



**Matematikos ir
informatikos
fakultetas**

**VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
FINANSŲ IR DRAUDIMO MATEMATIKOS MAGISTRO STUDIJŲ PROGRAMA**

**Gyvenimo kokybės indekso analizė Lietuvos
savivaldybėse**

**Analysis of the Quality of Life Index in Lithuanian
Municipalities**

Baigiamasis magistro darbas

Autorius: Ugnė Linkutė

VU el. paštas: ugne.linkute@mif.stud.vu.lt

Darbo vadovas: Doc. Dr. Martynas Manstavičius

**Vilnius
2024**

Turinys

Iliustracijų sąrašas	1
Lentelių sąrašas	2
1 Įvadas	6
2 Teorinė dalis	8
2.1 Kintamieji	8
2.2 Metodologija	9
2.2.1 Paneliniai duomenys	9
2.2.2 Testai	9
2.2.3 ARIMA modelis	9
2.2.4 GAM modelis	12
2.2.5 LSTM modelis	13
2.2.6 BiLSTM modelis	14
2.2.7 Modelio tikslumą nusakančios metrikos	15
2.2.8 Gyvenimo kokybės indekso formulė	15
2.2.9 Monte Karlo simuliacijos	16
3 Praktinė dalis	17
3.1 Duomenų aprašymas	17
3.2 Laiko eilučių modelių pritaikymas	20
3.2.1 ARIMA modelio pritaikymas	20
3.2.2 GAM modelio pritaikymas	23
3.2.3 LSTM modelio pritaikymas	30
3.2.4 BiLSTM modelio pritaikymas	35
3.3 Prognozavimas	39
3.3.1 ARIMA modelio prognozės	39
3.3.2 GAM modelio prognozės	45
3.3.3 LSTM modelio prognozės	49
3.3.4 BiLSTM modelio prognozės	55
4 Išvados bei rekomendacijos	62
Literatūra	65
5 Priedai	66

Iliustracijų sąrašas

1	Mažiausių prognozuotų ir tikrų reikšmių pagal ARIMA modelį palyginimas	44
2	Didžiausių prognozuotų ir tikrų reikšmių pagal ARIMA modelį palyginimas	44
3	Mažiausių prognozuotų ir tikrų reikšmių pagal GAM modelį palyginimas	49
4	Didžiausių prognozuotų ir tikrų reikšmių pagal GAM modelį palyginimas	50
5	Mažiausių prognozuotų ir tikrų reikšmių pagal LSTM modelį palyginimas	55
6	Didžiausių prognozuotų ir tikrų reikšmių pagal LSTM modelį palyginimas	56
7	Mažiausių prognozuotų ir tikrų reikšmių pagal BiLSTM modelį palyginimas	61
8	Didžiausių prognozuotų ir tikrų reikšmių pagal BiLSTM modelį palyginimas	62

Lentelių sąrašas

1	ARIMA modelio rezultatai visiems subindeksams pagal mažiausią GKI turinčias savivaldybes	22
2	ARIMA modelio rezultatai visiems subindeksams pagal didžiausią GKI turinčias savivaldybes	24
3	GAM modelio rezultatai subindeksams pagal mažiausią GKI turinčias savivaldybes	25
4	GAM modelio rezultatai subindeksams pagal didžiausią GKI turinčias savivaldybes	26
5	GAM modelių įverčiai pagal mažiausią GKI turinčias savivaldybes ir 95% pasikliovimo intervalai	27
6	GAM modelių įverčiai pagal didžiausią GKI turinčias savivaldybes ir 95% pasikliovimo intervalai	29
7	LSTM modelio mokymosi nuostoliai per 5 epochas ir procentinis sumažėjimas mažiausią GKI turinčioms savivaldybėms	32
8	LSTM modelio mokymosi nuostoliai per 5 epochas ir procentinis sumažėjimas didžiausią GKI turinčioms savivaldybėms	34
9	BiLSTM modelio mokymosi nuostoliai per 5 epochas ir procentinis sumažėjimas mažiausią GKI turinčioms savivaldybėms	36
10	BiLSTM modelio mokymosi nuostoliai per 5 epochas ir procentinis sumažėjimas didžiausią GKI turinčioms savivaldybėms	38
11	ARIMA modelio prognozavimo rezultatai su 95% prognozių pasikliovimo intervalais pagal mažiausią GKI turinčias savivaldybes	40
12	ARIMA modelio prognozavimo rezultatai su 95% prognozių pasikliovimo intervalais pagal didžiausią GKI turinčias savivaldybes	42
13	ARIMA modelio metrikų rezultatai	43
14	Mažiausių prognozuotų ir tikrų reikšmių pagal ARIMA modelį palyginimas . . .	43
15	Didžiausių prognozuotų ir tikrų reikšmių pagal ARIMA modelį palyginimas . . .	45
16	GAM modelių prognozės pagal mažiausią GKI turinčias savivaldybes ir jų 95% pasikliovimo intervalai	46
17	GAM modelių prognozės pagal didžiausią GKI turinčias savivaldybes ir jų 95% pasikliovimo intervalai	47
18	GAM modelio metrikų rezultatai	48
19	Mažiausių prognozuotų ir tikrų reikšmių pagal GAM modelį palyginimas	48
20	Didžiausių prognozuotų ir tikrų reikšmių pagal GAM modelį palyginimas	49
21	LSTM modelių prognozės pagal mažiausią GKI turinčias savivaldybes ir jų 95% pasikliovimo intervalai	51
22	LSTM modelių prognozės pagal didžiausią GKI turinčias savivaldybes ir jų 95% pasikliovimo intervalai	53
23	LSTM modelio metrikų rezultatai	54
24	Mažiausių prognozuotų ir tikrų reikšmių pagal LSTM modelį palyginimas	54
25	Didžiausių prognozuotų ir tikrų reikšmių pagal LSTM modelį palyginimas	55
26	BiLSTM modelių prognozės pagal mažiausią GKI turinčias savivaldybes ir jų 95% pasikliovimo intervalai	57
27	BiLSTM modelių prognozės pagal didžiausią GKI turinčias savivaldybes ir jų 95% pasikliovimo intervalai	59
28	BiLSTM modelio metrikų rezultatai	60
29	Mažiausių prognozuotų ir tikrų reikšmių pagal BiLSTM modelį palyginimas . . .	60
30	Didžiausių prognozuotų ir tikrų reikšmių pagal BiLSTM modelį palyginimas . . .	61

Gyvenimo kokybės indekso analizė Lietuvos savivaldybėse

Santrauka

Šis magistro darbas yra bakalauro darbo, pavadinimu „Lietuvos savivaldybių finansinių rodiklių analizė“, tęsinys. Magistro darbe yra nagrinėjamos 10 Lietuvos savivaldybių, turinčių mažiausią bei didžiausią Gyvenimo kokybės indeksą (GKI) 2022 metais.

Šiame magistro darbe yra siekiama prognozuoti Gyvenimo kokybės indeksą bei jį sudarančius subindeksus. Tyrime naudojami paneliniai duomenys apie Lietuvos savivaldybių Gyvenimo kokybės indeksą bei jį sudarančius 6 subindeksus nuo 2013 iki 2022 metų. Duomenys buvo gauti iš Lietuvos Respublikos finansų ministerijos administruojamos svetainės www.lietuvosfinansai.lt. Duomenų analizei ir prognozių rezultatams gauti buvo naudojamos „RStudio“ bei „Python“ programos.

Darbo tikslas yra išnagrinėti 5 Lietuvos savivaldybes pagal žemiausią bei 5 pagal aukščiausią gyvenimo kokybės indeksą 2022 metais bei pateikti prognozes ir jas palyginti su tikromis reikšmėmis.

Duomenų analizei taikomi šie metodai: ARIMA modelis, GAM modelis, LSTM modelis, BiLSTM modelis, ADF testas, Ljung–Box testas, Durbin–Watson testas, Jarque–Bera testas, Breusch–Pagan testas, Shapiro–Wilk testas, ARCH LM testas, GKI formulė.

Atlikus analizę gauta, kad geriausiai 2022 metų subindeksų ir GKI reikšmes prognozavo ARIMA modelis. Šiuo modeliu prognozuotos reikšmės labiausiai atitiko realius duomenis.

Raktiniai žodžiai: GKI, Paneliniai duomenys, ARIMA modelis, GAM modelis, LSTM modelis, BiLSTM modelis.

Analysis of the Quality of Life Index in Lithuanian Municipalities

Abstract

This master's thesis is a continuation of the bachelor's thesis entitled „Analysis of financial indicators of Lithuanian municipalities”. The master's thesis examines 10 Lithuanian municipalities with the lowest and highest Quality of Life Index (QLI) in year 2022.

This master's thesis aims to forecast the Quality of Life Index and its constituent subindices. The study uses panel data on the quality of life index of Lithuanian municipalities and its 6 sub-indices from years 2013 to 2022. The data was obtained from the website www.lietuvosfinansai.lt administered by the Ministry of Finance of the Republic of Lithuania. „RStudio” and „Python” programs were used for data analysis and prediction results.

The aim of this work is to examine the 5 Lithuanian municipalities according to the lowest and 5 according to the highest quality of life index in 2022, and to present forecasts and compare them with real values.

The following methods were used for data analysis: ARIMA model, GAM model, LSTM model, BiLSTM model, ADF test, Ljung–Box test, Durbin–Watson test, Jarque–Bera test, Breusch–Pagan test, Shapiro–Wilk test, ARCH LM test, QLI formula.

After the analysis, it was found that ARIMA model predicted the values of sub-indices and QLI in 2022 the best. The values predicted by this model was the most consistent with the real data.

Key words: QLI, Panel data, ARIMA model, GAM model, LSTM model, BiLSTM model.

Pažymėjimai

- GKI – Gyvenimo kokybės indeksas.

1 Įvadas

Visų šalių ekonomika pasaulyje yra nuolat besikeičiantis ir besiformuojantis procesas. Ekonomika kinta nuo daugelio veiksnių, ypatingai nuo šalies gyventojų poreikių ir kiekvienas iššūkis priverčia ją prisitaikyti ir evoliucionuoti. Ne paslaptis, jog šalies ekonomika yra glaudžiai susijusi su šalies finansais ir su Gyvenimo kokybės indeksu. Gyvenimo kokybės indeksas stipriai priklauso nuo kiekvienos šalies ekonominės padėties ir yra daug kur plačiai aptarinėjamas, kaip vienas reikšmingiausių rodiklių, kalbant apie šalies finansinį stabilumą ir gyventojų socialinę gerovę. Ne išimtis yra ir Gyvenimo kokybės indeksas Lietuvos savivaldybėse. Skirtingose Lietuvos savivaldybėse GKI skiriasi net po kelis kartus. Didžiausi skirtumai pastebimi tarp didžiųjų ir mažųjų, nutolusių nuo didžiųjų miestų, savivaldybių. Pasak autorių Raúl López-Blanco, Juan Herranz Martín, Ricardo S. Alonso ir Javier Prieto, gyvenimas mieste pastaruoju laikotarpiu tampa vis dominuojančia tendencija mažųjų miestų sąskaita ir yra tikimasi, kad 2050 metais net 68% pasaulio žmonių populiacijos gyvens miestuose arba didžiuosiuose miestuose [6]. Dėl šių priežasčių didžiuosiuose miestuose ir jų savivaldybėse Gyvenimo kokybės indekso rodiklis yra žymiai aukštesnis. Tuo tarpu mažuosiuose miestuose ir jų savivaldybėse GKI yra žemesnis ir tai sudaro ryškius regioninius skirtumus šalyje. Šiame darbe bus analizuojamos 5 Lietuvos savivaldybės, turinčios aukščiausią GKI 2022 metais bei 5 savivaldybės, turinčios žemiausią GKI tais pačiais metais. Savivaldybių GKI prognozės 2022 metams bus sudarinėjamos su 4 laiko eilučių modeliais – Autoregresiniu integruotu slenkančio vidurkio modeliu (ARIMA), Apibendrintuoju adityviuoju modeliu (GAM), Ilgalaiškės trumpalaikės atminties modeliu (LSTM) ir Dvikrypčiu ilgalaiškės trumpalaikės atminties modeliu (BiLSTM). Pirma prognozuojami bus Gyvenimo kokybės indeksą sudarantys 6 subindeksai, tuomet iš jų, padaugintų pagal kiekvieno subindekso svorį, sumos bus gaunamas prognozuotas GKI. Gavus prognozes, šios reikšmės bus lyginamos su tikromis to meto GKI reikšmėmis, norint ištirti, kurio modelio prognozės geriausiai atspindėjo realias duomenų reikšmes.

Šiame darbe bus prognozuojami subindeksai ir bendras GKI 2022 metams 10–iai Lietuvos savivaldybių, pagal žemiausias bei aukščiausias GKI reikšmes. Taip pat bus pateiktos rekomendacijos, kaip savivaldybės gali pagerinti bendrą GKI, arba bent jau jį išlaikyti.

Tyrimo tikslas: Išanalizuoti penkias žemiausias ir penkias aukščiausias Gyvenimo kokybės indeksą Lietuvoje turinčias savivaldybes, sudaryti laiko eilučių modelius bei atlikti 2022 metų GKI prognozes kiekvienai iš savivaldybių.

Tyrimui atlikti buvo išsikelti tokie uždaviniai:

1. Išanalizuoti mokslinę literatūrą, susijusią su Gyvenimo kokybės indeksu bei laiko eilučių modeliais.
2. Sudaryti keturis laiko eilučių modelius subindeksams, įvertinti visų modelių statistinį koeficientų reikšmingumą bei atlikti subindeksų prognozavimą.
3. Naudojant subindeksų prognozes, atlikti 2022 metų GKI prognozes penkioms žemiausia ir penkioms aukščiausia GKI 2022 metais turinčioms savivaldybėms bei palyginti jas su tikromis tų metų reikšmėmis.
4. Atlikus analizę, pateikti rekomendacijas savivaldybėms, kaip pagerinti GKI rodiklius, ypač toms, kuriose rodikliai buvo žemiausi.

Darbą sudaro 2 pagrindinės dalys – teorinė ir praktinė. Teorinėje dalyje apžvelgiami praktinėje dalyje analizuojami kintamieji bei tyrimui atlikti naudojami metodai: ARIMA modelis, GAM

modelis, LSTM modelis, BiLSTM modelis bei modelių tikslumą nusakantys rodikliai. Praktinėje dalyje apžvelgiami darbe naudojami duomenys, sudaromi modeliai, atliekamas duomenų prognozavimas, pateikiamos darbo išvados. Darbas buvo atliktas, naudojantis „RStudio“ bei „Python“ programomis.

2 Teorinė dalis

2.1 Kintamieji

Kintamieji buvo analizuojami 2013 – 2021 metų laikotarpiu ir prognozės daromos 2022 metams.

- Materialinių gyvenimo sąlygų subindeksas;
- Gyventojų verslumo ir verslo konkurencingumo subindeksas;
- Sveikatos paslaugų subindeksas;
- Švietimo paslaugų subindeksas;
- Demografijos, pilietinio ir visuomeninio aktyvumo subindeksas;
- Viešosios infrastruktūros, gyvenamosios aplinkos kokybės ir saugumo subindeksas;
- Gyvenimo kokybės indeksas.

2.2 Metodologija

2.2.1 Paneliniai duomenys

Tyrimui atlikti naudojami duomenys buvo transformuoti pagal poreikį ir analizė atliekama su paneline duomenų struktūra. Paneliniai duomenys (angl. panel data) – duomenų formatas, kai kintamieji yra stebimi tam tikrą laiką. Paneliniai duomenys gali būti 2 rūšių – subalansuoti ir nesubalansuoti. Subalansuoti duomenys yra tokie, kai kiekvienas kintamasis yra stebimas visais laiko momentais. Nesubalansuoti duomenys yra tokie, kai bent vienas kintamasis stebimas ne visais laiko momentais. Šiame darbe duomenys buvo subalansuoti.

2.2.2 Testai

- Pagerintasis (angl. augmented) Dickey–Fuller testas yra naudojamas duomenų stacionarumui tikrinti.
- Ljung–Box testas yra naudojamas modelio liekanų autokoreliacijai tikrinti.
- Durbin–Watson testas taip pat yra naudojamas modelio liekanų autokoreliacijai tikrinti.
- Jarque–Bera testas yra naudojamas modelio liekanų normalumui tikrinti.
- Breusch–Pagan testas yra naudojamas tikrinti modelio prielaidą apie liekanų heteroskedastiškumą.
- Shapiro–Wilk testas yra naudojamas modelio liekanų normalumui tikrinti.
- ARCH LM testas yra naudojamas nustatyti, ar modelio liekanų dispersija priklauso nuo ankstesnių laikotarpių arba ar yra heteroskedastiškumo efektas.

2.2.3 ARIMA modelis

Autoregresinis integruotas slenkančio vidurkio modelis (ARIMA) yra bendras autoregresinio slenkančio vidurkio modelio (ARMA) apibendrinimas, kuris sujungia autoregresinę (AR) procesą ir slenkančio vidurkio (MA) procesus ir sudaro laiko eilučių jungtinį modelį. ARIMA(p, d, q) apimantys pagrindiniai modelio elementai yra tokie:

- **AR**: Autoregresija. Regresijos modelis, kuris naudoja priklausomybes tarp stebėjimų ir keleto paslinktų stebėjimų (p).
- **I**: Integravimas. Norint laiko eilutę padaryti stacionaria, naudojamas stebėjimų skirtumas įvairiais laiko momentais (d).
- **MA**: Slenkantis vidurkis. Jis atsižvelgia į priklausomybę tarp stebėjimų ir liekamųjų klaidų terminų, kai slenkančio vidurkio modelis naudojamas lag’uotiems stebėjimams (q).

Autoregresinio modelio, kurio eilė yra (p), t.y. AR(p), formulė gali būti užrašyta kaip tiesinė lygtis:

$$x_t = c + \sum_{i=1}^p \phi_i x_{t-i} + \epsilon_t, \quad t \in \mathbb{Z}, \quad (2.1)$$

čia $\{x_t, t \in \mathbb{Z}\}$ yra stacionarus procesas, c yra konstanta, ϕ_i yra autokoreliacijos koeficientai esant poslinkiams $1, 2, \dots, p$ ir $\{\epsilon_t, t \in \mathbb{Z}\}$ yra Gauso baltasis triukšmas su nuliniu vidurkiu ir dispersija σ_ϵ^2 [3].

Atsitiktinis procesas $X = (X_t, t \in T)$ vadinamas **stipriai stacionariu** (stacionariu siaurąja prasme), jei atsitiktinių vektorių

$$(X_{t_1}, X_{t_2}, \dots, X_{t_m}) \quad (2.2)$$

ir

$$(X_{t_1+h}, X_{t_2+h}, \dots, X_{t_m+h}) \quad (2.3)$$

skirstiniai sutampa su visais $t_1 < t_2 < \dots < t_m \in T$ ir tokiu $h > 0$, kad $t_1 + h, \dots, t_m + h \in T$ [7].

Gauso baltasis triukšmas

Procesas $X = (X_t, t \in \mathbb{Z})$, kuomet $X_t, t \in \mathbb{Z}$ yra nepriklausomi atsitiktiniai dydžiai su vienuodu pasiskirstymu $\mathcal{N}(0, \sigma^2)$, yra vadinamas Gauso procesu su vidurkio funkcija

$$m_X(t) = 0, \quad t \in \mathbb{Z}, \quad (2.4)$$

ir kovariacine funkcija

$$\Gamma_X(s, t) = \begin{cases} \sigma^2, & \text{kai } s = t, \\ 0, & \text{kai } s \neq t, \end{cases} \quad s, t \in \mathbb{Z}. \quad (2.5)$$

Taip apibrėžtas procesas vadinamas **Gauso baltuoju triukšmu** [7].

Tiesinės regresijos lygtis

$$Y_t = \beta_0 + \beta_1 \cdot t + \epsilon_t, \quad (2.6)$$

čia

- Y_t – priklausomojo kintamojo reikšmė tam tikru laiko momentu t ,
- β_0 – pradinė reikšmė, kai $t = 0$,
- β_1 – trendo koeficientas,
- t – laiko momentas (metai),
- ϵ_t – paklaida, apibūdinanti atsitiktinius triukšmus.

ARIMA modelio parametrų formulės

- **Standartinė paklaida:**

$$\text{Standartinė paklaida} = \sqrt{\sigma^2 \cdot \text{Var}(x)}, \quad (2.7)$$

čia

- σ^2 – liekanų empirinė dispersija,
- $\text{Var}(x)$ – dispersija.

- **Liekanų empirinė dispersija (σ^2):**

$$\sigma^2 = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n - p}, \quad (2.8)$$

čia

- y_i – stebėtos reikšmės,
- $\hat{y}_i$ – numatytos modelio reikšmės,
- n – stebėjimų skaičius,
- p – modelio parametrų skaičius.

- **Akaike informacinis kriterijus (AIC):**

$$\text{AIC} = 2k - 2\log(L), \quad (2.9)$$

čia

- k – modelio parametrų skaičius,
- L – Logaritminė tikėtinumo funkcija (Log Likelihood).

