



VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
INFORMATIKOS INSTITUTAS  
KOMPIUTERINIO IR DUOMENŲ MODELIAVIMO KATEDRA

Signalų analizės pirmas projektinis darbas

## **Rekurentinės diagramos, algortimo parametrų analizė**

Atliko:  
Arnas Vaicekauskas

Vilnius  
2026

## **Turiny**

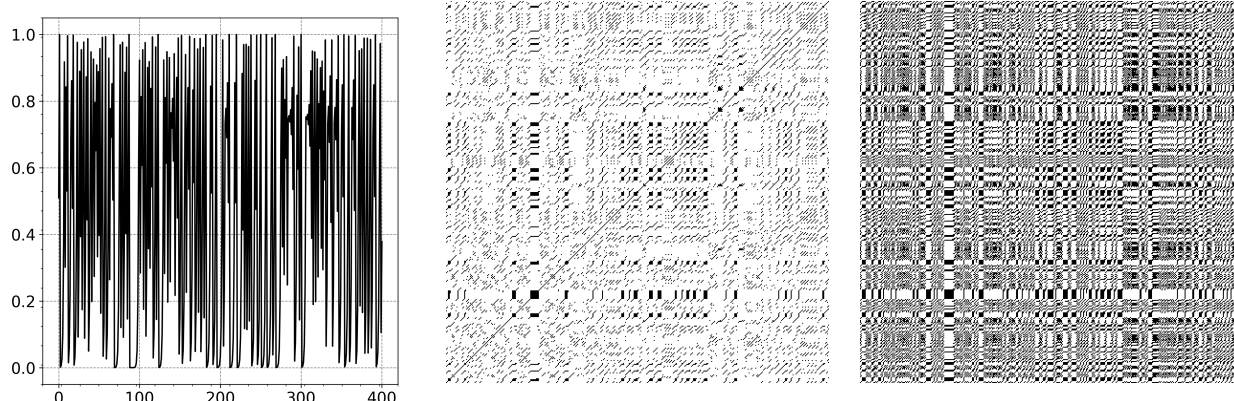
1	Signalų sąrašas .....	3
2	Logistinis atvaizdis .....	3
3	Įmonės „Apple Inc.“ akcijų kaina .....	3
4	Plaučių garsas .....	4
5	Širdies garsas .....	5
6	Oro kokybė .....	6
7	Išvados .....	7
	Literatūros šaltiniai .....	8
	Priedai .....	9

## 1 Signalų sąrašas

1. Logistinis atvaizdis [1]
2. Įmonės „Apple Inc.“ akcijų kainos svyravimai [2]
3. Klinikinio manekeno plaučių garso įrašas pro stetoskopą [3]
4. Klinikinio manekeno širdies garso įrašas pro stetoskopą [3]
5. Sensoriaus duomenys apie anglies monoksido koncentraciją ore nuo 1970 iki 1995 [4]

## 2 Logistinis atvaizdis

Pav. 1: Logistinio atvaizdžio  $x_{n+1} = 4x_n(1 - x_n)$ ,  $x_0 = 0.51$  chaotinis signalas (kairėje) ir jo rekurentinės diagramos su parametrais  $r = 0.1$  (centre),  $r = 0.5$  (dešinėje). Signalo imčių skaičius  $N = 400$ .

