
figure, plotregression(targets,outputs)
figure, plotresponse(targets,outputs)
figure, ploterrcorr(errors)
figure, plotinerrcorr(inputs,errors)

figure, ploterrhist(errors,'bins',20)

% Prognose
netc = closeloop(net);

xx = Lorenco11000(10001:10100,2);
yy = Lorenco11000(10001:10100,2);

inputSeries2 = tonndata(xx,false,false);
targetSeries2 = tonndata(yy,false,false);

netc = closeloop(net);
netc.name = [net.name ' - Closed Loop'];
[xc,xic,aic,tc] = preparets(netc,inputSeries2,{},targetSeries2);
yc = netc(xc,xic,aic);
closedLoopPerformance = perform(netc,tc,yc)
figure, plotresponse(tc,yc)

```