- **Bajeso informacinis kriterijus (BIC):**

$$\text{BIC} = \log(n)k - 2\log(L), \quad (2.10)$$

čia

- $\log(n)$ – stebėjimų skaičiaus logaritmas,
- k – modelio parametrų skaičius,
- L – Logaritminė tikėtinumo funkcija (Log Likelihood).

- **Logaritminė tikėtinumo funkcija (Log Likelihood):**

$$\log L(\theta) = \sum_{i=1}^n \log f(y_i|\theta), \quad (2.11)$$

čia

- $L(\theta)$ – tikėtinumo funkcija, apskaičiuota pagal modelio parametrus θ ,
- y_i – stebėtos reikšmės,
- $f(y_i|\theta)$ – tikimybės tankio funkcija stebėjimui y_i su parametrais θ .

- **95% Pasiklivimo intervalų formulė:**

$$\text{Prognozė} \pm 1,96 \cdot \text{Standartinė paklaida}, \quad (2.12)$$

arba

$$\hat{B}_t \pm 1,96 \cdot \sigma, \quad (2.13)$$

čia

- $\hat{B}_t$ – prognozuota reikšmė,
- σ – liekanų empirinis standartinis nuokrypis.

2.2.4 GAM modelis

Apibendrintasis adityvusis modelis (GAM) yra išplėstinė daugybinės tiesinės regresijos modelio versija, kurią galima išreikšti taip:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon, \quad (2.14)$$

Norint įvertinti netiesinius efektus, GAM modelis kiekvieną tiesinio sąryšio komponentą $\beta_j x_j$ pakeičia glodžia netiesine funkcija $f_j(x_j)$.

Taigi, bendroji GAM formulė yra:

$$y = \beta_0 + f_1(x_1) + f_2(x_2) + \cdots + f_p(x_p) + \epsilon, \quad (2.15)$$

Šis modelis vadinamas **adityviuoju modeliu**, nes kiekvieną $f_j(x_j)$ (čia $j = 1, \dots, p$) komponentę apskaičiuojame atskirai ir tuomet jas susumuojame [1].

GAM modelio parametrų formulės

- **Standartinė paklaida:**

$$SE(\hat{y}) = \sqrt{\text{Var}(\hat{f}(x)) + \sigma^2}, \quad (2.16)$$

čia

- $\text{Var}(\hat{f}(x))$ – dispersija,
- σ^2 – liekanų empirinė dispersija.

- **Koreguotas determinacijos koeficientas (Adjusted R-squared) (R_{adj}^2):**

$$R_{adj}^2 = 1 - \left(\frac{(1 - R^2)(n - 1)}{n - p - 1} \right), \quad (2.17)$$

čia

- R^2 – determinacijos koeficientas (paprastas R-squared),
- n – stebėjimų skaičius,
- p – modelio kintamųjų skaičius.

- **Nuokrypis (Deviance):**

$$\text{Nuokrypis} = 2 \sum_{i=1}^n \left(y_i \log \frac{y_i}{\hat{y}_i} - (y_i - \hat{y}_i) \right), \quad (2.18)$$

čia

- y_i – stebėtos reikšmės,
- $\hat{y}_i$ – modelio numatytos reikšmės.

- **Kryžminės validacijos įvertis (Generalized Cross-Validation, GCV):**

$$\text{GCV} = \frac{1}{n} \sum_{i=1}^n \left(\frac{y_i - \hat{y}_i}{1 - \frac{v_i}{n}} \right)^2, \quad (2.19)$$

čia

- y_i – stebėtos reikšmės,
- $\hat{y}_i$ – modelio numatytos reikšmės,
- v_i – stebinio įtakos koeficiento (leverage) reikšmė.

- **Mastelio įvertis (Scale Estimate):**

$$\text{Mastelio įvertis} = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n - p}}, \quad (2.20)$$

čia:

- y_i – stebėtos reikšmės,
- $\hat{y}_i$ – modelio numatytos reikšmės,
- n – stebėjimų skaičius,
- p – modelio parametrų skaičius.

2.2.5 LSTM modelis

Ilgalaikės trumpalaikės atminties (LSTM) modelis yra vienas iš rekurentinių neuro-ninių tinklų (RNN) tipų modelių. Šis modelis sugeba mokytis ilgalaikių priklausomybių ir yra labai populiarus dirbant su laiko eilučių duomenimis.

LSTM modelis turi trejus vartus, kurie valdo pagrindinį informacijos srautą: Įėjimo vartai, Pamiršimo vartai ir Išėjimo vartai.

Vartai aprašomi šiomis lygtimis [4]:

- Įėjimo vartai: $i_t = \sigma(W_i h_{t-1} + W_i h_t)$,
- Pamiršimo vartai: $f_t = \sigma(W_f h_{t-1} + W_f h_t)$,
- Išėjimo vartai: $o_t = \sigma(W_o h_{t-1} + W_o h_t)$,
- Tarpinė ląstelės būseną: $\tilde{C}_t = \tanh(W_c h_{t-1} + W_c h_t)$,
- Ląstelės būseną (kitas atminties įėjimas): $c_t = (i_t \cdot \tilde{C}_t) + (f_t \cdot c_{t-1})$,
- Nauja būseną: $h_t = o_t \cdot \tanh(c_t)$,

čia

- X_t – įėjimo vektorius,
- h_t – išėjimo vektorius,
- W, U, f – parametrų matricos ir vektorius,
- $\cdot$ žymi vektorių skaliarinę daugybą.

Monte Karlo pasiklovimo intervalų formulė

$$\text{Pasiklovimo intervalas} = \text{vidurkis} \pm 1,96 \cdot s, \quad (2.21)$$

čia

- vidurkis – tai visų Monte Karlo prognozių vidutinė reikšmė,
- s – standartinis nuokrypis.

Standartinis nuokrypis

$$s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n}}, \quad (2.22)$$

čia

- x_i – duomenų reikšmė,
- $\bar{x}$ – duomenų vidurkis,
- n – duomenų imties dydis.

Baltasis triukšmas

Baltuoju triukšmu vadinsime nekoreliuotų atsitiktinių dydžių su vidurkiu 0 ir dispersija σ^2 seką $\{X_t\}$, $t \in \mathbb{N}$ [8].

2.2.6 BiLSTM modelis

Dvikryptis ilgalaikės trumpalaikės atminties (BiLSTM) modelis yra patobulinta LSTM modelio algoritmo versija. Dvikryptis LSTM (BiLSTM) algoritmas apjungia pageidaujamas dvikrypčio rekurentinio neuroninio tinklo (RNN) ir LSTM savybes.

BiLSTM sujungia dvi paslėptąsias būsenas, kurios leidžia informacijai kilti tiek iš atgalinio sluoksnio, tiek iš į priekį einančio sluoksnio.

Čia į priekį einančios, atgalinės ir išėjimo sekos pateikiamos taip:

- Įėjimo seka: $\vec{H}_t = \mathcal{L}(X_t W_{x\vec{H}} + \vec{H}_{t-1} W_{\vec{H}\vec{H}} + b_{\vec{H}}),$
- Grįžimo seka: $\overleftarrow{H}_t = \mathcal{L}(X_t W_{x\overleftarrow{H}} + \overleftarrow{H}_{t-1} W_{\overleftarrow{H}\overleftarrow{H}} + b_{\overleftarrow{H}}),$
- Išvestis: $Y_t = \vec{H}_t W_{\vec{H}Y} + \overleftarrow{H}_t W_{\overleftarrow{H}Y} + b_Y,$

čia

$\vec{H}_t$: rodyklė virš H žymi priekine kryptimi apdorojamą informaciją,

tai yra paslėptojo sluoksnio būsena t -uoju laiko momentu, apskaičiuota iš praeities į ateitį.

$\overleftarrow{H}_t$: rodyklė virš H žymi atgaline kryptimi apdorojamą informaciją,

tai yra paslėptojo sluoksnio būsena t -uoju laiko momentu, apskaičiuota iš ateities į praeitį.

Formulėje $\mathcal{L}$ yra sigmoidinė funkcija, kuri laikoma LSTM dalimi BiLSTM struktūroje. Sigmoidinė funkcija matematiškai apibūdinama taip:

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \quad (2.23)$$

ši funkcija suskirsto bet kurį realųjį skaičių į intervalą nuo 0 iki 1.

BiLSTM yra plačiai taikomas klasifikacijoje, tekstų atpažinime, kalbos atpažinime [5].

2.2.7 Modelio tikslumą nusakančios metrikos

- Šaknis iš vidutinės kvadratinės paklaidos – RMSE (angl. Root Mean Square Error)

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}}, \quad (2.24)$$

čia

- y_i – faktinė reikšmė,
- $\hat{y}_i$ – modelio įvertinta reikšmė,
- n – stebėjimų skaičius.

- Vidutinis kvadratinis nuokrypis – MSE (angl. Mean Squared Error)

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (2.25)$$

čia

- y_i – faktinė reikšmė,
- $\hat{y}_i$ – modelio įvertinta reikšmė,
- n – stebėjimų skaičius.

- Vidutinė absoliutinė procentinė paklaida – MAPE (angl. Mean Absolute Percentage Error)

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \cdot 100\%, \quad (2.26)$$

čia

- y_i – faktinė reikšmė,
- $\hat{y}_i$ – modelio įvertinta reikšmė,
- n – stebėjimų skaičius.

2.2.8 Gyvenimo kokybės indekso formulė

Pagrindiniai Gyvenimo kokybės indeksą sudarantys subindeksai yra 6 (skliausteliuose nurodytas kiekvieno subindekso svoris):

- Materialinės gyvenimo sąlygos – A (30%);
- Gyventojų verslumas ir verslo konkurencingumas – B (20%);
- Sveikatos paslaugos – C (10%);
- Švietimo paslaugos – D (10%);
- Demografija, pilietinis ir visuomeninis aktyvumas – E (15%);
- Viešoji infrastruktūra, gyvenamosios aplinkos kokybė ir saugumas – F (15%).

$$GKI = A \cdot 0.3 + B \cdot 0.2 + C \cdot 0.1 + D \cdot 0.1 + E \cdot 0.15 + F \cdot 0.15, \quad (2.27)$$

čia: A – Materialinės gyvenimo sąlygos; B – Gyventojų verslumas ir verslo konkurencingumas; C – Sveikatos paslaugos; D – Švietimo paslaugos; E – Demografija, pilietinis ir visuomeninis aktyvumas; F – Viešoji infrastruktūra, gyvenamosios aplinkos kokybė ir saugumas [2].

2.2.9 Monte Karlo simuliacijos

Monte Karlo simuliacijos leidžia tiksliai modeliuoti neapibrėžtumus ir generuoti tikimybinį rezultatų pasiskirstymą, arba kitaip tariant – apskaičiuoti pasiklivimo intervalų ribas. Monte Karlo simuliacijos buvo naudojamos prognozuoto Gyvenimo kokybės indekso reikšmių 95% pasikliautiniams intervalams gauti. Kuomet buvo gautos subindeksų prognozės bei jų 95% pasikliautiniai intervalai, buvo apskaičiuotas GKI, pagal prieš tai aprašytą (2.27) formulę. Norint patikrinti šių reikšmių patikimumą, buvo pasitelktos Monte Karlo simuliacijos ir šio metodo eiga atrodė taip :

1. **Parenkamas simuliacijų skaičius.** Kiekvienai iš savivaldybių buvo atliekama m ($m = 100$) simuliacijų.
2. **Subindeksų reikšmių generavimas.** Tuomet visose simuliacijose subindeksų reikšmės $S_i^{(j)}$ buvo generuojamos nepriklausomai nuo kitų subindeksų reikšmių pagal tolygųjį skirstinį (angl. Uniform distribution) tarp tam tikro subindekso ir savivaldybės prognozių pasiklivimo intervalų ribų $[PI_{apatinis}^i, PI_{viršutinis}^i]$, apskaičiuotų remiantis kiekvienu prognozės modeliu. Analizėje buvo nuspręsta pasiklivimo intervalų ribas imti 95%, kadangi šis pasiklivimo intervalų lygmuo yra dažniausiai taikomas moksliniuose tyrimuose. Nors tai yra įprastai praktikoje taikomi standartiniai pasiklivimo intervalai, tačiau likusių 5% ekstremumų pašalinimas nebuvo atliktas. Gautiems rezultatams tai galėjo turėti įtakos, tačiau ši analizė nebuvo atlikta, siekiant išvengti jau gautų rezultatų keitimo ir tai yra rimtesnės teorinės analizės reikalaujantis uždavinys. Tolygusis skirstinys duomenims buvo parinktas dėl pakankamai trumpos laiko eilutės ir nedidelio duomenų kiekio, kadangi nebuvo galimybės iki galo įvertinti subindeksų reikšmių pasiskirstymo. Buvo daroma prielaida, jog visos reikšmės pasiklivimo intervalų ribose yra vienodo tikėtumo:

$$S_i^{(j)} \sim \text{Uniform}(PI_{apatinis}^i, PI_{viršutinis}^i), \quad (2.28)$$

čia

- $S_i^{(j)}$ – j -osios simuliacijos i -ojo subindekso reikšmė,
 - $[PI_{apatinis}^i, PI_{viršutinis}^i]$ – prognozuotų reikšmių pasiklivimo intervalai.
3. **GKI apskaičiavimas kiekvienoje simuliacijoje.** Kiekvienoje simuliacijoje buvo apskaičiuojama GKI reikšmė $GKI^{(j)}$ pagal bendrą GKI formulę (2.27).
 4. **GKI pasiklivimo intervalai.** Po 100 simuliacijų iš visų $GKI^{(j)}$ reikšmių buvo apskaičiuotas jo vidurkis ir 95% pasiklivimo intervalai:
 - 95% pasiklivimo intervalai, naudojant 2,5% ir 97,5% kvantilius:

$$PI_{PI_{apatinis}^i} = \text{Quantile}_{0.025}(GKI^{(j)}), \quad PI_{PI_{viršutinis}^i} = \text{Quantile}_{0.975}(GKI^{(j)}). \quad (2.29)$$

3 Praktinė dalis

3.1 Duomenų aprašymas

Tyrimui atlikti buvo naudojami 4 duomenų rinkiniai iš Lietuvos Respublikos finansų ministerijos prižiūrimos svetainės www.lietuvosfinansai.lt. Pirmi 2 rinkiniai buvo apie 10-ies Lietuvos savivaldybių Gyvenimo kokybės indeksą bei jį sudarančius 6 subindeksus 2013 – 2021 metų laikotarpiu. Pirmasis duomenų rinkinys buvo sudarytas iš 5 savivaldybių (rikiuojant nuo žemiausios reikšmės): Skuodo rajono, Kalvarijos, Vilkaviškio rajono, Lazdijų rajono ir Kelmės rajono, kuriame buvo nurodytos 6 subindeksų bei GKI reikšmės 2013 – 2021 metų laikotarpiu kiekvienai iš savivaldybių. Antrasis duomenų rinkinys buvo sudarytas iš kitų 5 savivaldybių (rikiuojant nuo didžiausios reikšmės): Vilniaus miesto, Kauno miesto, Klaipėdos miesto, Kauno rajono bei Birštono, ir čia taip pat buvo nurodytos 6 subindeksų bei GKI reikšmės 2013 – 2021 metų laikotarpiu visoms savivaldybėms. Šie duomenys buvo paimti 2024 metų rugsėjo mėnesio 24 dieną.

Kiti 2 rinkiniai apie 2022 metų GKI ir subindeksų duomenis 10-čiai savivaldybių buvo paimti tą pačią 2024 metų rugsėjo 24 dieną. Šie duomenys buvo naudojami prognozių palyginimams su tikromis duomenų reikšmėmis. Šių duomenų struktūra buvo identiška pirmų 2 duomenų rinkinių struktūrai.

Darbe analizuojamos 10 savivaldybių buvo parinktos pagal naujausius 2022 metų GKI bei subindeksų duomenis. Penkios savivaldybės, turinčios žemiausią GKI ir penkios kitos savivaldybės, turinčios aukščiausią GKI 2022 metais.

Duomenų rinkiniuose pagal transformuotą struktūrą rodikliai buvo išsidėstę taip:

- Materialinės gyvenimo sąlygos;
- Gyventojų verslumas ir verslo konkurencingumas;
- Sveikatos paslaugos;
- Švietimo paslaugos;
- Demografija, pilietinis ir visuomeninis aktyvumas;
- Viešoji infrastruktūra, gyvenamosios aplinkos kokybė ir saugumas;
- Gyvenimo kokybės indeksas.

Kiekvienas subindeksas dar turi keletą savo sudedamųjų komponentių, tačiau darbe jos neanalizuojamos (kiekvieno subindekso komponentė turi savo svorį, o visų tam tikro subindekso komponentių svorių suma yra lygi to subindekso svoriui):

- Materialinės gyvenimo sąlygos – A (30%);
 - * Vidutinis mėnesinis neto darbo užmokestis – A1 (30%);
 - * Užimtų gyventojų dalis tarp darbingo amžiaus gyventojų – A2 (20%);
 - * Registruotų bedarbių ir darbingo amžiaus gyventojų santykis – A3 (10%);
 - * Gyvenamasis fondas: naudingas plotas, tenkantis vienam gyventojui – A4 (10%);
 - * Asmenų (šeimų), buvusių sąrašuose socialiniam būstui nuomoti, skaičius, tenkantis 1 tūkst. gyventojų metų pabaigoje – A5 (15%);
 - * Renovuotų daugiabučių dalis – A6 (15%).
- Gyventojų verslumas ir verslo konkurencingumas – B (20%);

- * Veikiančių ūkio subjektų skaičius, tenkantis 1 tūkst. gyventojų – B7 (15%);
- * Materialinės investicijos, tenkančios 1 tūkst. gyventojų – B8 (20%);
- * Tiesioginės užsienio investicijos, tenkančios 1 gyventojui metų pabaigoje – B9 (25%);
- * Įmonių apyvarta, tenkanti 1 tūkst. gyventojų – B10 (10%);
- * Pridėtinė vertė gamybos sąnaudomis, tenkanti 1 tūkst. gyventojų – B11 (30%).
- Sveikatos paslaugos – C (10%);
 - * Praktikuojančių sveikatos priežiūros specialistų skaičius, tenkantis 10 tūkst. gyventojų – C12 (20%);
 - * Paliatyvios pagalbos, globos, slaugos ir palaikomojo gydymo lovų skaičius, tenkantis 1 tūkst. gyventojų – C13 (15%);
 - * Sporto, varžybų ir sveikatingumo renginių dalyvių skaičius, tenkantis 1 tūkst. gyventojų – C14 (15%);
 - * Lovų ligoninėse (be slaugos lovų) skaičius, tenkantis 10 tūkst. gyventojų – C15 (15%);
 - * Stacionaro ligonių skaičius, tenkantis 1 tūkst. gyventojų – C16 (15%);
 - * Mirtingumo nuo neužkrečiamų ligų skaičius 100 tūkst. gyventojų – C17 (20%).
- Švietimo paslaugos – D (10%);
 - * Vaikų, dalyvaujančių ikimokykliniame ugdyme, dalis tarp 1–6 metų amžiaus gyventojų – D18 (20%);
 - * Dėl socialinių, psichologinių ir kitų priežasčių nesimokantys mokyklinio amžiaus vaikai, tenkantys 1 tūkst. nuolatinio mokyklinio amžiaus gyventojų – D19 (10%);
 - * Profesinių mokymo įstaigų mokinių skaičius, tenkantis 1 tūkst. gyventojų – D20 (10%);
 - * Lietuvių kalbos brandos egzaminą laikiusių absolventų, kurių egzamino rezultatai įvertinti 86–100 taškų, dalis – D21 (10%);
 - * Matematikos brandos egzaminą laikiusių absolventų, kurių egzamino rezultatai įvertinti 86–100 taškų, dalis – D22 (10%);
 - * Aukštos kvalifikacijos mokytojų dalis – D23 (10%);
 - * Universitetų ir kolegijų studentų skaičius, tenkantis 1 tūkst. gyventojų – D24 (20%);
 - * Geltonais autobusiukais į mokyklą vežiojamų mokinių dalis savivaldybėje – D25 (10%).
- Demografija, pilietinis ir visuomeninis aktyvumas – E (15%);
 - * Savivaldybių tarybų rinkimuose dalyvavusių rinkėjų skaičius, palyginti su visų rinkėjų skaičiumi – E26 (25%);
 - * Išlaikomo amžiaus žmonių koeficientas metų pabaigoje, tenkantis 100 gyventojų – E27 (20%);
 - * Bendrasis natūralios gyventojų kaitos rodiklis – E28 (25%);
 - * Atvykusių ir išvykusių asmenų skaičius (neto migracija), tenkantis 1 tūkst. gyventojų – E29 (15%);
 - * Kultūros centrų dalyvių skaičius, tenkantis 1 tūkst. gyventojų – E30 (15%).
- Viešojo infrastruktūra, gyvenamosios aplinkos kokybė ir saugumas – F (15%).