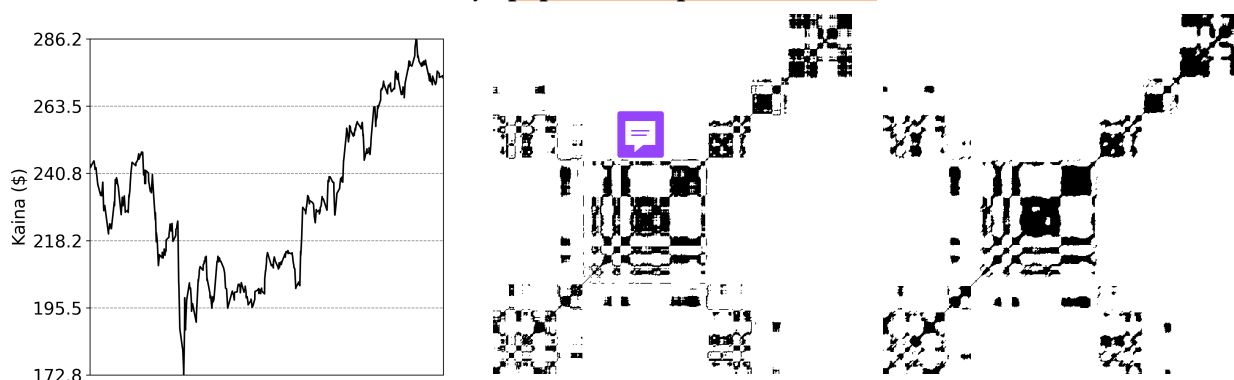


Pav. 1 pateiktas logistinio atvaizdžio signalas yra skirtas parodymui, kad rekurentinių diagramų algoritmas yra įgyvendintas korektiškai. Šiuos rezultatus reikėtų lyginti su kurso medžiagoje [1] pateiktomis logistinio atvaizdžio rekurentinėmis diagramomis su tokiais pačiais parametrais. Kaip pažymėta kurso medžiagoje, trumpos atkarpos paralelios centrinei linijai sufleruoja, kad signalo struktūroje yra panašumų, tačiau jie yra trumpi ir greit nyksta, dėl to diagrama gana panaši į visiškai atsitiktinio signalo.

## 3 Įmonės „Apple Inc.“ akcijų kaina

**Pav. 2:** „Apple Inc.“ akcijų kainos svyravimai 2025 m. Duomenys gauti naudojant yfinance python paketą [2]. Rekurentinės diagramos gautos naudojant parametrus  $p = 0.1$ ,  $d = 1$ , kur  $p$  yra vartotojo pasirinktas juodų ir baltų taškų santykis diagramoje. Diagramoms gauti naudota Euklidinė norma. Centrinės diagramos parametras  $D = 1$ , o esančios dešinėje  $D = 5$ . Signalo imčių skaičius  $N = 494$  su 4 val. tarpais (neskaitant laiko, kada akcijų birža nedirba).

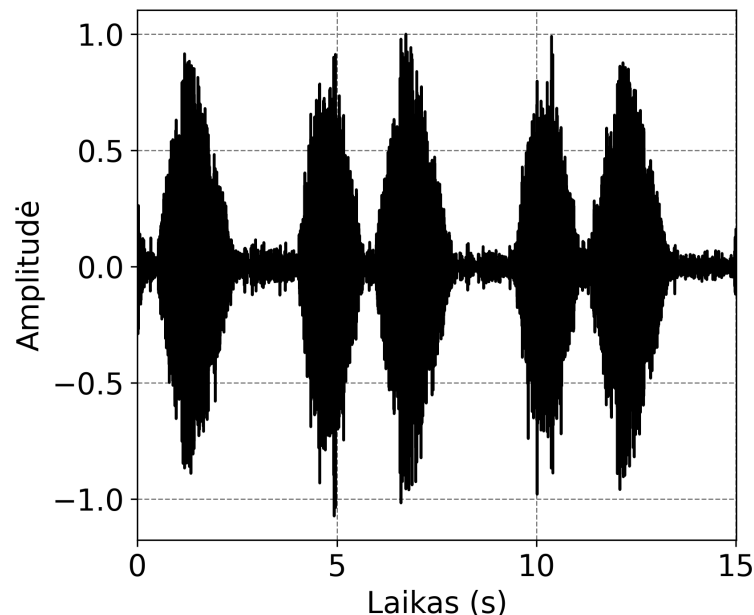
Duomenys papildomai apdoroti nebuvo.



Pav. 2 pateiktos rekurentinės diagramose matome nehomogeniškai pasiskirsčiusius juodus taškus, signalas yra nestacionarus. Diagramose taip pat pastebime vertikalių ir horizontalių linijų, tai rodo, kad signale yra atkarpų per kurias jis kinta nežymiai palyginus su visa signalo imtimi. Diagrama su didesniu  $D$  parametru pasireiškia glotnesniais raštais, kurie primena Gauso suliejimą (*angl. Gaussian blur*); šio parametro didinimas padeda atsikratyti triukšmo.