- * Teršalų kiekis, išmestas į aplinkos orą iš stacionarių taršos šaltinių, tenkančių 1 tūkst. gyventojų – F32 (20%);
- * Užregistruotų nusikaltimų, baudžiamųjų nusižengimų skaičius, tenkantis 100 tūkst. gyventojų – F33 (15%);
- * Kelių eismo įvykių skaičius, tenkantis 1 tūkst. gyventojų – F34 (10%);
- * Automobilių kelių su patobulinta danga dalis bendrame kelių tinkle – F35 (15%);
- * Dviračių takų ilgis metų pabaigoje, tenkantis 1 tūkst. gyventojų – F36 (10%);
- * Keleivių apyvarta kelių transportu, tenkanti 1 tūkst. gyventojų – F37 (10%);
- * Visų tipų apgyvendinimo įstaigose suteiktų nakvynių skaičius, tenkantis 1 tūkst. gyventojų – F38 (5%);
- * Muziejų skaičius, tenkantis 1 tūkst. gyventojų – F39 (5%);
- * Savivaldybių viešosios bibliotekos, tenkančios 1 tūkst. gyventojų – F40 (5%);
- * Kultūros centrų skaičius, tenkantis 1 tūkst. gyventojų – F41 (5%).

3.2 Laiko eilučių modelių pritaikymas

3.2.1 ARIMA modelio pritaikymas

Pirmasis modelis, kuris buvo taikytas savivaldybių laiko eilučių duomenims buvo ARIMA (Autoregressive Integrated Moving Average) modelis. Šis modelis, kaip ir visi kiti, buvo pasirinktas pagal aktualią literatūrą [3, 4, 6]. Taip pat modelis buvo parinktas, atsižvelgiant į jo pritaikomumą turimiems duomenims. Modeliui sudaryti buvo naudojami 2013–2021 metų subindeksų duomenys 10-čiai savivaldybių.

Pirma buvo atliekama analizė su 5 savivaldybėmis pagal žemiausią GKI, po to su 5 savivaldybėmis pagal aukščiausią GKI.

Duomenys buvo papildomai transformuoti ir modelis buvo taikomas kiekvienam iš 6 subindeksų visoms savivaldybėms. Prieš pritaikant modelį, buvo tikrinamos tam tikros duomenų bei modelio prielaidos, atliekami testai. Iš transformuoto duomenų rinkinio buvo pašalintos trūkstamos reikšmės (jei tokių buvo) bei stulpeliai konvertuoti į skaitinį duomenų tipą. Taip pat kiekvienai savivaldybei buvo atlikta pirminių duomenų trendo analizė, pritaikant tiesinį regresijos modelį. Šis trendas buvo apibrėžtas kaip priklausomybė tarp metų ir subindekso reiškių. Tiesinės regresijos pritaikymas trendui padėjo nustatyti, ar reikšmės laikui bėgant tolygiai didėja, ar mažėja. Rezultatai parodė, kad visose savivaldybėse buvo teigiamas trendas ($p < 0.001$), rodantis nuoseklų reiškių augimą. Atliekant tolimesnę analizę, trendas buvo pašalintas iš pradinių duomenų, siekiant atskirti stacionarią dalį. Tiesa, pirminė duomenų analizė, kuri buvo atlikta su ARIMA modeliu, buvo be trendo ir visi šio modelio bei prognozių rezultatai atspindi ARIMA reikšmes be trendo pirminiems duomenims. Analizės nuspręsta nepildyti su trendo rezultatais, kadangi jie skyrėsi tik šimtųjų tikslumu ir visi ARIMA modelio ir jo prognozių rezultatai tekste bei lentelėse yra be trendo reiškių pradiniais duomenims. Tuomet kiekvienos savivaldybės subindekso reikšmės buvo tikrinamos dėl stacionarumo, nes modelį galima taikyti tik stacionariams duomenims. Stacionarumui tikrinti buvo naudojamas ADF (Augmented Dickey–Fuller) testas. Jei testo p -reikšmė buvo didesnė už 0,05, reiškė, jog duomenys nestacionarūs ir buvo taikomas laiko eilutės diferencijavimas bei imti kintamųjų skirtumai laike. Diferencijavimas - tai metodas, taikomas laiko eilučių duomenims, norint pasiekti duomenų stacionarumą. Jis atliekamas atimant ankstesnę duomenų reikšmę iš dabartinės duomenų reikšmės, taip gaunant duomenų reiškių skirtumus laike. Diferencijuotus duomenis, atlikus analizę, reikia grąžinti į pradinę skalę, tam, kad prognozės būtų interpretuojamos lengviau ir būtų tinkamos praktiniam naudojimui. Diferencijavimą galima taikyti kelis kartus, tai buvo atlikta ir šiame darbe su tam tikrais duomenų rinkiniais. Jei pritaikius 1 lygio diferencijavimą nestacionarumo problema nedingdavo, būdavo taikytas 2 lygio diferencijavimas, tam, kad laiko eilutė taptų stacionari ir būtų galima taikyti ARIMA modelį. Pritaikius 2 lygio diferencijavimą, duomenys visada būdavo stacionarūs ir tuomet buvo atliekama tolimesnė analizė. Sudarant modelį buvo naudojama *auto.arima()* funkcija, kuri automatiškai parenka labiausiai tinkančius ARIMA(p, d, q) koeficientus, tad šių koeficientų nereikėjo rinkti rankiniu būdu. Visiems subindeksams buvo pritaikytas ARIMA(0, 0, 0) modelis ir pagal visas savybes tai buvo baltasis triukšmas, todėl modelio parametrų, tokių kaip AR, MA ar vidurkio koeficientų su standartinėmis paklaidomis nebuvo. To pasekoje nebuvo įmanoma gauti ir paties ARIMA modelio pasiklovimo intervalų, kadangi ARIMA(0, 0, 0) modelis turėjo nulinį vidurkį, bet buvo apskaičiuoti prognozių pasiklovimo intervalai. Prognozių 95% pasiklovimo intervalai yra pagrįsti modelio liekanų

dispersija bei standartine paklaida ir leidžia įvertinti prognozių tikslumą. Pritaikius modelį duomenims, buvo tikrinamos modelio liekanos bei papildomai brėžiami ACF bei PACF (atitinkamai autokoreliacijos bei dalinės autokoreliacijos) grafikai. Modelio liekanų autokoreliacijai tikrinti buvo taikomas Ljung–Box testas bei liekanų normalumui tikrinti buvo naudojamas Jarque–Bera testas. Pastarasis Jarque–Bera testas rodė, jog liekanų normalumas yra tinkamas modeliui, tačiau Ljung–Box testas buvo ne itin naudingas. Šis testas negalėjo niekur nustatyti autokoreliacijos, taip pat liekanų autokoreliacijos (ACF) grafikai parodė, jog kvadratinių ir teigiamų liekanų autokovariacijos nėra reikšmingos ir yra beveik lygios nuliui. Tai galėjo nutikti dėl per mažo arba nulinio liekanų kiekio, arba dėl labai mažos modelio dispersijos. ARIMA modelio liekanoms buvo atliekamas ARCH LM testas, kuris parodė, jog liekanos buvo stabilios ir neturėjo didelių dispersijos svyravimų. Liekanų heteroskedastiškumas buvo labai silpnas arba jo išvis nebuvo, dėl šios priežasties GARCH efektai nebuvo nustatyti. Taip pat subindeksų reikšmės buvo tik 9 metams, o tokia trumpa laiko eilutė nėra pritaikyta GARCH efektams analizuoti. Kadangi darbe naudojami laiko eilučių duomenys buvo metiniai, sezoninių svyravimų juose nebuvo, nes laiko intervalai yra pakankamai dideli. Taip pat autokoreliacijos funkcijos grafikai (ACF) neparodė periodų svyravimų, todėl galima buvo daryti duomenų prielaidą, jog sezoniškumas juose neegzistuoja. Subindeksų reikšmių ARIMA modelio bei prognozių rezultatai buvo gauti 2 būdais - be trendo priminiams duomenims ir su trendu ir stacionaria dalimi, tačiau, kaip ir buvo minėta anksčiau, nuspręsta rezultatų su trendu nedėti į šį darbą, kadangi jie skyrėsi tik šimtųjų tikslumu ir nebuvo labai informatyvūs. Šis ARIMA modelio algoritmas buvo taikomas visiems subindeksams 10-čiai savivaldybių.

Iš 1 lentelės matome ARIMA modelio 4 įverčių rezultatus visiems subindeksams. Pirmasis įvertis – liekanų dispersija, arba kitaip σ^2 . Kuo mažesnė σ^2 reikšmė, tuo geriau modelis paaiškina duomenis, nes liekanų dispersija yra maža. Matome, jog A subindekse mažiausią σ^2 reikšmę turėjo Kelmės r. sav. 0,001142, o didžiausią σ^2 reikšmę turėjo Lazdijų r. sav. 0,01226. Sekantys 2 įverčiai buvo Akaike informacinis kriterijus (AIC) ir Bajeso informacinis kriterijus (BIC). Abu įverčiai padeda įvertinti modelio sudėtingumą. Kuo mažesnės šių įverčių reikšmės, tuo labiau duomenims tinka modelis. Mažiausias šių įverčių reikšmes turėjo Kelmės r. sav., atitinkamai –28,56 ir –28,41, o didžiausias – Lazdijų r. sav. –8,94 ir –9. Ketvirtasis ARIMA modelio įvertis buvo Logaritminė tikėtinumo funkcija (log-likelihood), kuri taip pat parodo modelio atitikimą turimiems duomenims. Šį įvertį reikėtų vertinti kartu su AIC ir BIC įverčiais, nes jie yra gan glaudžiai susiję. Kuo didesnė Logaritminė tikėtinumo funkcija, tuo geriau veikia modelis. Taip pat, čia matome, jog didžiausią logaritminę tikėtinumo funkcijos reikšmę turėjo Kelmės r. sav. 16,28, o mažiausią – Lazdijų r. sav. 5,47 ir šie rezultatai stipriai koreliuoja su AIC ir BIC įverčių rezultatais. Tad galime teigti, jog ARIMA modelis su A subindekso savivaldybių duomenimis geriausiai veikė su Kelmės r. sav., o prasčiausiai – su Lazdijų r. sav. Kalbant apie B subindekso modelio rezultatus matome, jog mažiausią σ^2 reikšmę turėjo Lazdijų r. sav. 0,0001144, o didžiausią – Kelmės r. sav. 0,0007404, AIC ir BIC mažiausią reikšmę turėjo Vilkaviškio r. sav., atitinkamai –46,34 ir –46,18. Taip pat ši savivaldybė turėjo didžiausią Logaritminės tikėtinumo funkcijos reikšmę 25,17, tad ARIMA modelis geriausiai suveikė su Vilkaviškio rajono savivaldybe. Tuo tarpu didžiausią AIC ir BIC reikšmę turėjo Kelmės r. sav., atitinkamai –28,59 ir –28,65 bei ši savivaldybė turėjo mažiausią Logaritminės tikėtinumo funkcijos reikšmę, t.y. 15,3. Šie įverčiai rodo, jog Kelmės rajono savivaldybė pagal B subindekso duomenis buvo sumodeliuota prasčiausiai. Iš C subindekso duomenų matome, jog geriausi modelio rezultatai buvo gauti su Lazdijų r. sav., šios savivaldybės AIC, BIC

Savivaldybė	σ^2	AIC	BIC	Log Likelihood
A Subindeksas				
Kalvarijos sav.	0,005734	−14,26	−14,32	8,13
Kelmės r. sav.	0,001142	−28,56	−28,41	16,28
Lazdijų r. sav.	0,01226	−8,94	−9	5,47
Skuodo r. sav.	0,001824	−22,28	−22,34	12,14
Vilkaviškio r. sav.	0,001231	−25,03	−25,09	13,52
B Subindeksas				
Kalvarijos sav.	0,0003893	−33,09	−33,15	17,55
Kelmės r. sav.	0,0007404	−28,59	−28,65	15,3
Lazdijų r. sav.	0,0001144	−41,66	−41,72	21,83
Skuodo r. sav.	0,0003901	−33,08	−33,13	17,54
Vilkaviškio r. sav.	0,0001238	−46,34	−46,18	25,17
C Subindeksas				
Kalvarijos sav.	0,002666	−19,62	−19,68	10,81
Kelmės r. sav.	0,004664	−15,71	−15,76	8,85
Lazdijų r. sav.	0,0005385	−30,82	−30,88	16,41
Skuodo r. sav.	0,002916	−19	−19,05	10,5
Vilkaviškio r. sav.	0,001137	−29,53	−29,45	15,76
D Subindeksas				
Kalvarijos sav.	0,01432	−7,86	−7,91	4,93
Kelmės r. sav.	0,01321	−8,42	−8,48	5,21
Lazdijų r. sav.	0,00702	−14,97	−14,89	8,48
Skuodo r. sav.	0,001233	−25,02	−25,08	13,51
Vilkaviškio r. sav.	0,001262	−28,7	−28,62	15,35
E Subindeksas				
Kalvarijos sav.	0,002103	−21,29	−21,34	11,64
Kelmės r. sav.	0,002761	−19,38	−19,43	10,69
Lazdijų r. sav.	0,00256	−23,04	−22,96	12,52
Skuodo r. sav.	0,005249	−14,88	−14,94	8,44
Vilkaviškio r. sav.	0,00175	−22,57	−22,63	12,29
F Subindeksas				
Kalvarijos sav.	0,001	−30,56	−30,48	16,28
Kelmės r. sav.	0,002117	−21,24	−21,29	11,62
Lazdijų r. sav.	0,0009287	−27,01	−27,06	14,5
Skuodo r. sav.	0,0001488	−39,82	−39,88	20,91
Vilkaviškio r. sav.	0,0007618	−28,39	−28,45	15,2

1 lentelė: ARIMA modelio rezultatai visiems subindeksams pagal mažiausią GKI turinčias savivaldybes

Čia:

σ^2 – Liekanų empirinė dispersija (2.8).

AIC – Akaike informacinis kriterijus (2.9). 22

BIC – Bajeso informacinis kriterijus (2.10).

Log Likelihood – Logaritminė tikėtinumo funkcija (2.11).

ir Logaritminės tikėtinumo funkcijos įverčiai buvo atitinkamai $-30,82$, $-30,88$ ir $16,41$. Tuo tarpu prasčiausias ARIMA modelis C subindeksui buvo Kelmės rajono savivaldybei. Šios savivaldybės ARIMA modelio AIC, BIC ir Logaritminės tikėtinumo funkcijos įverčiai buvo $-15,71$, $-15,76$ ir $8,85$. Vertinant tuos pačius ARIMA modelio įverčius tolimesniam D subindeksui, matome, jog geriausiai sumodeliuota buvo Vilkaviškio rajono savivaldybė, AIC buvo $-28,7$, BIC buvo $-28,62$ ir Logaritminės tikėtinumo funkcijos reikšmė buvo $15,35$. Prasčiausiai sumodeliuota buvo Kalvarijos savivaldybė – AIC buvo $-7,86$, BIC buvo $-7,91$ bei Logaritminės tikėtinumo funkcijos reikšmė buvo $4,93$. Žiūrint į E subindekso rezultatus, matome, jog geriausi modelio rezultatai atsispindėjo Lazdijų rajono savivaldybėje, o prasčiausi – Skuodo rajono savivaldybėje. F subindekso rezultatai leidžia daryti išvadą, jog geriausiai sumodeliuota buvo Skuodo rajono savivaldybė, o prasčiausiai – Kelmės rajono savivaldybė.

Žiūrint į 2 lentelę, matome ARIMA modelio įverčius likusiems subindeksams. Pagal A subindeksą aukščiausią σ^2 reikšmę $0,04938$ turėjo Birštono savivaldybė. Ši savivaldybė taip pat turėjo aukščiausias AIC $0,81$ bei BIC $0,75$ reikšmes ir žemiausią Logaritminės tikėtinumo funkcijos reikšmę $0,6$, tad Birštono savivaldybės rezultatai pagal A subindeksą buvo sumodeliuoti prasčiausiai. Geriausiai sumodeliuota buvo Kauno miesto savivaldybė, kadangi jos σ^2 , AIC ir BIC reikšmės buvo mažiausios, o Logaritminės tikėtinumo funkcijos reikšmė buvo didžiausia. Sekančio B subindekso rezultatai rodo, jog prasčiausiai sumodeliuota savivaldybė buvo Klaipėdos miesto, čia σ^2 reikšmė buvo $0,03824$, AIC buvo $-0,98$, BIC buvo $-1,04$, o Logaritminės tikėtinumo funkcijos reikšmė buvo $1,49$. Geriausiai sumodeliuota savivaldybė pagal B subindekso ARIMA modelio rezultatus buvo Kauno rajono savivaldybė. Kalbant apie C subindekso rezultatus, iš lentelės 2 matome, jog pagal įverčių AIC, BIC ir Logaritminės tikėtinumo funkcijos reikšmes prasčiausiai sumodeliuota buvo Birštono savivaldybė, jos įverčiai buvo atitinkamai $-3,43$, $-3,48$ ir $2,71$. O tuo tarpu geriausiai sumodeliuota buvo Klaipėdos miesto savivaldybė. Tiesa, Klaipėdos miesto savivaldybės įverčiai buvo labai panašūs ir į Vilniaus miesto savivaldybės įverčius, tad galime teigti, jog šių abiejų miestų savivaldybės buvo geriausiai sumodeliuotos pagal C subindeksą, geriausią GKI turinčioms savivaldybėms. Pagal D subindekso rezultatus matome, jog didžiausią σ^2 reikšmę turėjo Vilniaus miesto savivaldybė. Ši savivaldybė taip pat turėjo aukščiausias AIC $-2,72$ ir BIC $-2,77$ reikšmes bei žemiausią Logaritminės tikėtinumo funkcijos reikšmę $2,36$ ir tai rodo, jog ji sumodeliuota buvo prasčiausiai. Geriausiai pagal D subindeksą sumodeliuota buvo Kauno rajono savivaldybė. Šios savivaldybės σ^2 reikšmė buvo $0,003869$, AIC buvo $-17,02$, BIC buvo $-17,07$, o Logaritminės tikėtinumo funkcijos reikšmė buvo $9,51$. E subindekso prasčiausiai sumodeliuotas reikšmes turėjo Klaipėdos miesto savivaldybė, o geriausiai sumodeliuotas reikšmes turėjo Vilniaus miesto savivaldybė. Paskutiniojo F subindekso rezultatai rodo, jog prasčiausiai sumodeliuota buvo Birštono savivaldybė, o geriausiai – Vilniaus miesto savivaldybė.

3.2.2 GAM modelio pritaikymas

Sekantis modelis, kuris buvo taikytas savivaldybių duomenims, buvo GAM modelis. Šis modelis taip pat buvo pasirinktas pagal aktualią literatūrą [6] bei pagal modelio pritaikomumą duomenims.

Analizė buvo atliekama identiška tvarka – pirma su 5 savivaldybėmis pagal žemiausią GKI, po to su 5 savivaldybėmis pagal aukščiausią GKI. Modeliavimo procesui buvo naudojami 2013–2021 metų savivaldybių subindeksų duomenys.

Savivaldybė	σ^2	AIC	BIC	Log Likelihood
A Subindeksas				
Birštono sav.	0,04938	0,81	0,75	0,6
Kauno m. sav.	0,0007519	−31,91	−31,75	17,95
Kauno r. sav.	0,001318	−27,42	−27,26	15,71
Klaipėdos m. sav.	0,0009873	−29,73	−29,57	16,86
Vilniaus m. sav.	0,0008999	−30,47	−30,31	17,24
B Subindeksas				
Birštono sav.	0,006557	−13,32	−13,38	7,66
Kauno m. sav.	0,005333	−14,77	−14,83	8,39
Kauno r. sav.	0,0023	−20,66	−20,71	11,33
Klaipėdos m. sav.	0,03824	−0,98	−1,04	1,49
Vilniaus m. sav.	0,003708	−17,32	−17,37	9,66
C Subindeksas				
Birštono sav.	0,02696	−3,43	−3,48	2,71
Kauno m. sav.	0,003994	−16,8	−16,85	9,4
Kauno r. sav.	0,004222	−16,41	−16,46	9,2
Klaipėdos m. sav.	0,003493	−17,73	−17,79	9,87
Vilniaus m. sav.	0,003657	−17,41	−17,47	9,71
D Subindeksas				
Birštono sav.	0,01234	−8,9	−8,95	5,45
Kauno m. sav.	0,009564	−10,68	−10,74	6,34
Kauno r. sav.	0,003869	−17,02	−17,07	9,51
Klaipėdos m. sav.	0,005811	−14,17	−14,22	8,09
Vilniaus m. sav.	0,02984	−2,72	−2,77	2,36
E Subindeksas				
Birštono sav.	0,005431	−14,64	−14,7	8,32
Kauno m. sav.	0,004589	−15,82	−15,88	8,91
Kauno r. sav.	0,005186	−14,97	−15,02	8,48
Klaipėdos m. sav.	0,006423	−13,47	−13,52	7,73
Vilniaus m. sav.	0,001629	−23,07	−23,13	12,54
F Subindeksas				
Birštono sav.	0,004038	−16,72	−16,77	9,36
Kauno m. sav.	0,001514	−23,59	−23,64	12,79
Kauno r. sav.	0,0008266	−32,08	−32	17,04
Klaipėdos m. sav.	0,001684	−22,84	−22,89	12,42
Vilniaus m. sav.	0,000301	−40,16	−40,08	21,08

2 lentelė: ARIMA modelio rezultatai visiems subindeksams pagal didžiausią GKI turinčias savivaldybes

Čia:

σ^2 – Liekanų empirinė dispersija (2.8).

AIC – Akaike informacinis kriterijus (2.9). 24

BIC – Bajeso informacinis kriterijus (2.10).

Log Likelihood – Logaritminė tikėtinumo funkcija (2.11).

Prieš pritaikant GAM modelį, duomenys buvo transformuoti į tą patį formatą, kaip ir sudarinėjant ARIMA modelį. Taip pat buvo tikrinamos tam tikros duomenų bei modelio prielaidos ir testai. Prieš sudarinėjant apibendrintąjį adityvųjį modelį, buvo tikrinamas duomenų stacionarumas, naudojant ADF (Augmented Dickey–Fuller) testą. Jei duomenys buvo nestacionarūs, buvo taikomas diferencijavimas ir sudaryti skirtumai laike, kurie padėjo pasiekti stacionarumą. Šiame modelyje buvo tikrinamos koreliacijos tarp kintamųjų ir brėžiamas koreliacijų grafikas, siekiant patikrinti, ar nėra multikolinearumo problemos. Tuomet buvo sudarytas bei pritaikytas GAM modelis. Taip pat duomenyse buvo pastebėtas ilgalaikis trendo poveikis, todėl modelyje jis buvo įtrauktas ir žymimas $s(Year)$. Šis trendas buvo didėjantis ir atspindėjo ankstesnių metų duomenų reikšmių augimo tendenciją. Trendo įtraukimas į modelį pagerino tiek modeliavimo rezultatus, tiek prognozes. Tuomet buvo peržiūrėta GAM modelio diagnostika, naudojantis `gam.check(gam_model)` funkcija. Ši diagnostika tikrino liekanų pasiskirstymą, modelio dispersiją. Tai padeda lengviau įvertinti, ar modelis gerai pritaikytas duomenims, ir, ar ateities prognozės bus kiek įmanoma tikslesnės. Autokoreliacijai modelio liekanose nustatyti buvo naudojamas Durbin – Watson testas. Jei šio testo p -reikšmė buvo mažesnė nei 0,05, tai liekanose buvo statistiškai reikšminga autokoreliacija ir tuomet buvo taikomas mišrus modelis (GAMM) su AR(1) struktūra. Šis mišrus modelis padeda pašalinti autokoreliacijos problemą liekanose. Po šio mišraus modelio vėl buvo atliekama modelio diagnostika, patikrinamos liekanų charakteristikos. Liekanų dispersijai tikrinti buvo naudojamas Breusch–Pagan testas. Jei testo p -reikšmė buvo mažesnė už 0,05, tai rodė heteroskedastiškumo problemą, tačiau šiame modelyje to nebuvo ir toliau buvo tikrinamas liekanų normalumas. Liekanų normalumui tikrinti buvo naudojamas Shapiro–Wilk testas ir brėžiamas kvantilių palyginimo (Q–Q plot) grafikas. Kadangi šio testo reikšmė buvo didesnė nei 0,05 ir kvantilių palyginimo grafike taškai buvo pasiskirstę aplink tiesę, taigi modelio liekanos buvo normaliai pasiskirsčiusios, modelio prielaidos buvo tenkinamos ir prognozės turėtų būti tikslios. Liekanoms dar buvo tikrinti GARCH efektai ir atliekamas ARCH LM testas, kuris įvertina, ar likučių dispersija kinta laikui bėgant. Kadangi šio testo p -reikšmė buvo 0,5261, GARCH efektai buvo nenustatyti, o tai rodė, kad likučių dispersija buvo stabili.

Šis GAM modelio algoritmas buvo taikomas visiems subindeksams 10–čiai savivaldybių.

Subindeksas	R^2 (adj.)	Nuokrypis	GCV	Mastelio įvertis
A	–0,00587	12,3%	0,0025378	0,0021571
B	0,129	25,5%	0,00018648	0,00015547
C	0,0997	21,5%	0,0012756	0,0010843
D	–0,0147	14,5%	0,0049138	0,0040351
E	0,212	39,1%	0,0017793	0,0013411
F	–0,12	2,32%	0,00068147	0,00057925

3 lentelė: GAM modelio rezultatai subindeksams pagal mažiausią GKI turinčias savivaldybes

Čia:

R^2 (adj.) – Koreguotas determinacijos koeficientas (2.17).

Nuokrypis – Nuokrypis (2.18).

GCV – Kryžminės validacijos įvertis (2.19).

Mastelio įvertis – Mastelio įvertis (2.20).

Gam modelio įverčiai, pateikti 3 lentelėje vaizduoja 4 statistinius GAM modelio įverčius

visiems subindeksams. Pirmasis – Determinacijos koeficientas R^2 (adj.), kuris padeda tiksliau įvertinti modelio galingumą. Kadangi GAM modeliai naudoja ne tik linijinį suderinamumą, R^2 (adj.) reikšmę reikėtų interpretuoti atsargiai. Matome, jog mažiausią R^2 (adj.) reikšmę turėjo F subindeksas $-0,12$, tuo tarpu aukščiausią šio koeficiento reikšmę turėjo E subindeksas $0,212$. Antrasis įvertis – Nuokrypis, kuris parodo modelio pritaikomumą duomenims. Kuo mažesnis nuokrypis, tuo geriau modelis paaiškina duomenis, tad matome, jog mažiausią reikšmę turėjo F subindeksas $2,32\%$, o didžiausią šio įverčio reikšmę turėjo E subindeksas $39,1\%$. Trečiasis įvertis yra kryžminio validavimo įvertis (Generalized Cross-Validation), kuris naudojamas parenkant modelio hiperparametrus. Kuo mažesnė GCV reikšmė, tuo geriau. Mažiausią GCV reikšmę šioje lentelėje matome B subindeksui $0,00018648$, o didžiausią – D subindeksui $0,0049138$. Ir paskutinis – Mastelio įvertis, kuris parodo liekanų dispersiją. Kuo mažesnis mastelio įvertis, tuo geriau, nes tai rodo, jog modelis geriau atitinka duomenis ir liekanų dispersija atitinkamai irgi yra maža. Iš šios 3 lentelės matome, jog mažiausią mastelio įvertį turėjo B subindeksas $0,00015547$, o didžiausią – D subindeksas $0,0040351$.

Subindeksas	R^2 (adj.)	Nuokrypis	GCV	Mastelio įvertis
A	$-0,123$	2,07%	$0,0048907$	$0,0041571$
B	$-0,042$	$9,16\%$	$0,0052855$	$0,0044927$
C	$-0,0787$	$5,96\%$	$0,0037643$	$0,0031997$
D	0,395	$56,4\%$	$0,0034449$	$0,0024215$
E	$-0,0549$	$8,04\%$	$0,003048$	$0,0025908$
F	$-0,0213$	11%	0,0010097	0,00085826

4 lentelė: GAM modelio rezultatai subindeksams pagal didžiausią GKI turinčias savivaldybes

Čia:

R^2 (adj.) – Koreguotas determinacijos koeficientas (2.17).

Nuokrypis – Nuokrypis (2.18).

GCV – Kryžminės validacijos įvertis (2.19).

Mastelio įvertis – Mastelio įvertis (2.20).

Iš GAM modelio rezultatų subindeksams, pateiktų 4 lentelėje, matome identiškai suskirstytus statistinius GAM modelio įverčius. Mažiausią Determinacijos koeficientą R^2 (adj.) turėjo A subindeksas $-0,123$, o didžiausią – D subindeksas $0,395$. Nuokrypio įverčio mažiausią reikšmę turėjo A subindeksas $2,07\%$, o didžiausią – D subindeksas $56,4\%$. Trečiasis, kryžminio validavimo įvertis, (Generalized Cross-Validation), kuris naudojamas parenkant modelio hiperparametrus, mažiausias buvo F subindeksui $0,0010097$, o didžiausias – B subindeksui $0,0052855$. Mažiausią mastelio įvertį turėjo F subindeksas $0,00085826$, o didžiausią – B subindeksas $0,0044927$.

Savivaldybė	Įvertis	Paklaida	95% P.I.
A Subindeksas			
Kalvarijos sav.	0,0653	0,0164	[0,0331; 0,0974]
Kelmės r. sav.	-0,0040	0,0232	[-0,0495; 0,0415]
Lazdijų r. sav.	-0,0108	0,0232	[-0,0563; 0,0347]
Skuodo r. sav.	0,0015	0,0232	[-0,0440; 0,0470]
Vilkaviškio r. sav.	0,0011	0,0232	[-0,0445; 0,0466]
B Subindeksas			
Kalvarijos sav.	0,0150	0,0044	[0,0064; 0,0237]
Kelmės r. sav.	-0,0060	0,0062	[-0,0182; 0,0062]
Lazdijų r. sav.	-0,0121	0,0062	[-0,0243; 0,0001]
Skuodo r. sav.	-0,0136	0,0062	[-0,0258; -0,0013]
Vilkaviškio r. sav.	-0,0065	0,0062	[-0,0188; 0,0057]
C Subindeksas			
Kalvarijos sav.	0,0178	0,0051	[0,0078; 0,0279]
Kelmės r. sav.	0,0041	0,0072	[-0,0100; 0,0181]
Lazdijų r. sav.	-0,0144	0,0072	[-0,0284; -0,0003]
Skuodo r. sav.	0,0002	0,0072	[-0,0139; 0,0142]
Vilkaviškio r. sav.	-0,0081	0,0072	[-0,0222; 0,0059]
D Subindeksas			
Kalvarijos sav.	0,0078	0,0118	[-0,0152; 0,0309]
Kelmės r. sav.	-0,0005	0,0166	[-0,0330; 0,0320]
Lazdijų r. sav.	-0,0114	0,0166	[-0,0439; 0,0211]
Skuodo r. sav.	0,0078	0,0166	[-0,0247; 0,0403]
Vilkaviškio r. sav.	-0,0047	0,0166	[-0,0372; 0,0278]
E Subindeksas			
Kalvarijos sav.	-0,0126	0,0083	[-0,0288; 0,0037]
Kelmės r. sav.	-0,0018	0,0117	[-0,0248; 0,0211]
Lazdijų r. sav.	0,0095	0,0117	[-0,0134; 0,0325]
Skuodo r. sav.	0,0081	0,0117	[-0,0149; 0,0310]
Vilkaviškio r. sav.	0,0098	0,0117	[-0,0132; 0,0327]
F Subindeksas			
Kalvarijos sav.	0,0128	0,0085	[-0,0039; 0,0295]
Kelmės r. sav.	-0,0002	0,0120	[-0,0238; 0,0234]
Lazdijų r. sav.	-0,0042	0,0120	[-0,0278; 0,0194]
Skuodo r. sav.	-0,0090	0,0120	[-0,0326; 0,0146]
Vilkaviškio r. sav.	-0,0055	0,0120	[-0,0291; 0,0181]

5 lentelė: GAM modelių įverčiai pagal mažiausią GKI turinčias savivaldybes ir 95% pasiklovimo intervalai

Čia:

Įvertis – Modelio įverčio reikšmė.

Paklaida – Standartinė paklaida (2.16).

95% P.I. – 95% Pasiklovimo intervalas (2.12).

GAM modelių įverčiai pagal mažiausią GKI turinčias savivaldybes, pateikti 5 lentelėje, atskleidė, jog pagal A subindekso duomenis atitiko 95% pasiklivimo intervalus. Didžiausią modelio įverčio koeficientą turėjo Kalvarijos savivaldybė 0,0653, taip pat ši savivaldybė turėjo ir mažiausią standartinės paklaidos reikšmę 0,0164. Kitų savivaldybių standartinės paklaidos reikšmės sutapo ir buvo didesnės, o mažiausią modelio įverčio koeficiento reikšmę turėjo Lazdijų rajono savivaldybė $-0,0108$. B subindekse visi įverčiai taip pat atitiko 95% pasiklivimo intervalus, o didžiausią įvertį turėjo taip pat Kalvarijos savivaldybė 0,0150. Taip pat ši savivaldybė turėjo mažiausią standartinės paklaidos reikšmę 0,0044. Mažiausią modelio įverčio reikšmę, B subindekso duomenimis, turėjo Skuodo rajono savivaldybė $-0,0136$. C subindekso duomenyse taip pat visi įverčiai patenka į 95% pasiklivimo intervalą. Mažiausią įverčio reikšmę čia turėjo Lazdijų r. sav. $-0,0144$, o mažiausią standartinės paklaidos reikšmę turėjo Kalvarijos sav. 0,0051. D subindekso įverčiai taip pat patenka į pasiklivimo intervalo ribas. Mažiausią įverčio reikšmę turėjo Lazdijų r. sav. $-0,0114$, didžiausią – Kalvarijos ir Skuodo rajono savivaldybės 0,0078. E subindekso modelio įverčiai taip pat atitinka 95% pasiklivimo intervalų ribas. Čia didžiausią įverčio reikšmę turėjo Vilkaviškio rajono savivaldybė 0,0098, o mažiausią turėjo Kalvarijos savivaldybė $-0,0126$. F subindekso kiekvienos savivaldybės GAM modelio įverčių koeficientų reikšmės atitinka pasiklivimo intervalų ribas. Čia didžiausią įverčio reikšmę turėjo Kalvarijos savivaldybė 0,0128, o mažiausią – Skuodo rajono savivaldybė $-0,0090$.

Savivaldybė	Įvertis	Paklaida	95% P.I.
A Subindeksas			
Birštono sav.	0,0856	0,0228	[0,0409; 0,1303]
Kauno m. sav.	-0,0059	0,0322	[-0,0691; 0,0573]
Kauno r. sav.	0,0035	0,0322	[-0,0597; 0,0667]
Klaipėdos m. sav.	-0,0207	0,0322	[-0,0839; 0,0425]
Vilniaus m. sav.	0,0001	0,0322	[-0,0631; 0,0633]
B Subindeksas			
Birštono sav.	0,0334	0,0237	[-0,0131; 0,0798]
Kauno m. sav.	0,0131	0,0335	[-0,0526; 0,0787]
Kauno r. sav.	0,0273	0,0335	[-0,0384; 0,0930]
Klaipėdos m. sav.	0,0129	0,0335	[-0,0528; 0,0786]
Vilniaus m. sav.	0,0567	0,0335	[-0,0090; 0,1224]
C Subindeksas			
Birštono sav.	0,0225	0,0111	[0,0007; 0,0443]
Kauno m. sav.	0,0022	0,0157	[-0,0286; 0,0329]
Kauno r. sav.	-0,0045	0,0157	[-0,0353; 0,0262]
Klaipėdos m. sav.	-0,0040	0,0157	[-0,0347; 0,0268]
Vilniaus m. sav.	-0,0063	0,0157	[-0,0370; 0,0245]
D Subindeksas			
Birštono sav.	0,0180	0,0174	[-0,0161; 0,0521]
Kauno m. sav.	-0,0093	0,0246	[-0,0575; 0,0390]
Kauno r. sav.	-0,0038	0,0246	[-0,0521; 0,0444]
Klaipėdos m. sav.	-0,0077	0,0246	[-0,0559; 0,0405]
Vilniaus m. sav.	-0,0178	0,0246	[-0,0660; 0,0305]
E Subindeksas			
Birštono sav.	0,0042	0,0180	[-0,0311; 0,0395]
Kauno m. sav.	-0,0059	0,0255	[-0,0558; 0,0440]
Kauno r. sav.	0,0103	0,0255	[-0,0396; 0,0602]
Klaipėdos m. sav.	-0,0124	0,0255	[-0,0623; 0,0375]
Vilniaus m. sav.	-0,0135	0,0255	[-0,0634; 0,0364]
F Subindeksas			
Birštono sav.	0,0279	0,0104	[0,0076; 0,0482]
Kauno m. sav.	-0,0196	0,0146	[-0,0483; 0,0091]
Kauno r. sav.	-0,0190	0,0146	[-0,0477; 0,0097]
Klaipėdos m. sav.	-0,0165	0,0146	[-0,0452; 0,0122]
Vilniaus m. sav.	-0,0173	0,0146	[-0,0460; 0,0114]

6 lentelė: GAM modelių įverčiai pagal didžiausią GKI turinčias savivaldybes ir 95% pasiklovimo intervalai

Čia:

Įvertis – Modelio įverčio reikšmė.

Paklaida – Standartinė paklaida (2.16).

95% P.I. – 95% Pasiklovimo intervalas (2.12).

Kalbant apie GAM modelio koeficientus ir pasiklovimo intervalus didžiausią GKI turinčioms savivaldybėms, iš 6 lentelės matome, jog pagal A subindekso duomenis visi įverčiai atitiko 95% pasiklovimo intervalą. Pagal A subindekso duomenis didžiausią įverčio reikšmę turėjo Birštono savivaldybė 0,0856, o mažiausią turėjo Klaipėdos miesto savivaldybė $-0,0207$. Tuomet pagal B subindekso duomenis visi įverčiai taip pat atitiko pasiklovimo intervalus. Didžiausią įverčio reikšmę turėjo Vilniaus miesto savivaldybė 0,0567, o mažiausią – Klaipėdos miesto savivaldybė 0,0129. C subindekso modelio įverčiai atitiko 95% pasiklovimo intervalus. Didžiausią įverčio reikšmę turėjo Birštono savivaldybė 0,0225, taip pat ši savivaldybė turėjo ir mažiausią standartinės paklaidos reikšmę 0,0111, o mažiausią įverčio reikšmę turėjo Vilniaus miesto savivaldybė $-0,0063$. Pagal D subindekso duomenis visi įverčiai atitiko pasiklovimo intervalų ribas. Čia didžiausią įverčio reikšmę turėjo taip pat Birštono savivaldybė 0,0180, o mažiausią turėjo vėlgi Vilniaus miesto savivaldybė $-0,0178$. E bei F subindeksų duomenyse matome, jog GAM modelių įverčiai visur atitiko pasiklovimo intervalus. Didžiausią įverčio reikšmę, E subindekso duomenimis, turėjo Kauno rajono savivaldybė 0,0103, o mažiausią – Vilniaus miesto savivaldybė $-0,0135$. Birštono savivaldybė pagal F subindekso duomenis turėjo didžiausią modelio įverčio koeficiento reikšmę 0,0279, o mažiausią turėjo Kauno miesto savivaldybė $-0,0196$.

3.2.3 LSTM modelio pritaikymas

Trečiasis modelis, taikytas savivaldybių subindeksams, o po to ir gyvenimo kokybės indeksui prognozuoti, buvo LSTM (Long Short-Term Memory) modelis. Tiek LSTM, tiek sekantis BiLSTM, modeliai buvo sudarinėjami su „Python“ programa, kadangi tai yra giliojo mokymosi modeliai. LSTM modelis buvo parinktas, atsižvelgiant į aktualią literatūrą [3, 4, 5, 6]. Modelis buvo pasirinktas ir dėl savo patikimumo prognozuojant laiko eilučių duomenis. Šis, giliojo neuroninio tinklo, modelis įsisavina duomenų informaciją per ilgą laikotarpį ir geba ignoruoti triukšmą, tad jis yra tinkamas panelinių duomenų analizei bei prognozavimui.

Prieš sudarinėjant modelį, buvo pasitelkta ta pati transformuotų duomenų struktūra, kaip ir su ARIMA bei GAM modeliais. Tuomet buvo tikrinamas duomenų stacionarumas – svarbiausia laiko eilučių duomenų analizės prielaida. Stacionarumas buvo tikrinamas su ADF (Augmented Dickey – Fuller) testu. Jei duomenyse buvo aptiktas nestacionarumas, buvo atliktas diferencijavimas ir pritaikyti skirtumai laike. Kadangi LSTM modelis yra jautrus duomenų skalės skirtumams, buvo atliktas normalizavimas. Normalizavimas visas duomenų reikšmes sumažino iki intervalo $[0, 1]$, taip užtikrindamas efektyvesnį modelio veikimą bei tikslesnės ateities prognozes. Normalizavimui atlikti buvo pasitelkta *MinMaxScaler* funkcija.

LSTM modelis buvo sukurtas, naudojant šias 3 komponentes: Įvesties sluoksnis, LSTM sluoksnis bei tankio sluoksnis. Įvesties (Input) sluoksnis buvo 3 laiko žingsnių. Šios 3 ankstesnės duomenų reikšmės padėjo prognozuoti 4-ąją. Tuomet pagrindinė neuronų struktūra šiame modelyje buvo 1 LSTM sluoksnis su 50 neuronų. Šis sluoksnis fiksuoja laiko priklausomybes, kurios yra duomenyse. Taip pat, šis sluoksnis turi sudėtingą vidinę struktūrą, kurią sudaro LSTM ląstelės su 3 pagrindiniais vartais: Įėjimo vartai (Input Gate), Pamišimo vartai (Forget Gate) ir Išėjimo vartai (Output Gate). Įėjimo vartai kontroliuoja, kokią naują informaciją iš įvesties įtraukti į ląstelės būseną. Pamišimo vartai nusprendžia, kuri informacija iš ląstelės būsenos turi būti pamišta, bei išėjimo vartai, kurie skirti, jog nuspręstų, kuri dalis ląstelės būsenos turėtų būti išvedama į kitą sluoksnį arba gaunama kaip

galutinę prognozę. Ir paskutinytis – Tankio (Dense) sluoksnis, kuris sujungia LSTM sluoksnio išėjimą su vienintele prognozuota išvesties reikšme. Modelis buvo taikytas, naudojant Adam optimizatorių, kuris yra dažniausiai naudojamas giliojo mokymosi algoritmuose dėl greito ir efektyvaus optimizavimo. Modelio optimizavimas yra būtinas tam, jog LSTM modelis galėtų mokytis. Nuostolių funkcija modelyje buvo vidutinio kvadratinio nuokrypio (MSE). Ji padėjo nustatyti, kaip mokymosi metu modelis atnaujiną savo svorius. Norint išvengti modelio „persimokymo“ (overfitting), buvo naudojama *EarlyStopping* funkcija. Jei modelio nuostolių reikšmė nesumažėja per 2 epochas iš eilės – modelio mokymasis yra sustabdomas. Tai padėjo išvengti per daug gero modelio pritaikymo treniravimo duomenims ir netikslių ateities prognozių. Modelio treniravimas vyko per 5 epochas ir jo treniravimas bei testavimas buvo atliekamas su tais pačiais duomenimis. Toliau buvo atliekama vieno žingsnio prognozė subindeksams kiekvienai iš savivaldybių.