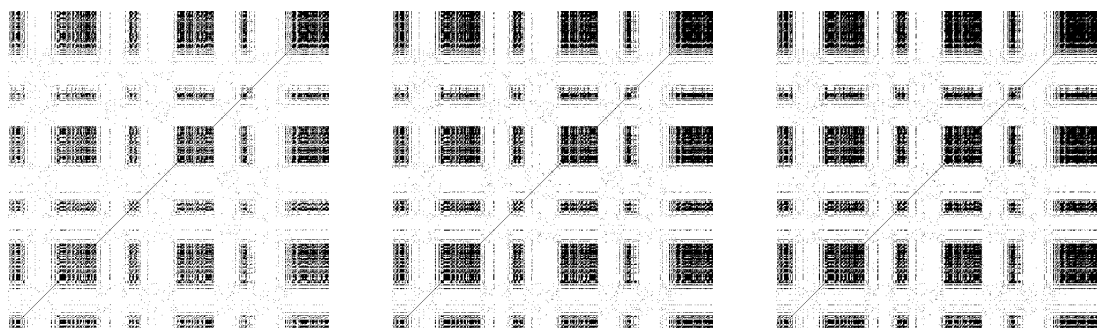
#### 4 Plaučių garsas

Pav. 3: Plaučių garso įrašo signalas



Pav. 3 matome pavyzdinį sveikos moters dešiniojo viršutinio priekinio plaučio garso įrašą paimtą iš širdies ir plaučių duomenų rinkinio įrašyto su klinikiu manekenu naudojant skaitmeninį stetoskopą [3]. Garso įrašo diskretizavimo dažnis (*angl. sampling rate*) yra 4000Hz. Prieš rekurentinių diagramų generavimą duomenys buvo praretinti 100 kartų. Rezultate diagramos buvo generuojamos iš signalo, kurio imčių skaičius yra  $N = 599$ .

Pav. 4: Plaučių garso įrašo rekurentinės diagramos su parametrais  $D = 2$ ,  $r = 0.1$ ,  $d = 1$  ir skirtingomis normomis (iš kairės į dešinę): Manheteno, Euklidinė ir maksimumo.



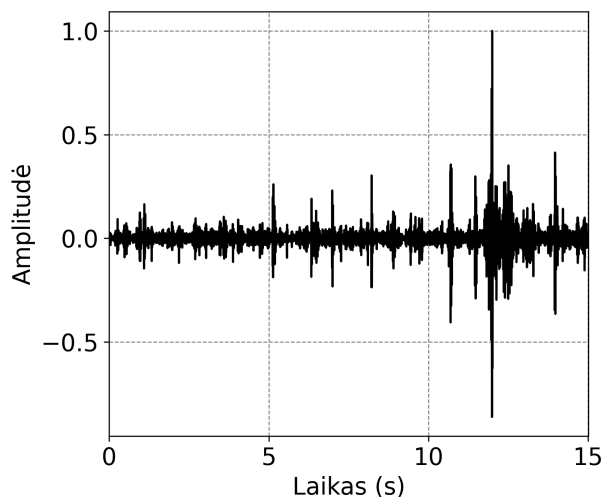
Pav. 4 pavaizduotos diagramos pasireiškia nehomogeniškų juodų taškų pasiskirstimu, tai reiškia, kad signalas yra nestacionarus. Taip pat matome daugelį juodų taškų telkinių, kurie signalizuoja (pun intended), kad tarpais signalo būseną nekinta ir taip iš ties galime patyti net

pačiame signale – iš viso turime šešis intervalus, kuriuose signalo reikšmė atrodo pastovi ir tai sutampa su rekurentinėje diagramoje matomu skaičiumi telkinių (horizontaliai).

Šiame pavyzdyje bandome keisti normą, o kitus parametrus laikyti pastoviais. Kaip ir nurodytą mokymo priemonėje [1] **kokybinio skirtumo nėra**, tačiau kiekvieną diagrama turi vis daugiau juodų taškų, tai nėra optinė apgaulė, juodų **taškų koncentraciją** pavyzdyje yra (iš kairės į dešinę): ~13%, ~18% ir ~21%.