A subindeksas						
Savivaldybė	1E	2E	3E	4E	5E	Pokytis %
Kalvarijos sav.	0,4541	0,4395	0,4252	0,4113	0,3976	12,44%
Kelmės r. sav.	0,5689	0,5518	0,5349	0,5183	0,5018	11,79%
Lazdijų r. sav.	0,2969	0,2851	0,2737	0,2627	0,2520	15,12%
Skuodo r. sav.	0,6341	0,6122	0,5906	0,5693	0,5481	13,56%
Vilkaviškio r. sav.	0,3218	0,3077	0,2941	0,2809	0,2683	16,63%
B subindeksas						
Kalvarijos sav.	0,3365	0,3265	0,3168	0,3072	0,2979	11,47%
Kelmės r. sav.	0,1244	0,1177	0,1113	0,1051	0,0992	20,26%
Lazdijų r. sav.	0,7394	0,7142	0,6896	0,6655	0,6420	13,17%
Skuodo r. sav.	0,4956	0,4728	0,4503	0,4281	0,4063	18,02%
Vilkaviškio r. sav.	0,2765	0,2698	0,2632	0,2568	0,2506	9,37%
C subindeksas						
Kalvarijos sav.	0,3018	0,2931	0,2846	0,2764	0,2684	11,07%
Kelmės r. sav.	0,2040	0,2001	0,1964	0,1928	0,1893	7,21%
Lazdijų r. sav.	0,5684	0,5500	0,5320	0,5144	0,4972	12,53%
Skuodo r. sav.	0,4811	0,4616	0,4425	0,4238	0,4055	15,71%
Vilkaviškio r. sav.	0,2828	0,2742	0,2660	0,2580	0,2502	11,53%
D subindeksas						
Kalvarijos sav.	0,3071	0,2989	0,2910	0,2833	0,2757	10,22%
Kelmės r. sav.	0,5727	0,5534	0,5344	0,5157	0,4974	13,15%
Lazdijų r. sav.	0,5195	0,5029	0,4867	0,4709	0,4554	12,34%
Skuodo r. sav.	0,4056	0,3947	0,3841	0,3736	0,3634	10,40%
Vilkaviškio r. sav.	0,3432	0,3316	0,3205	0,3097	0,2993	12,79%
E subindeksas						
Kalvarijos sav.	0,3733	0,3643	0,3555	0,3471	0,3389	9,22%
Kelmės r. sav.	0,4842	0,4694	0,4549	0,4409	0,4273	11,75%
Lazdijų r. sav.	0,3836	0,3680	0,3527	0,3379	0,3236	15,64%
Skuodo r. sav.	0,3364	0,3245	0,3131	0,3022	0,2917	13,29%
Vilkaviškio r. sav.	0,3680	0,3561	0,3446	0,3333	0,3225	12,36%
F subindeksas						
Kalvarijos sav.	0,1727	0,1639	0,1555	0,1472	0,1393	19,34%
Kelmės r. sav.	0,4766	0,4566	0,4372	0,4182	0,3998	16,11%
Lazdijų r. sav.	0,2077	0,2006	0,1937	0,1872	0,1810	12,86%
Skuodo r. sav.	0,2294	0,2236	0,2180	0,2126	0,2075	9,55%
Vilkaviškio r. sav.	0,3435	0,3299	0,3167	0,3039	0,2915	15,14%

7 lentelė: LSTM modelio mokymosi nuostoliai per 5 epochas ir procentinis sumažėjimas mažiausią GKI turinčioms savivaldybėms

Čia:

1E – 1 Epochos reikšmė, **2E** – 2 Epochos reikšmė ir t.t.

Žvelgiant į 7 lentelę, kuri atspindi LSTM Modelio mokymosi nuostolius per 5 epochas, mažiausią GKI turinčioms savivaldybėms, matome visų savivaldybių pagal 6 subindeksus 5 epochų konvergavimą treniruojant LSTM modelį. Kuomet pirmą kartą buvo sudarinėjamas LSTM modelis, epochų skaičius buvo 10, tačiau dėl per ilgo modelio suveikimo

laiko, epochų skaičius buvo pakeistas į 5. Penkios epochos buvo pasirinktos, atsižvelgiant į geriausią modelio prisitaikymą prie duomenų, nesukeliant prieš tai minėtos modelio persimokymo „overfitting“ problemos. Pasirinkus šį epochų skaičių, modelis padėjo sumažinti nuostolių funkcijos reikšmes bei padidinti savo tikslumą. Pagal A subindeksą, iš 7 lentelės matome, kaip sumažėjo Mažiausių kvadratų nuostolių funkcijos reikšmės nuo pirmos iki penktos epochos kiekvienai savivaldybei. Didžiausią nuostolių funkcijos sumažėjimą patyrė Vilkaviškio rajono savivaldybė. Jos nuostolių funkcijos reikšmė sumažėjo nuo 0,3218 pirmoje epochoje iki 0,2683 penktoje epochoje, procentaliai tai yra 16,63%. Mažiausiai nuostolių funkcija sumažėjo Kelmės rajono savivaldybėje 11,79%. Kalbant apie B subindekso duomenis, matome, jog labiausiai nuostolių funkcija sumažėjo Kelmės rajono savivaldybei 20,26%, o mažiausiai – Vilkaviškio rajono savivaldybei 9,37%. C subindekso modelio epochų konvergavimo rezultatuose matome, jog labiausiai nuostolių funkcijos reikšmė sumažėjo Skuodo rajono savivaldybėje 15,71%, o mažiausiai Kelmės rajono savivaldybėje 7,21%. Tuomet matome, jog pagal D subindeksą didžiausias reikšmių pokytis buvo Kelmės rajono savivaldybėje ir atitiko 13,15%, o mažiausias buvo Kalvarijos savivaldybėje ir atitiko 10,22%. E subindekso rezultatai rodo, jog didžiausią nuostolių funkcijos reikšmę turėjo Lazdijų r. sav., kuri sumažėjo nuo 0,3836 iki 0,3236 (15,64%), o mažiausiai pakitusią reikšmę turėjo Kalvarijos savivaldybė – nuo 0,3733 iki 0,3389 (9,22%). Ir paskutiniojo – F subindekso rezultatai rodo, jog didžiausias nuostolių funkcijos pokytis buvo Kalvarijos savivaldybėje 19,34%, o mažiausias – Skuodo rajono savivaldybėje 9,55%.

A subindeksas						
Savivaldybė	1E	2E	3E	4E	5E	Pokytis %
Birštono sav.	0,5181	0,5001	0,4825	0,4655	0,4490	13,34%
Kauno m. sav.	0,1360	0,1297	0,1235	0,1176	0,1119	17,72%
Kauno r. sav.	0,3671	0,3583	0,3498	0,3415	0,3334	9,18%
Klaipėdos m. sav.	0,6499	0,6295	0,6095	0,5899	0,5706	12,20%
Vilniaus m. sav.	0,3183	0,3058	0,2936	0,2819	0,2705	15,02%
B subindeksas						
Birštono sav.	0,2251	0,2185	0,2121	0,2059	0,2000	11,15%
Kauno m. sav.	0,3488	0,3358	0,3230	0,3106	0,2984	14,45%
Kauno r. sav.	0,4758	0,4645	0,4534	0,4424	0,4314	9,33%
Klaipėdos m. sav.	0,3698	0,3552	0,3410	0,3274	0,3142	15,04%
Vilniaus m. sav.	0,3586	0,3482	0,3380	0,3280	0,3182	11,27%
C subindeksas						
Birštono sav.	0,3921	0,3805	0,3691	0,3581	0,3473	11,43%
Kauno m. sav.	0,2248	0,2207	0,2167	0,2128	0,2090	7,03%
Kauno r. sav.	0,3983	0,3847	0,3714	0,3584	0,3457	13,21%
Klaipėdos m. sav.	0,3021	0,2930	0,2843	0,2758	0,2676	11,42%
Vilniaus m. sav.	0,2061	0,2028	0,1996	0,1965	0,1936	6,07%
D subindeksas						
Birštono sav.	0,5710	0,5511	0,5317	0,5130	0,4949	13,33%
Kauno m. sav.	0,2650	0,2522	0,2398	0,2278	0,2162	18,42%
Kauno r. sav.	0,2938	0,2848	0,2761	0,2676	0,2594	11,71%
Klaipėdos m. sav.	0,3331	0,3193	0,3061	0,2933	0,2809	15,67%
Vilniaus m. sav.	0,5454	0,5287	0,5125	0,4967	0,4812	11,77%
E subindeksas						
Birštono sav.	0,3501	0,3371	0,3244	0,3123	0,3005	14,17%
Kauno m. sav.	0,3994	0,3861	0,3732	0,3608	0,3487	12,69%
Kauno r. sav.	0,3809	0,3696	0,3586	0,3480	0,3377	11,34%
Klaipėdos m. sav.	0,1978	0,1930	0,1885	0,1843	0,1804	8,80%
Vilniaus m. sav.	0,5833	0,5605	0,5386	0,5174	0,4969	14,81%
F subindeksas						
Birštono sav.	0,3148	0,3019	0,2894	0,2773	0,2657	15,60%
Kauno m. sav.	0,7945	0,7687	0,7437	0,7193	0,6955	12,46%
Kauno r. sav.	0,2364	0,2248	0,2135	0,2024	0,1917	18,91%
Klaipėdos m. sav.	0,2125	0,2072	0,2022	0,1974	0,1930	9,18%
Vilniaus m. sav.	0,3187	0,3082	0,2981	0,2885	0,2793	12,36%

8 lentelė: LSTM modelio mokymosi nuostoliai per 5 epochas ir procentinis sumažėjimas didžiausią GKI turinčioms savivaldybėms

Čia:

1E – 1 Epochos reikšmė, **2E** – 2 Epochos reikšmė ir t.t.

Kalbant apie LSTM modelio epochų rezultatus, pagal didžiausią GKI turinčias savivaldybes, pateiktas 8 lentelėje matome, jog Mažiausių kvadratų nuostolių funkcijos didžiausias pokytis buvo Kauno miesto savivaldybėje. Čia funkcijos reikšmės nuo 1 iki 5 epochos kito nuo 0,1360 iki 0,1119 (17,72%), tuo tarpu mažiausias pokytis buvo Kauno rajono

savivaldybėje – nuo 0,3671 iki 0,3334 (9,18%). B subindekso rezultatuose matome, jog didžiausias pokytis buvo Klaipėdos miesto savivaldybėje 15,04%, o mažiausias – Kauno rajono savivaldybėje 9,33%. Tuo tarpu pastaroji, Kauno rajono savivaldybė, pagal C subindekso duomenis turėjo didžiausią nuostolių funkcijos pokytį 13,21%, o mažiausią turėjo Vilniaus miesto savivaldybė 6,07%. Kalbant apie D subindekso duomenis, čia didžiausią Mažiausių kvadratų nuostolių funkcijos reikšmę turėjo Kauno miesto savivaldybė 18,42%, o mažiausią – Kauno rajono savivaldybė 11,71%. Paskutiniuosiuose E ir F subindeksuose didžiausias nuostolių funkcijų reikšmės turėjo atitinkamai Vilniaus miesto 14,81% ir Kauno rajono savivaldybės 18,91%, o mažiausią procentinį pokytį abiejuose subindeksuose turėjo Klaipėdos miesto savivaldybė, atitinkamai 8,8% bei 9,18%. Visų kitų savivaldybių nuostolių funkcijų reikšmės per 5 epochas galima pamatyti 8 lentelėje.

3.2.4 BiLSTM modelio pritaikymas

Ketvirtas modelis, taikytas savivaldybių laiko eilučių duomenims prognozuoti, buvo BiLSTM (Bidirectional Long Short-Term Memory) modelis. Šis dvikryptis modelis yra patobulinta LSTM modelio versija. Šio modelio prognozavimo technika išsiskiria tuo, jog yra sudaryta iš 2 kryptių. Pirmoji kryptis, einanti į priekį, arba kitaip – „Forward”, remiasi ankstesniais laiko žingsniais, o sekanti, atgalinė kryptis „Backward”, analizuoja praeities duomenų informaciją, atsižvelgdama į būsimus laiko žingsnius, sudarinėjant prognozes. Pagrindinis skirtumas tarp LSTM ir BiLSTM modelių yra toks, jog LSTM modelis prognozuoja tik iš praeities duomenų, o tuo tarpu BiLSTM modelis prognozuoja 2 kryptimis – iš praeities į dabartį ir iš ateities į dabartį. Prognozavimas 2 kryptimis pagerina dabarties reikšmių modeliavimą ir modeliuotų reikšmių prognozių tikslumą. BiLSTM modelis buvo sudarytas, naudojantis „Python” programa ir parinktas pagal aktualią literatūrą [5]. Tad BiLSTM modelis buvo sudarytas, norint palyginti šio bei kito giliojo mokymosi modelio LSTM prognozes.

Sudarinėjant BiLSTM modelį, buvo analizuojami savivaldybių laiko eilučių duomenys, kurie buvo transformuoti identišškai, kaip ir prieš tai aprašytų modelių analizėje. Transformuotiems duomenims buvo tikrinama stacionarumo prielaida, naudojantis ADF (Augmented Dickey–Fuller) testu. Jei šio testo p -reikšmė buvo didesnė, nei 0,05, reiškė, jog duomenys buvo nestacionarūs ir buvo atliekamas diferencijavimas. Taip pat duomenys buvo normalizuoti, naudojant *MinMaxScaler* funkciją, intervale [0;1]. Normalizavimas padeda užtikrinti efektyvesnį modelio veikimą ir tikslesnes prognozes, nes duomenys būna toje pačioje skalėje.

BiLSTM modelis buvo sukurtas, naudojant 3 komponentes: Įvesties sluoksnis, Dvikryptis LSTM sluoksnis ir Tankio sluoksnis. Įvesties (Input) sluoksnis turėjo 3 laiko žingsnius, kaip ir LSTM modelyje. Tuomet, pagrindinis Dvikryptis LSTM sluoksnis turėjo 50 neuronų ir veikė 2 kryptimis (Forward ir Backward). Šio sluoksnio ląstelės viduje naudojami 3 pagrindiniai vartai (Įėjimo, Pamiršimo ir Išėjimo vartai), kurie reguliuoja, kokią informaciją išsaugoti arba atmesti kiekviename laiko žingsnyje. Paskutinis – Tankio (Dense) sluoksnis susieja dvikrypčio LSTM sluoksnio išėjimą su viena galutine prognoze. BiLSTM modelis buvo optimizuotas, naudojant efektyvų Adam optimizavimo algoritmą bei Vidutinio kvadratinio nuokrypio nuostolių funkciją. Taip pat, norint išvengti „persimokymo” (overfitting) problemos, buvo naudojama *EarlyStopping* funkcija, jei modelio nuostoliai per 2 epochas nesumažėjo. Modelis buvo treniruojamas per 5 epochas. Modelio treniravimas ir testavimas vyko su tais pačiais duomenimis ir toliau buvo atliekama 1 žingsnio

prognozė subindeksams.

A subindeksas						
Savivaldybė	1E	2E	3E	4E	5E	Pokytis %
Kalvarijos sav.	0,4213	0,4003	0,3798	0,3600	0,3408	19,11%
Kelmės r. sav.	0,5691	0,5402	0,5120	0,4846	0,4579	19,54%
Lazdijų r. sav.	0,3052	0,2875	0,2708	0,2548	0,2398	21,43%
Skuodo r. sav.	0,7925	0,7515	0,7115	0,6725	0,6345	19,94%
Vilkaviškio r. sav.	0,3083	0,2890	0,2707	0,2535	0,2374	23,00%
B subindeksas						
Kalvarijos sav.	0,3224	0,3086	0,2954	0,2826	0,2704	16,13%
Kelmės r. sav.	0,1679	0,1545	0,1420	0,1302	0,1192	29,01%
Lazdijų r. sav.	0,6743	0,6368	0,6005	0,5654	0,5315	21,18%
Skuodo r. sav.	0,7306	0,6886	0,6480	0,6088	0,5710	21,85%
Vilkaviškio r. sav.	0,2943	0,2824	0,2709	0,2599	0,2494	15,26%
C subindeksas						
Kalvarijos sav.	0,2966	0,2830	0,2699	0,2572	0,2450	17,40%
Kelmės r. sav.	0,2159	0,2094	0,2033	0,1977	0,1924	10,88%
Lazdijų r. sav.	0,6402	0,6140	0,5885	0,5637	0,5396	15,71%
Skuodo r. sav.	0,6569	0,6248	0,5936	0,5632	0,5337	18,75%
Vilkaviškio r. sav.	0,2590	0,2491	0,2395	0,2304	0,2216	14,44%
D subindeksas						
Kalvarijos sav.	0,2891	0,2783	0,2679	0,2580	0,2485	14,04%
Kelmės r. sav.	0,6149	0,5871	0,5601	0,5339	0,5085	17,30%
Lazdijų r. sav.	0,4712	0,4472	0,4240	0,4016	0,3801	19,33%
Skuodo r. sav.	0,3759	0,3619	0,3482	0,3349	0,3221	14,31%
Vilkaviškio r. sav.	0,2823	0,2701	0,2586	0,2477	0,2374	15,91%
E subindeksas						
Kalvarijos sav.	0,3535	0,3428	0,3325	0,3225	0,3130	11,46%
Kelmės r. sav.	0,4289	0,4065	0,3853	0,3652	0,3462	19,28%
Lazdijų r. sav.	0,3384	0,3200	0,3024	0,2854	0,2692	20,45%
Skuodo r. sav.	0,3467	0,3301	0,3142	0,2991	0,2846	17,91%
Vilkaviškio r. sav.	0,3162	0,3018	0,2880	0,2749	0,2624	17,01%
F subindeksas						
Kalvarijos sav.	0,1495	0,1389	0,1288	0,1190	0,1097	26,62%
Kelmės r. sav.	0,4563	0,4324	0,4091	0,3866	0,3648	20,05%
Lazdijų r. sav.	0,2602	0,2458	0,2324	0,2197	0,2080	20,06%
Skuodo r. sav.	0,2319	0,2245	0,2174	0,2107	0,2044	11,86%
Vilkaviškio r. sav.	0,2929	0,2766	0,2609	0,2458	0,2313	21,03%

9 lentelė: BiLSTM modelio mokymosi nuostoliai per 5 epochas ir procentinis sumažėjimas mažiausią GKI turinčioms savivaldybėms

Čia:

1E – 1 Epochos reikšmė, 2E – 2 Epochos reikšmė ir t.t.

Šioje 9 lentelėje matome BiLSTM modelio mokymosi nuostolių rezultatus per 5 epochas, mažiausią GKI turinčioms savivaldybėms. BiLSTM modelyje epochų mokymosi procesas

yra toks pats, kaip ir LSTM modelyje. Čia taip pat buvo pirma parinktos 10 epochų, tačiau dėl per ilgo modelio veikimo laiko, epochų skaičius buvo pakeistas į 5. Pagal A subindekso rezultatus visoms savivaldybėms matome, jog Mažiausių kvadratų nuostolių funkcijos didžiausias sumažėjimas buvo Vilkaviškio rajono savivaldybėje – nuo 0,3083 iki 0,2374 (23,00%), o mažiausias buvo Kalvarijos savivaldybėje – nuo 0,4213 iki 0,3408 (19,11%). Kalbant apie B subindekso rezultatus, matome, jog nuostolių funkcijos didžiausias skirtumas tarp 1 ir 5 epochos buvo Kelmės rajono savivaldybėje 29,01%, o mažiausias buvo Vilkaviškio rajono savivaldybėje 15,26%. C subindekso rezultatai rodo, jog didžiausias nuostolių funkcijos reikšmės pokytis buvo Skuodo rajono savivaldybėje 18,75%, o mažiausias buvo Kelmės rajono savivaldybėje 10,88%. Tuomet D subindekso epochų rezultatai rodo, jog didžiausias epochų skirtumas buvo Lazdijų rajono savivaldybėje ir siekė 19,33%, tuo tarpu mažiausias 5 epochų skirtumas lyginant nuo 1 iki 5 buvo Kalvarijos savivaldybėje ir siekė 14,04%. Paskutiniuosiuose E ir F subindeksuose atitinkamai didžiausi epochų skirtumai buvo Lazdijų rajono (20,45%) ir Kalvarijos (26,62%) savivaldybėse, o mažiausi skirtumai buvo atitinkamai Kalvarijos 11,46% ir Skuodo rajonų savivaldybėse 11,86%.

A subindeksas						
Savivaldybė	1E	2E	3E	4E	5E	Pokytis %
Birštono sav.	0,4546	0,4321	0,4103	0,3891	0,3686	18,92%
Kauno m. sav.	0,1656	0,1552	0,1453	0,1360	0,1272	23,19%
Kauno r. sav.	0,3286	0,3164	0,3049	0,2940	0,2838	13,63%
Klaipėdos m. sav.	0,6096	0,5834	0,5578	0,5326	0,5079	16,68%
Vilniaus m. sav.	0,2988	0,2826	0,2670	0,2521	0,2377	20,45%
B subindeksas						
Birštono sav.	0,2395	0,2299	0,2208	0,2123	0,2042	14,74%
Kauno m. sav.	0,3818	0,3610	0,3410	0,3220	0,3038	20,43%
Kauno r. sav.	0,4996	0,4798	0,4605	0,4415	0,4230	15,33%
Klaipėdos m. sav.	0,3968	0,3775	0,3591	0,3415	0,3246	18,20%
Vilniaus m. sav.	0,3330	0,3188	0,3050	0,2917	0,2787	16,31%
C subindeksas						
Birštono sav.	0,3643	0,3488	0,3339	0,3194	0,3054	16,17%
Kauno m. sav.	0,2296	0,2238	0,2182	0,2129	0,2078	9,49%
Kauno r. sav.	0,3950	0,3741	0,3540	0,3349	0,3166	19,85%
Klaipėdos m. sav.	0,2867	0,2748	0,2633	0,2522	0,2416	15,73%
Vilniaus m. sav.	0,2043	0,1997	0,1953	0,1912	0,1872	8,37%
D subindeksas						
Birštono sav.	0,4146	0,3920	0,3702	0,3490	0,3286	20,74%
Kauno m. sav.	0,3129	0,2946	0,2770	0,2601	0,2440	22,02%
Kauno r. sav.	0,2849	0,2728	0,2613	0,2504	0,2400	15,76%
Klaipėdos m. sav.	0,3142	0,2973	0,2813	0,2662	0,2519	19,83%
Vilniaus m. sav.	0,4520	0,4325	0,4136	0,3953	0,3776	16,46%
E subindeksas						
Birštono sav.	0,3844	0,3646	0,3459	0,3282	0,3115	18,96%
Kauno m. sav.	0,3249	0,3083	0,2924	0,2773	0,2630	19,05%
Kauno r. sav.	0,3415	0,3284	0,3159	0,3040	0,2926	14,32%
Klaipėdos m. sav.	0,2111	0,2033	0,1959	0,1891	0,1829	13,36%
Vilniaus m. sav.	0,4501	0,4227	0,3964	0,3714	0,3477	22,75%
F subindeksas						
Birštono sav.	0,3419	0,3232	0,3053	0,2882	0,2718	20,50%
Kauno m. sav.	0,7154	0,6768	0,6397	0,6041	0,5700	20,32%
Kauno r. sav.	0,3845	0,3606	0,3378	0,3160	0,2953	23,20%
Klaipėdos m. sav.	0,2070	0,1998	0,1933	0,1874	0,1821	12,03%
Vilniaus m. sav.	0,3443	0,3283	0,3131	0,2987	0,2852	17,17%

10 lentelė: BiLSTM modelio mokymosi nuostoliai per 5 epochas ir procentinis sumažėjimas didžiausią GKI turinčioms savivaldybėms

Čia:

1E – 1 Epochos reikšmė, **2E** – 2 Epochos reikšmė ir t.t.

BiLSTM modelio mokymosi nuostolių 10 lentelėje matome BiLSTM modelio mokymosi nuostolius per 5 epochas, didžiausią GKI turinčioms savivaldybėms. Pagal A subindekso duomenis, didžiausias pokytis tarp epochų buvo Kauno miesto savivaldybėje – nuo 0,1656 iki 0,1272 (23,19%), o mažiausias buvo Kauno rajono savivaldybėje – nuo 0,3286 iki 0,2838

(13,63%). Pagal B subindekso duomenis matome, jog didžiausias epochų pokytis tarp 1 ir 5 buvo Kauno miesto savivaldybėje – nuo 0,3818 iki 0,3038 (20,43%), o mažiausias Biršto-
no savivaldybėje – nuo 0,2395 iki 0,2042 (14,74%). Kalbant apie C subindekso rezultatus, didžiausią nuostolių funkcijos pokytį turėjo Kauno rajono savivaldybė – nuo 0,3950 iki 0,3166 (19,85%), o mažiausią turėjo Vilniaus miesto savivaldybė – nuo 0,2043 iki 0,1872 (8,37%). D ir E subindeksų rezultatai rodo, jog didžiausi Mažiausių kvadratų nuostolių funkcijos reikšmių pokyčiai buvo atitinkamai Kauno miesto savivaldybėje – nuo 0,3129 iki 0,2440 (22,02%) ir Vilniaus miesto savivaldybėje – nuo 0,4501 iki 0,3477 (22,75%). O ma-
žiausi pokyčiai buvo atitinkamai Kauno rajono savivaldybėje 15,76% bei Klaipėdos miesto savivaldybėje 13,36%. Paskutiniojo F subindekso epochų rezultatai rodo, jog didžiausias pokytis buvo Kauno rajono savivaldybėje 23,20%, o mažiausias – Klaipėdos miesto savi-
valdybėje 12,03%.

3.3 Prognozavimas

3.3.1 ARIMA modelio prognozės

Sudarius 4 modelius subindeksų duomenims, buvo atliktas laiko eilučių prognozavimas 2022 metams, t.y., vieneriems metams į priekį. Norint prognozuoti GKI, visų pirma reikia turėti subindeksų prognozes. Pirmos prognozės buvo atliktos su ARIMA modeliu. Kaip buvo minėta anksčiau, jei laiko eilutė buvo nestacionari, jai buvo taikytas diferencijavimas ir imti skirtumai laike. Kadangi po skirtumų laike 2013 metų reikšmės kai kur tapo tuščios, jas reikėjo pašalinti. Atliekant prognozavimą, kai kur buvo imtos ne originalios duomenų reikšmės, o skirtumai laike. Taigi, tuo atveju, ARIMA modelis buvo pritaikytas jau di-
ferencijuotiems duomenims, tad norint gauti vėl originalias prognozuotas reikšmes reikėjo panaikinti diferencijavimą. Jei laiko eilutei buvo atliktas pirmo lygio diferencijavimas, prie prognozuotos reikšmės buvo pridėdama paskutinė originalios laiko eilutės reikšmė. Jei rei-
kėjo taikyti antro lygio diferencijavimą, tai prognozuotos reikšmės buvo susumuojamos su paskutiniu skirtumu tarp ankstesnių metų ir dar pridėtos prie priešpaskutinės originalios reikšmės ir taip gautos originalios prognozuotos subindeksų reikšmės. Kuomet jau buvo gautos subindeksų prognozės, prognozuotos reikšmės buvo padaugintos iš atitinkamų svo-
rių, susumtuos ir buvo gautas Gyvenimo kokybės indeksas. Gyvenimo kokybės indekso prognozuotoms reikšmėms tikrinti buvo pasitelktos Monte Karlo simuliacijos (žr. 2.2.9 skyrelį). Monte Karlo simuliacijos leidžia įvertinti GKI prognozių tikslumą, atsižvelgiant į kiekvieno subindekso prognozuotas reikšmes. Kiekvienai savivaldybei buvo atliekama 100 simuliacijų. Simuliacijų metu subindeksų reikšmės buvo generuojamos atsitiktinai, laikant, jog skirstinys yra tolygusis. Reikšmės atspindėjo galimus subindeksų svyravimus. Tuomet sugeneruotos subindeksų reikšmės simuliacijų metu buvo padaugintos iš savo svorių ir bu-
vo gautos GKI reikšmės. Iš visų gautų GKI reikšmių, kurių buvo 100, buvo apskaičiuotas GKI vidurkis ir gauti 95% pasiklovimo intervalai. Pasiklovimo intervalai buvo gauti pa-
sitelkiant 2,5% ir 97,5% kvantilius. Ši analizė padėjo įvertinti prognozuotų GKI reikšmių patikimumą ir buvo taikoma su visais kitais likusiais modeliais: GAM, LSTM ir BiLSTM.

Kalbant apie ARIMA modelio prognozių rezultatus pagal mažiausią GKI turinčias savi-
valdybes kiekvienam iš subindeksų, 11 lentelėje matome, jog pagal A subindekso duome-
nis, visos prognozės atitiko 95% pasiklovimo intervalus. Didžiausią liekanų dispersijos ir standartinės paklaidos reikšmę turėjo Lazdijų rajono savivaldybė, atitinkamai 0,01226 ir 0,1107, o mažiausias atitinkamas reikšmes turėjo Kelmės r. sav. 0,001142 ir 0,0316.

Savivaldybė	Progozė	σ^2	Paklaida	95% P.I.
A Subindeksas				
Kalvarijos sav.	0,72768	0,005734	0,0757	[0,5795, 0,8758]
Kelmės r. sav.	0,85726	0,001142	0,0316	[0,7951, 0,9194]
Lazdijų r. sav.	0,72289	0,01226	0,1107	[0,5059, 0,9399]
Skuodo r. sav.	0,81978	0,001824	0,0427	[0,7361, 0,9035]
Vilkaviškio r. sav.	0,76828	0,001231	0,0351	[0,6992, 0,8374]
B Subindeksas				
Kalvarijos sav.	0,13388	0,0003893	0,0197	[0,0954, 0,1724]
Kelmės r. sav.	0,09938	0,0007404	0,0272	[0,0461, 0,1527]
Lazdijų r. sav.	0,06186	0,0001144	0,0107	[0,0409, 0,0828]
Skuodo r. sav.	0,04916	0,0003901	0,0198	[0,0104, 0,0879]
Vilkaviškio r. sav.	0,1044	0,0001238	0,0104	[0,0840, 0,1248]
C Subindeksas				
Kalvarijos sav.	0,37361	0,002666	0,0516	[0,2725, 0,4747]
Kelmės r. sav.	0,4529	0,004664	0,0683	[0,3180, 0,5878]
Lazdijų r. sav.	0,4169	0,0005385	0,0232	[0,3714, 0,4624]
Skuodo r. sav.	0,33993	0,002916	0,0540	[0,2330, 0,4469]
Vilkaviškio r. sav.	0,38497	0,001137	0,0337	[0,3187, 0,4513]
D Subindeksas				
Kalvarijos sav.	0,37402	0,01432	0,1196	[0,1396, 0,6084]
Kelmės r. sav.	0,42818	0,01321	0,1149	[0,2029, 0,6535]
Lazdijų r. sav.	0,41722	0,00702	0,0838	[0,2520, 0,5824]
Skuodo r. sav.	0,49527	0,001233	0,0351	[0,4263, 0,5642]
Vilkaviškio r. sav.	0,35427	0,001262	0,0355	[0,2848, 0,4237]
E Subindeksas				
Kalvarijos sav.	0,42008	0,002103	0,0459	[0,3301, 0,5101]
Kelmės r. sav.	0,23013	0,002761	0,0525	[0,1273, 0,3330]
Lazdijų r. sav.	0,38483	0,00256	0,0506	[0,2857, 0,4840]
Skuodo r. sav.	0,20834	0,005249	0,0724	[0,0664, 0,3503]
Vilkaviškio r. sav.	0,32588	0,00175	0,0418	[0,2430, 0,4087]
F Subindeksas				
Kalvarijos sav.	0,47743	0,001	0,0316	[0,4155, 0,5393]
Kelmės r. sav.	0,56753	0,002117	0,0460	[0,4763, 0,6588]
Lazdijų r. sav.	0,54469	0,0009287	0,0305	[0,4847, 0,6047]
Skuodo r. sav.	0,57452	0,0001488	0,0122	[0,5506, 0,5985]
Vilkaviškio r. sav.	0,52398	0,0007618	0,0276	[0,4699, 0,5781]

11 lentelė: ARIMA modelio prognozavimo rezultatai su 95% prognozių pasiklovimo intervalais pagal mažiausią GKI turinčias savivaldybes

Čia:

σ^2 – Liekanų empirinė dispersija (2.8).

Paklaida – Standartinė paklaida (2.7).

95% P.I. – 95% Pasiklovimo intervalas (2.12).

B subindekso modelio rezultatai rodo, jog visos prognozės taip pat atitiko pasiklivimo intervalus. Didžiausią liekanų dispersijos ir standartinės paklaidos reikšmę turėjo Kelmės r. sav., atitinkamai 0,0007404 ir 0,0272, o mažiausią liekanų dispersijos reikšmę turėjo Lazdijų r. sav. 0,0001144. Mažiausią standartinės paklaidos reikšmę turėjo Vilkaviškio r. sav. 0,0104. C subindekso reikšmių prognozės taip pat atitiko 95% pasiklivimo intervalų ribas. Čia didžiausią liekanų dispersijos reikšmę 0,004664 turėjo Kelmės rajono savivaldybė, taip pat ši savivaldybė turėjo ir didžiausią standartinės paklaidos reikšmę 0,0683. Mažiausias standartinės paklaidos ir liekanų dispersijos reikšmes turėjo Lazdijų r. sav. 0,0232 ir 0,0005385. Kalbant apie likusius D, E bei F subindeksus, matome, jog jų prognozės visur atitiko pasiklivimo intervalus. Mažiausias liekanų dispersijos ir standartinės paklaidos reikšmes turėjo Skuodo rajono savivaldybė 0,001233 ir 0,0351. Tuo tarpu didžiausias šių koeficientų reikšmes turėjo Kalvarijos savivaldybė, atitinkamai 0,01432 ir 0,1196. Kalbant apie E subindekso mažiausias liekanų dispersijos ir standartinės paklaidos reikšmes, matome, jog jas turėjo Vilkaviškio r. sav. 0,00175 ir 0,0418. Didžiausias šių koeficientų reikšmes pagal E subindekso duomenis turėjo Skuodo r. sav. 0,005249 ir 0,0724. Ir pagal paskutiniojo F subindekso duomenis matome, jog didžiausias liekanų dispersijos ir standartinės paklaidos reikšmes turėjo Kelmės rajono savivaldybė 0,002117 ir 0,0460, o mažiausias turėjo Skuodo rajono savivaldybė 0,0001488 ir 0,0122.

Žvelgiant į ARIMA modelio prognozių rezultatus kartu su pasiklivimo intervalais pagal didžiausią GKI turinčias savivaldybes, 12 lentelėje matome, jog pagal A subindeksą visos savivaldybių prognozės atitiko 95% pasiklivimo intervalus. Čia mažiausią liekanų dispersijos ir standartinės paklaidos reikšmę turėjo Kauno miesto savivaldybė 0,0007519 ir 0,0256. O didžiausias šių koeficientų reikšmes turėjo Birštono savivaldybė 0,04938 ir 0,2222. Tuomet pagal B subindekso duomenis matome, jog prognozės taip pat atitiko pasiklivimo intervalus. Didžiausias reikšmes tiek liekanų dispersijos, tiek standartinės paklaidos turėjo Klaipėdos miesto savivaldybė 0,03824 ir 0,196, o mažiausias šių koeficientų reikšmes turėjo Kauno rajono savivaldybė 0,0023 ir 0,048. C subindekso rezultatai rodo, jog prognozės taip pat atitiko pasiklivimo intervalus. Didžiausias liekanų dispersijos bei standartinės paklaidos reikšmes turėjo Birštono savivaldybė 0,02696 ir 0,1642, o mažiausias – Klaipėdos miesto savivaldybė, atitinkamai 0,003493 ir 0,0591. Tuomet pagal D subindekso duomenis matome, jog didžiausias pastarųjų koeficientų reikšmes turėjo Vilniaus miesto savivaldybė 0,02984 ir 0,1727, o mažiausias liekanų dispersijos bei standartinės paklaidos reikšmes turėjo Kauno rajono savivaldybė 0,003869 ir 0,0622. E ir F subindeksų duomenys rodo, jog prognozės taip pat atitiko pasiklivimo intervalus. Pagal E subindekso rezultatus matome, jog didžiausias liekanų dispersijos bei standartinės paklaidos reikšmes turėjo Klaipėdos miesto savivaldybė 0,006423 ir 0,0801, o mažiausias – Vilniaus miesto savivaldybė 0,001629 ir 0,0404. Paskutiniojo F subindekso prognozių rezultatai rodo, jog pastarųjų koeficientų didžiausias reikšmes turėjo Birštono savivaldybė, atitinkamai 0,004038 ir 0,0635, o mažiausias liekanų dispersijos bei standartinės paklaidos reikšmes turėjo Vilniaus miesto savivaldybė 0,000301 ir 0,0174.

Turint originalias reikšmes, visi prognozuoti subindeksai buvo susumuoti bei padauginti iš savo svorių (pagal GKI (2.27) formulę). Tuomet buvo gautos GKI reikšmės 2022 metams.

Originalios subindeksų ir GKI prognozės buvo lyginamos su 2022 metų tikrais visų subindeksų bei GKI duomenimis. Tam buvo naudojamos RMSE, MSE ir MAPE metrikos, kurios parodo įvairias paklaidas tarp tikrų ir prognozuotų reikšmių. Kadangi prognozavimas buvo atliekamas ta pačia tvarka kaip ir modeliavimas, lentelėse ARIMA_MIN stulpelis atspindi prasčiausių savivaldybių tikrų ir prognozuotų rezultatų skirtumus, o stulpelis

Savivaldybė	Progozė	σ^2	Paklaida	95% P.I.
A Subindeksas				
Birštono sav.	1,07778	0,04938	0,2222	[0,6423, 1,5133]
Kauno m. sav.	1,25017	0,0007519	0,0256	[1,2000, 1,3003]
Kauno r. sav.	1,27549	0,001318	0,0340	[1,2080, 1,3429]
Klaipėdos m. sav.	1,14010	0,0009873	0,0294	[1,0825, 1,1977]
Vilniaus m. sav.	1,40685	0,0008999	0,0281	[1,3518, 1,4619]
B Subindeksas				
Birštono sav.	0,35679	0,006557	0,081	[0,1972, 0,5164]
Kauno m. sav.	0,76415	0,005333	0,073	[0,6203, 0,9080]
Kauno r. sav.	0,68523	0,0023	0,048	[0,5902, 0,7803]
Klaipėdos m. sav.	0,89076	0,03824	0,196	[0,5064, 1,2751]
Vilniaus m. sav.	1,57352	0,003708	0,061	[1,4531, 1,6939]
C Subindeksas				
Birštono sav.	0,56415	0,02696	0,1642	[0,2424, 0,8859]
Kauno m. sav.	0,79891	0,003994	0,0632	[0,6750, 0,9228]
Kauno r. sav.	0,40153	0,004222	0,0650	[0,2749, 0,5282]
Klaipėdos m. sav.	0,71792	0,003493	0,0591	[0,6011, 0,8347]
Vilniaus m. sav.	0,68478	0,003657	0,0605	[0,5662, 0,8034]
D Subindeksas				
Birštono sav.	0,57883	0,01234	0,1111	[0,3601, 0,7976]
Kauno m. sav.	0,67055	0,009564	0,0978	[0,4789, 0,8622]
Kauno r. sav.	0,44913	0,003869	0,0622	[0,3272, 0,5711]
Klaipėdos m. sav.	0,59388	0,005811	0,0762	[0,4445, 0,7433]
Vilniaus m. sav.	0,67671	0,02984	0,1727	[0,3382, 1,0152]
E Subindeksas				
Birštono sav.	0,51493	0,005431	0,0737	[0,3705, 0,6594]
Kauno m. sav.	0,39402	0,004589	0,0677	[0,2613, 0,5267]
Kauno r. sav.	0,70547	0,005186	0,0720	[0,5643, 0,8466]
Klaipėdos m. sav.	0,32519	0,006423	0,0801	[0,1682, 0,4822]
Vilniaus m. sav.	0,53937	0,001629	0,0404	[0,4602, 0,6185]
F Subindeksas				
Birštono sav.	0,83212	0,004038	0,0635	[0,7076, 0,9566]
Kauno m. sav.	0,56328	0,001514	0,0389	[0,4871, 0,6395]
Kauno r. sav.	0,53926	0,0008266	0,0288	[0,4828, 0,5957]
Klaipėdos m. sav.	0,5928	0,001684	0,0410	[0,5124, 0,6732]
Vilniaus m. sav.	0,59248	0,000301	0,0174	[0,5584, 0,6266]

12 lentelė: ARIMA modelio prognozavimo rezultatai su 95% prognozių pasiklivimo intervalais pagal didžiausią GKI turinčias savivaldybes

Čia:

σ^2 – Liekanų empirinė dispersija (2.8).

Paklaida – Standartinė paklaida (2.7).

95% P.I. – 95% Pasiklivimo intervalas (2.12).

ARIMA_MAX geriausių savivaldybių skirtumus.