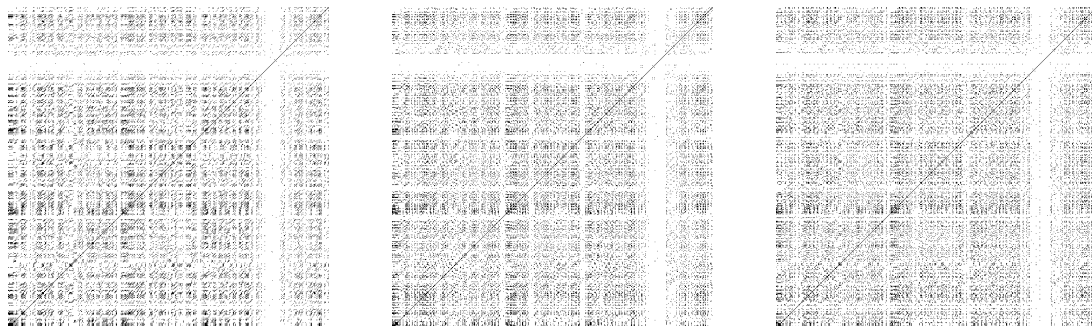
## 5 Širdies garsas

Pav. 5: Širdies garso įrašo signalas



Pav. 5 matome pavyzdinį sveiko vyro dešinėsios viršutinės krūtinkaulio kraštinės (širdies) garso įrašą paimtą iš širdies ir plaučių duomenų rinkinio įrašyto su klinikiu manekenu naudojant skaitmeninę stetoskopą [3]. Garso įrašo diskretizavimo dažnis (*angl. sampling rate*) yra 4000Hz. Prieš rekurentinių diagramų generavimą duomenys buvo praretinti 100 kartų. Rezultate diagramos buvo generuojamos iš signalo, kurio imčių skaičius yra  $N = 599$ .

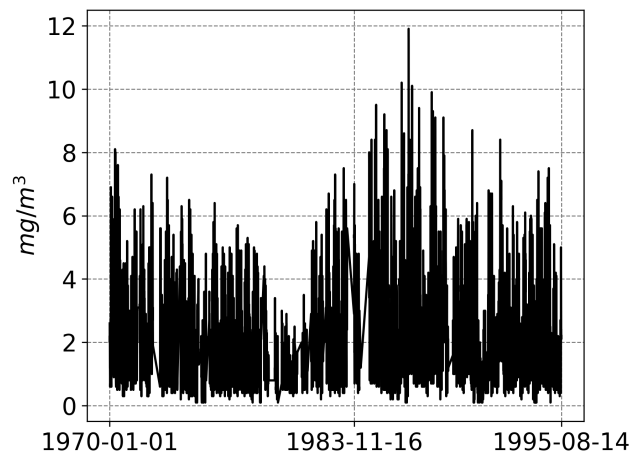
Pav. 6: Širdies garso įrašo rekurentinės diagramos su parametrais  $D = 2, p = 0.1$  (vartotojo pasirinktas juodų taškų procentas diagramoje) ir Euklidine norma. Keičiamas parametras  $d$  (iš kairės į dešinę): 2, 5, 7



Pav. 6 matome, kad širdies garso signalas pasižymi beveik homogenišku juodų taškų išsidėstymu, kas žymi signalo stacionarumą. Išsidėstymas turi nežymią tvarką, kuri nyksta didinant parametą  $d$  ir kartu su homogenišku juodų taškų pasiskirstymu žymi vyraujančią atsitiktinį triukšmą.