	ARIMA_MIN	ARIMA_MAX
RMSE	0,02960291	0,069199542
MSE	0,000876333	0,004788577
MAPE	5,6697742%	6,8970725%

13 lentelė: ARIMA modelio metrikų rezultatai

Čia:

RMSE – Šaknis iš vidutinės kvadratinės paklaidos (2.24).

MSE – Vidutinis kvadratinis nuokrypis (2.25).

MAPE – Vidutinė absoliutinė procentinė paklaida (2.26).

Iš 13 lentelės, kurioje atvaizduoti ARIMA modelio metrikų rezultatai, matome, jog ARIMA modelis geriau prognozavo savivaldybes, pasirinktas pagal žemiausią GKI. Čia šaknies iš vidutinės kvadratinės paklaidos reikšmė buvo 0,02, vidutinis kvadratinis nuokrypis buvo tik 0,0008, o vidutinė absoliutinė procentinė paklaida nesiekė 6%. Visos šios 3 metrikos yra kur kas mažesnės, nei savivaldybių pagal aukščiausią GKI.

Savivaldybės	Prognozuotas GKI	Tikras GKI	95% P.I.
Kalvarijos sav.	0,4544695	0,4537610	[0,4109; 0,5068]
Kelmės r. sav.	0,4848102	0,4760570	[0,4474; 0,5300]
Lazdijų r. sav.	0,4520790	0,4734530	[0,3717; 0,5164]
Skuodo r. sav.	0,4567150	0,4527395	[0,4184; 0,4884]
Vilkaviškio r. sav.	0,4527632	0,4648895	[0,4155; 0,4804]

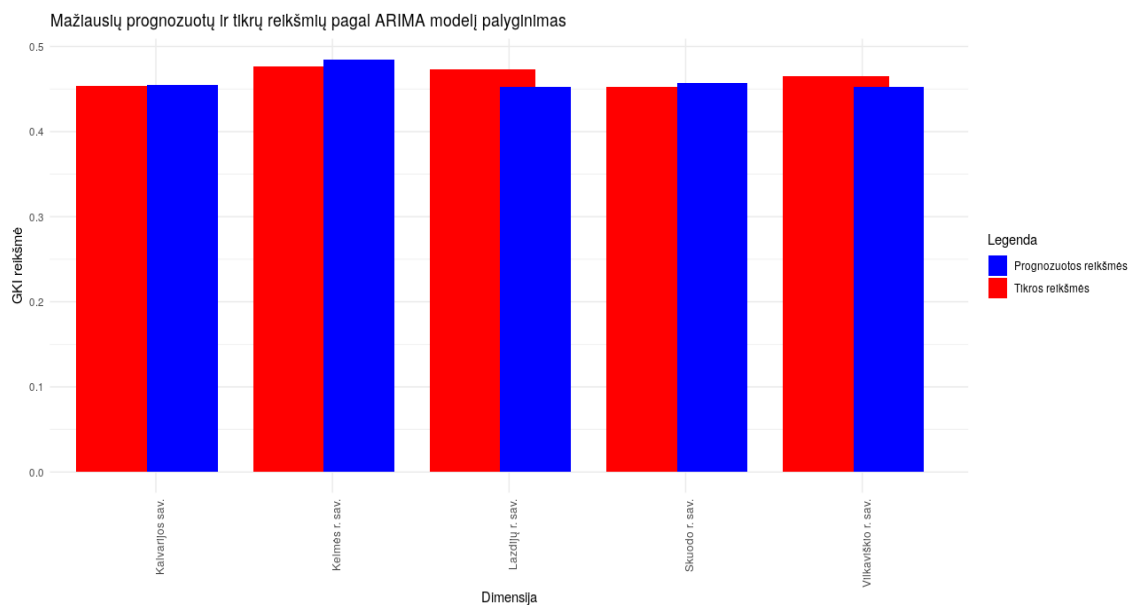
14 lentelė: Mažiausių prognozuotų ir tikrų reikšmių pagal ARIMA modelį palyginimas

Čia:

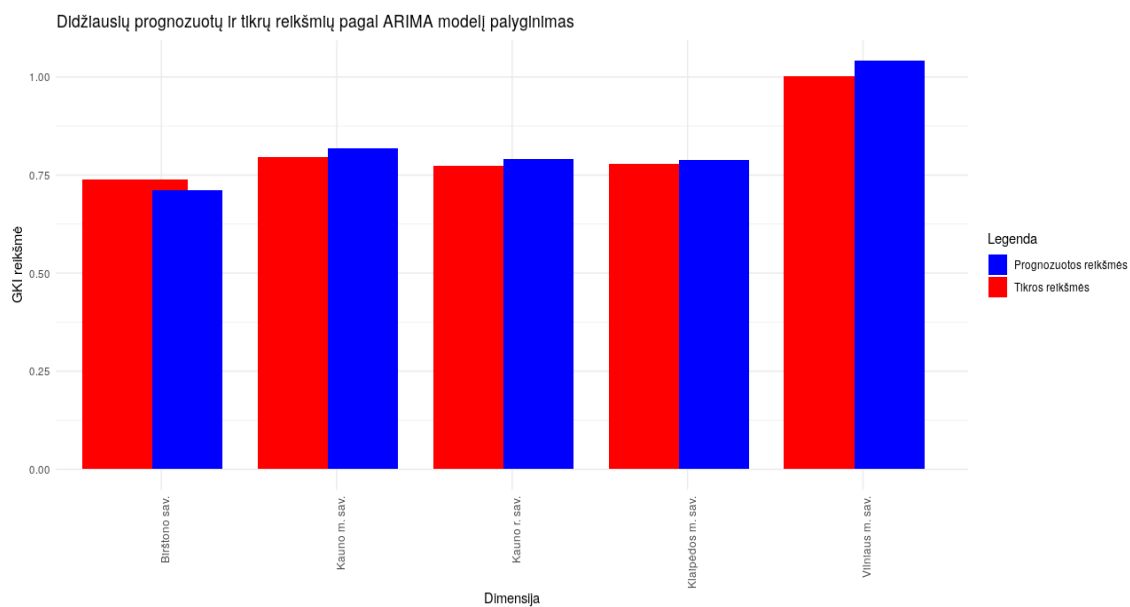
95% P.I. – Monte Karlo simuliacijomis gautas GKI 95% pasiklivimo intervalas; žr. 2.2.9 skyrelį.

Iš mažiausių prognozuotų ir tikrų reikšmių pagal ARIMA modelį palyginimų, pateiktų 14 lentelėje matome, jog pagal prognozuotas ir tikras reikšmes savivaldybių išsirikiavimas skiriasi. Pagal prognozuotas reikšmes, nuo žemiausią GKI turinčios savivaldybės, išsidėstymas atrodo taip: Lazdijų r. sav., Vilkaviškio r. sav., Kalvarijos sav., Skuodo r. sav. bei Kelmės r. sav. O pagal tikras reikšmes atrodo taip: Skuodo r. sav., Kalvarijos sav., Vilkaviškio r. sav., Lazdijų r. sav. ir Kelmės r. sav. Tad vienintelė Kelmės r. sav. atitiko 5-ąją vietą pagal prognozuotas ir tikras reikšmes. Tikras ir prognozuotas reikšmes galima palyginti 1 paveiksle. Taip pat matome, jog pagal ARIMA modelį mažiausią GKI turinčioms savivaldybėms prognozuotos GKI reikšmės atitiko Monte Karlo simuliacijų būdu gautas GKI 95% pasiklivimo intervalų ribas.

Iš didžiausių prognozuotų ir tikrų reikšmių pagal ARIMA modelį palyginimų, pateiktų 15 lentelėje, matome, jog pagal prognozuotas ir tikras reikšmes, pagal aukščiausią GKI, savivaldybių prognozių išsidėstymas atrodo taip: Vilniaus m. sav., Kauno m. sav., Kauno r. sav., Klaipėdos m. sav. bei Birštono sav. O pagal tikras reikšmes atrodo taip: Vilniaus m. sav., Kauno m. sav., Klaipėdos m. sav., Kauno r. sav. bei Birštono sav. Trijų savivaldybių išsirikiavimas sutapo tiek pagal prognozuotas, tiek pagal tikras reikšmes – Vilniaus, Kauno miestų ir Birštono. Reikšmių skirtumai pavaizduoti ir 2 paveiksle. Pagal ARIMA modelį didžiausią GKI turinčioms savivaldybėms prognozuotos reikšmės atitiko 95% pasiklivimo intervalų ribas, tad galime teigti, jog prognozės gavosi pakankamai tikslūs.



1 pav.: Mažiausių prognozuotų ir tikrų reikšmių pagal ARIMA modelį palyginimas



2 pav.: Didžiausių prognozuotų ir tikrų reikšmių pagal ARIMA modelį palyginimas

Savivaldybės	Prognozuotas GKI	Tikras GKI	95% P.I.
Birštono sav.	0,7110475	0,7394860	[0,5888; 0,8590]
Kauno m. sav.	0,8184231	0,7960150	[0,7747; 0,8625]
Kauno r. sav.	0,7914692	0,7740490	[0,7492; 0,8356]
Klaipėdos m. sav.	0,7890612	0,7774145	[0,7012; 0,8719]
Vilniaus m. sav.	1,0426848	1,00141005	[0,9973; 1,1005]

15 lentelė: Didžiausių prognozuotų ir tikrų reikšmių pagal ARIMA modelį palyginimas

Čia:

95% P.I. – Monte Karlo simuliacijomis gautas GKI 95% pasiklivimo intervalas; žr. 2.2.9 skyrelį.

3.3.2 GAM modelio prognozės

Sekantis modelis, kuriuo buvo atliekamas prognozavimas savivaldybių Gyvenimo kokybės indeksui, buvo Apibendrintasis Adityvusis modelis. Šiuo modeliu buvo prognozuoti subindeksai 2022 metams. Iš subindeksų prognozių, padaugintų pagal atitinkamus svorius, sumos buvo gautas GKI. Prognozės buvo atliekamos vieneriems metams į priekį. Sudarinėjant šio modelio prognozes, prieš tai buvo taikomas diferencijavimas nestacionarioms laiko eilutėms. Atlikus diferencijavimą, buvo pašalintos 2013 metų reikšmės, nes jos tapo tuščios. Norint gauti originalias prognozuotas reikšmes, buvo pašalinti skirtumai laike. Jei buvo atliekamas pirmo lygio diferencijavimas, prognozuotai reikšmei atstatyti prie jos buvo pridėdama paskutinė originalios laiko eilutės reikšmė. Jei buvo taikomas antro lygio diferencijavimas, prognozuotos reikšmės buvo sudedamos su paskutiniu skirtumu tarp ankstesnių metų ir ši suma pridėdama prie priešpaskutinės originalios reikšmės. Tokiu būdu buvo atstatytos originalios prognozuotos subindeksų reikšmės, lygiai taip pat, kaip tai buvo daroma atliekant prognozes su ARIMA modeliu.

Iš GAM modelių prognozių, pateiktų 16 lentelėje, matome visų savivaldybių pagal mažiausią GKI subindeksų prognozes 2022 metams bei jų pasiklivimo intervalus. Matome, jog visos prognozės atitinka 95% pasiklivimo intervalus, taip pat ir modelio įverčiai šioms savivaldybėms atitiko pasiklivimo intervalus, tad galime daryti prielaidą, jog prognozės yra pakankamai tikslios ir modelis gerai atitiko duomenis.

Savivaldybė	Metai	Prognozė	95% P.I.
A Subindeksas			
Kalvarijos sav.	2022	0,7628	[0,7199; 0,8056]
Kelmės r. sav.	2022	0,8271	[0,7842; 0,8699]
Lazdijų r. sav.	2022	0,7472	[0,7043; 0,7900]
Skuodo r. sav.	2022	0,8563	[0,8135; 0,8992]
Vilkaviškio r. sav.	2022	0,8044	[0,7616; 0,8472]
B Subindeksas			
Kalvarijos sav.	2022	0,1525	[0,1381; 0,1669]
Kelmės r. sav.	2022	0,1120	[0,0976; 0,1264]
Lazdijų r. sav.	2022	0,0684	[0,0539; 0,0828]
Skuodo r. sav.	2022	0,0542	[0,0398; 0,0686]
Vilkaviškio r. sav.	2022	0,1080	[0,0936; 0,1224]
C Subindeksas			
Kalvarijos sav.	2022	0,3710	[0,3133; 0,4287]
Kelmės r. sav.	2022	0,4543	[0,3966; 0,5120]
Lazdijų r. sav.	2022	0,3999	[0,3422; 0,4576]
Skuodo r. sav.	2022	0,3375	[0,2798; 0,3952]
Vilkaviškio r. sav.	2022	0,3742	[0,3165; 0,4319]
D Subindeksas			
Kalvarijos sav.	2022	0,4060	[0,3419; 0,4701]
Kelmės r. sav.	2022	0,4596	[0,3955; 0,5237]
Lazdijų r. sav.	2022	0,4378	[0,3737; 0,5019]
Skuodo r. sav.	2022	0,5351	[0,4710; 0,5991]
Vilkaviškio r. sav.	2022	0,3815	[0,3174; 0,4456]
E Subindeksas			
Kalvarijos sav.	2022	0,3499	[0,2786; 0,4211]
Kelmės r. sav.	2022	0,1581	[0,0868; 0,2293]
Lazdijų r. sav.	2022	0,3241	[0,2528; 0,3954]
Skuodo r. sav.	2022	0,1462	[0,0749; 0,2174]
Vilkaviškio r. sav.	2022	0,2654	[0,1942; 0,3367]
F Subindeksas			
Kalvarijos sav.	2022	0,4912	[0,4690; 0,5134]
Kelmės r. sav.	2022	0,5811	[0,5589; 0,6033]
Lazdijų r. sav.	2022	0,5543	[0,5321; 0,5765]
Skuodo r. sav.	2022	0,5793	[0,5571; 0,6015]
Vilkaviškio r. sav.	2022	0,5323	[0,5101; 0,5545]

16 lentelė: GAM modelių prognozės pagal mažiausią GKI turinčias savivaldybes ir jų 95% pasiklovimo intervalai

Čia:

95% P.I. – 95% Pasiklovimo intervalas (2.12).

Savivaldybė	Metai	Prognozė	95% P.I.
A Subindeksas			
Birštono sav.	2022	1,1648	[1,1054; 1,2243]
Kauno m. sav.	2022	1,2517	[1,1922; 1,3111]
Kauno r. sav.	2022	1,2770	[1,2175; 1,3364]
Klaipėdos m. sav.	2022	1,1416	[1,0821; 1,2010]
Vilniaus m. sav.	2022	1,4083	[1,3489; 1,4678]
B Subindeksas			
Birštono sav.	2022	0,3960	[0,3342; 0,4578]
Kauno m. sav.	2022	0,8164	[0,7546; 0,8782]
Kauno r. sav.	2022	0,7517	[0,6899; 0,8135]
Klaipėdos m. sav.	2022	0,9428	[0,8810; 1,0046]
Vilniaus m. sav.	2022	1,6694	[1,6076; 1,7313]
C Subindeksas			
Birštono sav.	2022	0,6442	[0,5976; 0,6909]
Kauno m. sav.	2022	0,8811	[0,8345; 0,9278]
Kauno r. sav.	2022	0,4771	[0,4304; 0,5237]
Klaipėdos m. sav.	2022	0,7940	[0,7474; 0,8407]
Vilniaus m. sav.	2022	0,7586	[0,7119; 0,8052]
D Subindeksas			
Birštono sav.	2022	0,6947	[0,5704; 0,8190]
Kauno m. sav.	2022	0,7772	[0,6529; 0,9014]
Kauno r. sav.	2022	0,5612	[0,4369; 0,6854]
Klaipėdos m. sav.	2022	0,7020	[0,5777; 0,8263]
Vilniaus m. sav.	2022	0,7748	[0,6505; 0,8991]
E Subindeksas			
Birštono sav.	2022	0,4980	[0,4511; 0,5450]
Kauno m. sav.	2022	0,3712	[0,3243; 0,4182]
Kauno r. sav.	2022	0,6988	[0,6519; 0,7458]
Klaipėdos m. sav.	2022	0,2959	[0,2489; 0,3428]
Vilniaus m. sav.	2022	0,5090	[0,4620; 0,5559]
F Subindeksas			
Birštono sav.	2022	0,8482	[0,8211; 0,8752]
Kauno m. sav.	2022	0,5597	[0,5327; 0,5867]
Kauno r. sav.	2022	0,5363	[0,5093; 0,5633]
Klaipėdos m. sav.	2022	0,5923	[0,5653; 0,6194]
Vilniaus m. sav.	2022	0,5913	[0,5642; 0,6183]

17 lentelė: GAM modelių prognozės pagal didžiausią GKI turinčias savivaldybes ir jų 95% pasiklovimo intervalai

Čia:

95% P.I. – 95% Pasiklovimo intervalas (2.12).

Iš GAM modelių prognozių pagal didžiausią GKI pateiktų 17 lentelėje, matome, jog visų savivaldybių prognozių rezultatai atitiko 95% pasiklovimo intervalų ribas. Kadangi ir

GAM modelio įverčiai didžiausią GKI turinčioms savivaldybėms pagal visus subindeksus atitiko pasiklivimo intervalų ribas, galime daryti prielaidą, jog prognozės gavosi pakankamai tikslios ir modelio pritaikymas duomenims buvo tinkamas.

Gavus originalias savivaldybių subindeksų prognozes ir iš subindeksų prognozių sumos gautą GKI 2022 metams, šios reikšmės buvo lyginamos su tikrais savivaldybių subindeksų ir GKI duomenimis. Tam buvo pasitelktos 3 metrikos – Šaknis iš vidutinės kvadratinės paklaidos (RMSE), Vidutinis kvadratinis nuokrypis (MSE) ir Vidutinė absoliutinė procentinė paklaida (MAPE). Kadangi prognozavimas buvo atliekamas panašiu duomenų transformavimo principu ir tokia pačia tvarka kaip ir su ARIMA modelio prognozėmis, lentelėse GAM_MIN stulpelis atspindi prasčiausių savivaldybių tikrų ir prognozuotų rezultatų skirtumus, o stulpelis GAM_MAX geriausių savivaldybių rezultatų skirtumus.

	GAM_MIN	GAM_MAX
RMSE	0,047377992	0,189435096
MSE	0,002244674	0,035885656
MAPE	9,1599295%	17,0608229%

18 lentelė: GAM modelio metrikų rezultatai

Čia:

RMSE – Šaknis iš vidutinės kvadratinės paklaidos (2.24).

MSE – Vidutinis kvadratinis nuokrypis (2.25).

MAPE – Vidutinė absoliutinė procentinė paklaida (2.26).

GAM modelio metrikų rezultatai, pateikti 18 lentelėje, atspindi, jog GAM modelis geriau prognozavo savivaldybes, pasirinktas pagal žemiausią GKI. Matome, jog šaknies iš vidutinės kvadratinės paklaidos reikšmė buvo 0,04, vidutinis kvadratinis nuokrypis buvo tik 0,0022, o vidutinė absoliutinė procentinė paklaida buvo 9,15%. Visos šios 3 metrikos yra kur kas mažesnės, nei savivaldybių pagal aukščiausią GKI, tačiau, lyginant su ARIMA modelio metrikomis, ARIMA modelis prognozavo kur kas geriau.

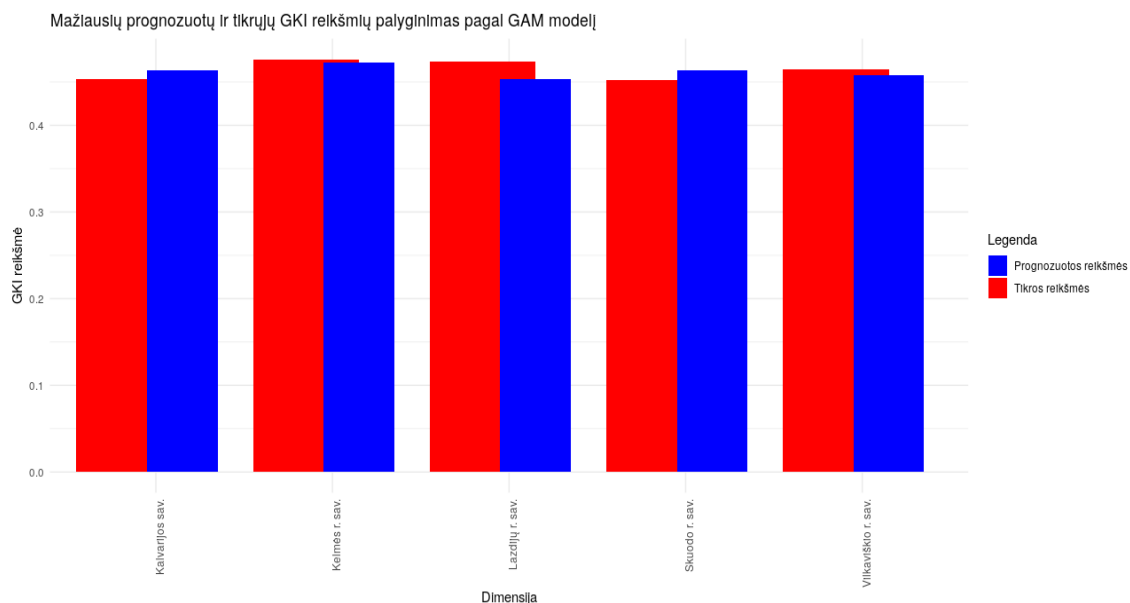
Savivaldybės	Prognozuotas GKI	Tikras GKI	95% P.I.
Kalvarijos sav.	0,4631833	0,4537610	[0,4396; 0,4881]
Kelmės r. sav.	0,4727958	0,4760570	[0,4474; 0,4959]
Lazdijų r. sav.	0,4533531	0,4734530	[0,4298; 0,4720]
Skuodo r. sav.	0,4638170	0,4527395	[0,4427; 0,4813]
Vilkaviškio r. sav.	0,4581520	0,4648895	[0,4376; 0,4793]

19 lentelė: Mažiausių prognozuotų ir tikrų reikšmių pagal GAM modelį palyginimas

Čia:

95% P.I. – Monte Karlo simuliacijomis gautas GKI 95% pasiklivimo intervalas; žr. 2.2.9 skyrelį.