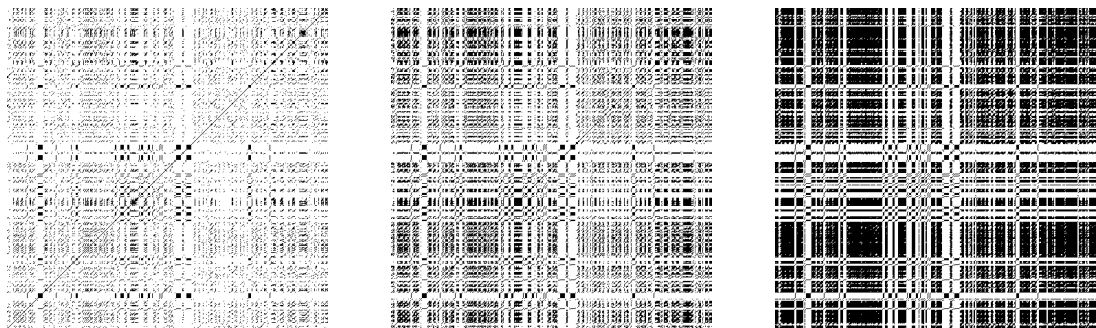
## 6 Oro kokybė

Pav. 7: Anglies monoksido koncentracijos ore duomenys iš sensoriaus, kuris buvo pastatytas Italijos miestelio laukuose. Duomenis gauti iš oro kokybės duomenų rinkinio [4].



Šio signalo imčių skaičius yra  $N = 9356$ , todėl prieš rekurentinių diagramų generavimą duomenys buvo praretinti 20 kartų.

Pav. 8: Rekurentinės diagramos sugeneruotos iš anglies monoksido koncentracijos signalo. Parametrai  $D = 2$ ,  $d = 1$ , naudojama Euklidinė metrika. Keičiamas parametras  $p$ , kuris žymi vartotojo pasirinktą juodų taškų procentą diagramoje (iš kairės į dešinę): 0.1, 0.25, 0.5



Pav. 8 matome, kad keičiant parametą  $p$  diagramose galima išvysti kokybinių skirtumų – kairėje diagramoje juodi taškai beveik homogeniškai be tvarkos pasiskirstę po erdvę, kas rodo, kad signale yra atsitiktinio triukšmo, tačiau matyti ir juodų taškų telkinių, kurie ryškėja didinant parametą  $p$ , kas rodo, kad signalo būseną laiko tarpais beveik arba visiškai nekinta.

## 7 Išvados

Rekurentinių diagramų sudarymo algoritmas turi kelis parametrus, kiekvienas iš kurių turi savo unikalų poveikį sugeneruotoms diagramoms.

- $r$  parametras nustato slenkstinę reikšmę dviejų signalo atkarpų lyginimui ir nustato ar taškas turėtų būti spalvinamas juodai arba baltai. Tinkamai pasirinkti šio signalo reikšmę yra labai svarbu, nes tai gali turėti didelį poveikį diagramos išvaizdai, pavyzdžiui galėtume laikyti logistinio atvaizdžio diagramą Pav. 1, kur maža  $r$  reikšmė atskleidžia homogenišką ir netvarkingą taškų pasiskirstymą, kuris nėra toks akivaizdus pažvelgus į diagramą su didesne  $r$  parametro reikšme.
- $D$  parametras nustato kiek signalo taškų lyginsime sprendami vieno taško spalvą. Šio parametro didinimas turi glotninantį efektą panašų į Gauso suliejimą. Tai akivaizdžiai matome diagramoje gautoje iš įmonės „Apple Inc.“ akcijų svyravimo Pav. 2.
- $d$  parametras reguliuoja atstumą tarp signalo taškų, kuriuos renkamės lyginimui, kai  $d = 1$  imsime taškus, kurie yra šalia vienas kito, kai  $d = 2$  imsime kas antrą tašką ir t. t. Šio parametro didinimas turi dvejetainį efektą dėl kurio gali pranykti diagramos struktūra. To pavyzdys galėtų būti Pav. 6, kuriame matome, kad diagrama gauta naudojant  $d = 2$  turi vos matomų vertikalių ir horizontalių baltų linijų, tačiau padidinus  $d$  iki 5, vaizdas tampa mažiau aiškus.
- $p$  parametras nėra apibrėžtas klasikiniame rekurentinių diagramų algoritme, o yra tik šios uždouties dalis. Jis kontroliuoja procentą juodų taškų skaičiaus diagramoje. Algoritmo įgyvendinime parametras  $p$  yra naudojamas tik parametro  $r$  apskaičiavimui, todėl kokybinis šio parametro didinimo efektas yra toks pat kaip parametro  $r$ . Tai gana aiškiai matosi iliustracijose Pav. 1 ir Pav. 8. Iš praktinės pusės, šį parametą naudoti yra patogiau nei parametą  $r$ , todėl, kad ne visada galime lengvai žinoti kokia slenkstinė reikšmė bus tinkama. Taip pat todėl, kad naudojant parametą  $p$  yra lengviau numatyti algoritmo rezultatą.
- Norma taip yra vienas iš algoritmo parametrų, kurį galime pasirinkti. Praktinis efektas diagramoms yra skirtingas juodų taškų skaičius. Kai naudojama Manheteno norma, juodų taškų bus šiek tiek mažiau nei tuomet, jei būtų naudojama Euklidinė norma, o daugiausiai juodų taškų atsiranda tada, kai naudojama maksimumo norma. Tai galima matyti iliustracijoje 4.

## Literatūros šaltiniai

- [1] T. Meškauskas, „Signalų analize ir apdorojimas“. 2021 m.
- [2] R. Aroussi, „yfinance: Yahoo! Finance market data downloader“. GitHub, 2024 m.
- [3] S. S. Torabi Yasaman ir J. P. Reilly, „HLS-CMDS: Heart and Lung Sounds Dataset Recorded from a Clinical Manikin using Digital Stethoscope“. 2025 m.
- [4] S. Vito, „Air Quality“. 2008 m.



## Priedai

Dokumentą sudaro du priedai Kodo fragmentas 1 ir Kodo fragmentas 2.

Kodo fragmentas 1: Klasikinio rekurentinių diagramų algoritmo įgyvendinimas

```
from math import ceil
from typing import Callable, Literal

def recurrence_plot(
    f: list[float],
    D: int,
    r: float,
    d: int = 1,
    ord: Literal[1, 2, 'inf'] = 2
) -> tuple[int, list[bool]]:
    size, field_values = recurrence_field_values(f, D, d, ord)
    return (size, [ v > r for v in field_values ])

def recurrence_field_values(
    f: list[float],
    D: int,
    d: int = 1,
    ord: Literal[1, 2, 'inf'] = 2
) -> tuple[int, list[float]]:

    assert len(f) > 0

    N: int = len(f) - 1
    M: int = N - (D - 1) * d
    field_values: list[float] = [ 0 ] * ( (M + 1) * (M + 1) )

    norm_fn_lookup: dict[Literal[1, 2, 'inf'], Callable] = {
        1: lambda xs: sum([ abs(x) for x in xs]),
        2: lambda xs: sum([ x**2 for x in xs]) ** 0.5,
        'inf': lambda xs: max([ abs(x) for x in xs])
    }

    norm_fn: Callable = norm_fn_lookup[ord]

    for i in range(M + 1):
        for j in range(M + 1):
            y_i = [ f[i + k * d] for k in range(D) ]
            y_j = [ f[j + k * d] for k in range(D) ]
            diff = [ a - b for a, b in zip(y_i, y_j) ]
            field_values[j + i * (M + 1)] = norm_fn(diff)

    return (M + 1, field_values)
```

Kodo fragmente 1 pateiktas klasikinio rekurentinių diagramų algoritmo įgyvendinimas python programavimo kalba nenaudojant jokių išorinių pagalbinių paketų išskyrus standartinės bibliotekos pagalbines funkcijas Ceil ir tipų aprašymo konstrukcijas Callable ir Literal. Didelė algoritmo dalis yra iškelta į metodą recurrence\_field\_values, kuris yra naudojamas įgyvendinant ir pritaikytą algoritmą, kuris palaiko parametą  $p$ .

Kodo fragmentas 2: Pritaikyto rekurentinių diagramų algoritmo įgyvendinimas

```
def recurrence_plot_custom(
    f: list[float],
    D: int,
    p: float,
    d: int = 1,
    ord: Literal[1, 2, 'inf'] = 2
) -> tuple[int, list[bool]]:
    size, field_values = recurrence_field_values(f, D, d, ord)
    r = choose_r(field_values, p)
    return (size, [ v > r for v in field_values ])

def choose_r(values: list[float], p: float) -> float:
    k = ceil(p * len(values))
    return sorted(values)[k - 1]
```

Kodo fragmente 2 pateiktas pritaikyto algoritmo įgyvendinimas. Palaikomas parametras  $p$ , kuris nustato juodų taškų procentą diagramoje. Panaudota praeitame priede 1 pateikta funkcija `recurrence_field_values`.