Pagal prognozuotas ir tikras reikšmes savivaldybių išsirikiavimas skiriasi, tačiau yra identiškas, kaip ir su ARIMA modeliu. Reikšmių skirtumus matome ir 19 lentelėje. Pagal prognozuotas reikšmes, nuo žemiausią GKI turinčios savivaldybės, išsidėstymas atrodo taip: Lazdijų r. sav., Vilkaviškio r. sav., Kalvarijos sav., Skuodo r. sav. bei Kelmės r. sav. O pagal tikras reikšmes atrodo taip: Skuodo r. sav., Kalvarijos sav., Vilkaviškio r. sav., Lazdijų r. sav. ir Kelmės r. sav. Tad vienintelė Kelmės r. sav. atitiko 5-ąją vietą pagal prognozuotas ir tikrąsias reikšmes. Taip pat 19 lentelėje matome, jog visos prognozuotos GKI reikšmės atitiko 95% pasiklivimo intervalų ribas, kurios buvo gautos



3 pav.: Mažiausių prognozuotų ir tikrų reikšmių pagal GAM modelį palyginimas

Monte Karlo simuliacijų būdu. Reikšmių skirtumus galima palyginti ir 3 paveiksle.

Savivaldybės	Prognozuotas GKI	Tikras GKI	95% P.I.
Birštono sav.	0,5338101	0,7394860	[0,7340; 0,7897]
Kauno m. sav.	0,7136675	0,7960150	[0,8147; 0,8696]
Kauno r. sav.	0,6649527	0,7740490	[0,7943; 0,8487]
Klaipėdos m. sav.	0,7542423	0,7774145	[0,7877; 0,8409]
Vilniaus m. sav.	1,1530922	1,00141005	[1,0448; 1,1036]

20 lentelė: Didžiausių prognozuotų ir tikrų reikšmių pagal GAM modelį palyginimas

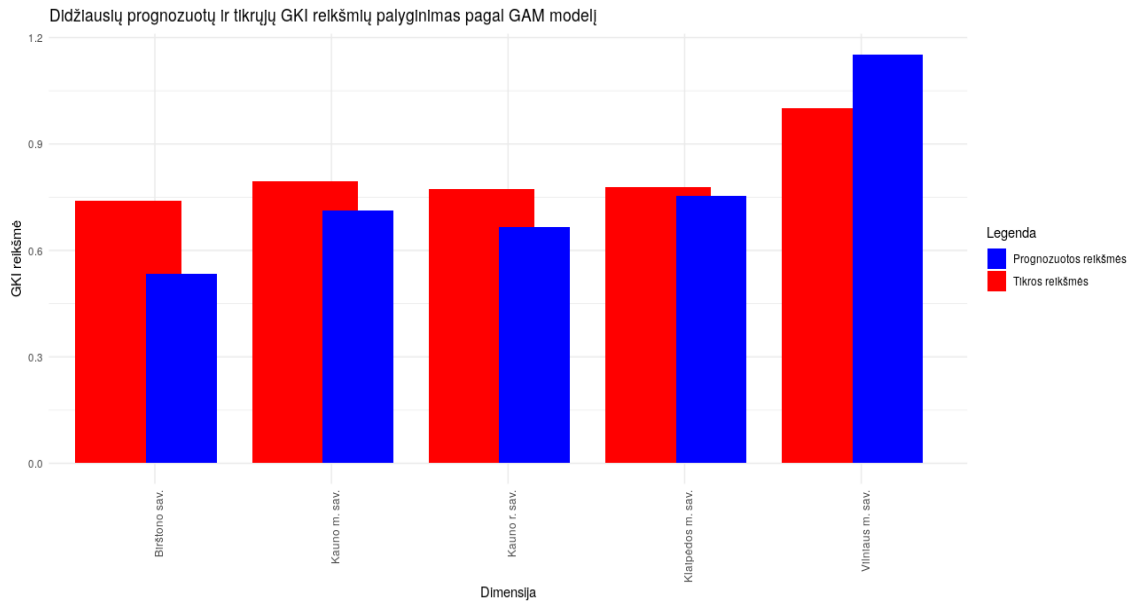
Čia:

95% P.I. – Monte Karlo simuliacijomis gautas GKI 95% pasiklovimo intervalas; žr. 2.2.9 skyrelį.

Prognozuotų ir tikrų reikšmių skirtumai pagal GAM modelį, pateikti 20 lentelėje pagal savivaldybių išsidėstymą daug kur sutapo. Pagal prognozuotas reikšmes savivaldybių išsidėstymas atrodo taip: Vilniaus m. sav., Klaipėdos m. sav., Kauno m. sav., Kauno r. sav., bei Birštono sav. O pagal tikras reikšmes atrodo taip: Vilniaus m. sav., Kauno m. sav., Klaipėdos m. sav., Kauno r. sav. bei Birštono sav. Taigi, pagal išsidėstymą nuo didžiausią GKI turinčių savivaldybių, sutapo Vilniaus miesto savivaldybė, Kauno rajono ir Birštono savivaldybės. Taip pat 20 lentelėje matome, jog prognozuotos GKI reikšmės neatitiko Monte Karlo simuliacijomis gautų pasiklovimo intervalų rėžių. Tikrų ir prognozuotų GKI reikšmių skirtumus galima pamatyti ir 4 paveiksle.

3.3.3 LSTM modelio prognozės

Trečiasis modelis, kuriuo buvo atliekamas laiko eilučių prognozavimas buvo Ilgalaikės trumpalaikės atminties neuronų tinklo (Long Short-Term Memory) modelis. Šis modelis yra tinkamas, norint išlaikyti bei išanalizuoti ilgalaikes priklausomybes tarp duomenų



4 pav.: Didžiausių prognozuotų ir tikrųjų reikšmių pagal GAM modelį palyginimas

žingsnių. Šiuo modeliu buvo prognozuoti visi 6 savivaldybių subindeksai 2022 metams. Iš subindeksų prognozių, padaugintų pagal atitinkamus svorius, sumos buvo gautas GKI ir prognozės buvo atliekamos vieneriems metams į priekį. Prognozėms atlikti su LSTM modeliu, buvo naudojama vieno žingsnio prognozė (1-step ahead forecast) 2022 metų reikšmėms gauti. Kai kurios duomenų reikšmės iš ankstesnių metų buvo pašalintos dėl nustatyto laiko žingsnių skaičiaus. Kaip ir buvo minėta anksčiau, šiame modelyje buvo naudojami 3 laiko žingsniai, reiškiantys, kad kiekvienai prognozei sudaryti, modelis naudodavo tris ankstesnių metų duomenų reikšmes kaip įvestį. Norėdamas prognozuoti 2022 metų reikšmę, modelis naudojo tik 2019, 2020 ir 2021 metų duomenis. Kadangi kai kurie duomenys buvo diferencijuoti ir normalizuoti, gavus jų prognozes, reikėjo paversti jas į originalias prognozuotas reikšmes. Prognozuotos reikšmės, kurios buvo normalizuotos naudojant *MinMaxScaler* funkciją, buvo konvertuotos atgal į originalų mastelį, taikant atvirkštinę transformaciją tuo pačiu metodu. Tam buvo naudojama funkcija *scaler.inverse_transform()*, kuri grąžino reikšmes iš normalizuoto intervalo $[0; 1]$ į pradines reikšmes. Jei buvo atliktas diferencijavimas, prognozuotai reikšmei atstatyti reikėjo prie jos pridėti paskutinę originalios laiko eilutės reikšmę. Pavyzdžiui, prognozuojant 2022 metus, pridėjome prognozuotą skirtumą prie paskutinės originalios 2021 metų reikšmės. Antro lygio diferencijavimas su šiuo modeliu nebuvo taikomas. Šis procesas užtikrino, kad galutines prognozes būtų galima interpretuoti, kaip realius duomenis.

Savivaldybė	Metai	Prognozė	95% P.I.
A Subindeksas			
Kalvarijos sav.	2022	0,7272	[0,7184; 0,7354]
Kelmės r. sav.	2022	0,3786	[0,3454; 0,4058]
Lazdijų r. sav.	2022	0,6713	[0,6650; 0,6786]
Skuodo r. sav.	2022	0,2990	[0,2502; 0,3440]
Vilkaviškio r. sav.	2022	0,7854	[0,7827; 0,7872]
B Subindeksas			
Kalvarijos sav.	2022	0,1356	[0,1345; 0,1367]
Kelmės r. sav.	2022	0,0884	[0,0864; 0,0908]
Lazdijų r. sav.	2022	0,0462	[0,0437; 0,0484]
Skuodo r. sav.	2022	0,0210	[0,0181; 0,0234]
Vilkaviškio r. sav.	2022	0,0942	[0,0930; 0,0955]
C Subindeksas			
Kalvarijos sav.	2022	0,3583	[0,3510; 0,3638]
Kelmės r. sav.	2022	0,4341	[0,4294; 0,4389]
Lazdijų r. sav.	2022	0,3893	[0,3861; 0,3927]
Skuodo r. sav.	2022	0,3096	[0,3042; 0,3159]
Vilkaviškio r. sav.	2022	0,3621	[0,3587; 0,3662]
D Subindeksas			
Kalvarijos sav.	2022	0,2872	[0,2819; 0,2919]
Kelmės r. sav.	2022	0,3516	[0,3459; 0,3566]
Lazdijų r. sav.	2022	0,2884	[0,2763; 0,3020]
Skuodo r. sav.	2022	0,4866	[0,4824; 0,4900]
Vilkaviškio r. sav.	2022	0,3403	[0,3385; 0,3417]
E Subindeksas			
Kalvarijos sav.	2022	0,3830	[0,3797; 0,3855]
Kelmės r. sav.	2022	0,1669	[0,1583; 0,1743]
Lazdijų r. sav.	2022	0,3252	[0,3163; 0,3344]
Skuodo r. sav.	2022	0,1314	[0,1211; 0,1391]
Vilkaviškio r. sav.	2022	0,2718	[0,2651; 0,2783]
F Subindeksas			
Kalvarijos sav.	2022	0,4459	[0,4420; 0,4491]
Kelmės r. sav.	2022	0,5138	[0,5091; 0,5182]
Lazdijų r. sav.	2022	0,5361	[0,5347; 0,5372]
Skuodo r. sav.	2022	0,5696	[0,5693; 0,5699]
Vilkaviškio r. sav.	2022	0,4969	[0,4954; 0,4980]

21 lentelė: LSTM modelių prognozės pagal mažiausią GKI turinčias savivaldybes ir jų 95% pasiklivimo intervalai

Čia:

95% P.I. – 95% Monte Karlo pasiklivimo intervalai (2.21).

LSTM modelio prognozių rezultatai, pateikti 21 lentelėje, atspindi savivaldybių, pagal visus subindeksus, prognozių reikšmes bei jų pasiklivimo intervalus. Iš pateiktų duomenų

matome, jog visos savivaldybių prognozės patenka į 95% pasiklivimo intervalus, tad galime daryti prielaidą, jog subindeksų prognozės gavosi pakankamai tikslios. LSTM modelio pasiklivimo intervalų gauti nepavyko, kadangi šis modelis nėra pritaikytas pasikliautiniams intervalams skaičiuoti. Šio modelio tikslumą geriau vertinti per jo prognozių pasiklivimo intervalus, kurie tiesiogiai atspindi prognozių patikimumą. Tam buvo pasitelktas Monte Karlo metodas ir šiuo metodu prognozių pasiklivimo intervalai buvo apskaičiuoti atliekant daugkartines LSTM modelio prognozes su įvesties duomenimis. Kiekvienam subindeksui prognozės buvo atliekamos kelis kartus, kiekvieną kartą prie įvesties duomenų pridedant nedidelį atsitiktinį triukšmą, siekiant įvertinti modelio prognozių neapibrėžtumą. Gavus subindeksų prognozių imtį, buvo apskaičiuotas imties prognozių vidurkis ir standartinis nuokrypis bei tuomet buvo apskaičiuoti pasiklivimo intervalai kiekvienam iš subindeksų, kiekvienai savivaldybei.

Savivaldybė	Metai	Prognozė	95% P.I.
A Subindeksas			
Birštono sav.	2022	0,9545	[0,9427; 0,9662]
Kauno m. sav.	2022	1,2269	[1,2241; 1,2297]
Kauno r. sav.	2022	1,2434	[1,2274; 1,2593]
Klaipėdos m. sav.	2022	0,6145	[0,5508; 0,6781]
Vilniaus m. sav.	2022	1,3794	[1,3754; 1,3834]
B Subindeksas			
Birštono sav.	2022	0,3518	[0,3455; 0,3581]
Kauno m. sav.	2022	0,7269	[0,7203; 0,7335]
Kauno r. sav.	2022	0,6976	[0,6877; 0,7076]
Klaipėdos m. sav.	2022	0,7517	[0,7297; 0,7736]
Vilniaus m. sav.	2022	1,6008	[1,5896; 1,6119]
C Subindeksas			
Birštono sav.	2022	0,2931	[0,2839; 0,3023]
Kauno m. sav.	2022	0,7893	[0,7820; 0,7967]
Kauno r. sav.	2022	0,3546	[0,3476; 0,3616]
Klaipėdos m. sav.	2022	0,6969	[0,6879; 0,7060]
Vilniaus m. sav.	2022	0,6852	[0,6739; 0,6965]
D Subindeksas			
Birštono sav.	2022	0,4917	[0,4911; 0,4923]
Kauno m. sav.	2022	0,6156	[0,6075; 0,6237]
Kauno r. sav.	2022	0,4180	[0,4145; 0,4215]
Klaipėdos m. sav.	2022	0,5241	[0,5182; 0,5301]
Vilniaus m. sav.	2022	0,5198	[0,4863; 0,5534]
E Subindeksas			
Birštono sav.	2022	0,4168	[0,4082; 0,4254]
Kauno m. sav.	2022	0,3173	[0,2969; 0,3376]
Kauno r. sav.	2022	0,6621	[0,6593; 0,6648]
Klaipėdos m. sav.	2022	0,3244	[0,3139; 0,3349]
Vilniaus m. sav.	2022	0,4752	[0,4677; 0,4826]
F Subindeksas			
Birštono sav.	2022	0,7901	[0,7855; 0,7946]
Kauno m. sav.	2022	0,5146	[0,5066; 0,5226]
Kauno r. sav.	2022	0,5094	[0,5012; 0,5177]
Klaipėdos m. sav.	2022	0,5811	[0,5731; 0,5891]
Vilniaus m. sav.	2022	0,5897	[0,5885; 0,5909]

22 lentelė: LSTM modelių prognozės pagal didžiausią GKI turinčias savivaldybes ir jų 95% pasiklovimo intervalai

Čia:

95% P.I. – 95% Monte Karlo pasiklovimo intervalai (2.21).

LSTM modelių prognozių pagal didžiausią GKI turinčias savivaldybes lentelėje 22, taip pat matome kiekvieno subindekso prognozes savivaldybėms pagal didžiausią GKI bei pasiklo-

vimo intervalus. Pagal didžiausią GKI turinčias savivaldybes, LSTM modelio prognozės taip pat atitiko 95% pasiklivimo intervalų ribas, tad galime daryti prielaidą, jog subindeksų prognozės gavosi pakankamai tikslios. Šiuo atveju LSTM modelio pasiklivimo intervalų gauti nepavyko, tačiau, tuo pačiu, prieš tai minėtuojų Monte Karlo metodu, buvo gautos subindeksų prognozių pasiklivimo intervalų reikšmės.

	LSTM_MIN	LSTM_MAX
RMSE	0,131017793	0,144688622
MSE	0,017165662	0,020934797
MAPE	18,7228421%	13,5503493%

23 lentelė: LSTM modelio metrikų rezultatai

Čia:

RMSE – Šaknis iš vidutinės kvadratinės paklaidos (2.24).

MSE – Vidutinis kvadratinis nuokrypis (2.25).

MAPE – Vidutinė absoliutinė procentinė paklaida (2.26).

LSTM modelio metrikų rezultatuose, pateiktuose 23 lentelėje, matome, jog LSTM modelis geriau prognozavo savivaldybes, pasirinktas pagal žemiausią GKI, tačiau metrikos atrodo panašiai ir skiriasi nedaug. Matome, jog pagal žemiausią GKI šaknis iš vidutinės kvadratinės paklaidos reikšmė buvo 0,13, vidutinis kvadratinis nuokrypis buvo tik 0,017, o vidutinė absoliutinė procentinė paklaida buvo 18,7%.

Savivaldybės	Prognozuotas GKI	Tikras GKI	95% P.I.
Kalvarijos sav.	0,431114	0,4537610	[0,4306; 0,4370]
Kelmės r. sav.	0,295376	0,4760570	[0,3025; 0,3198]
Lazdijų r. sav.	0,410607	0,4734530	[0,4046; 0,4110]
Skuodo r. sav.	0,299064	0,4527395	[0,2637; 0,2919]
Vilkaviškio r. sav.	0,438736	0,4648895	[0,4386; 0,4412]

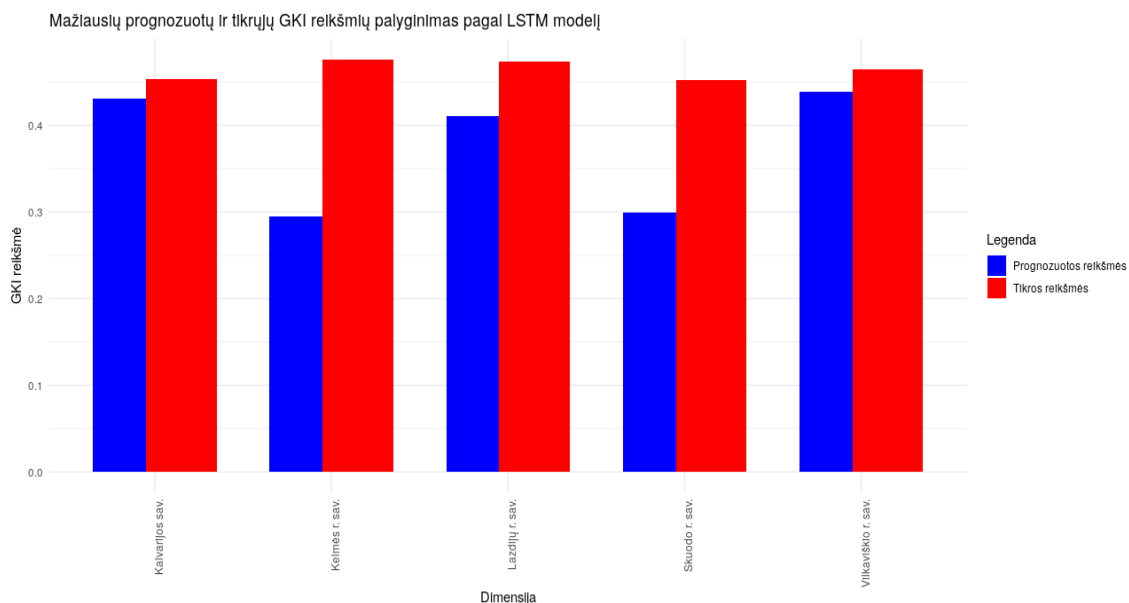
24 lentelė: Mažiausių prognozuotų ir tikrų reikšmių pagal LSTM modelį palyginimas

Čia:

95% P.I. – Monte Karlo simuliacijomis gautas GKI 95% pasiklivimo intervalas; žr. 2.2.9 skyrelį.

Iš mažiausių prognozuotų ir tikrų reikšmių pagal LSTM modelį palyginimų, pavaizduotų 24 lentelėje, matome, jog pagal prognozuotas ir tikras reikšmes savivaldybių išsiriavimas labai skiriasi. Pagal prognozuotas reikšmes, nuo žemiausią GKI turinčios savivaldybės, išsidėstymas atrodo taip: Kelmės r. sav., Skuodo r. sav., Lazdijų r. sav., Kalvarijos sav. ir Vilkaviškio r. sav. O pagal tikras reikšmes atrodo taip: Skuodo r. sav., Kalvarijos sav., Vilkaviškio r. sav., Lazdijų r. sav. ir Kelmės r. sav. GKI reikšmių skirtumus galima palyginti ir 5 paveiksle. Kelmės bei Skuodo rajonų savivaldybių prognozuotos GKI reikšmės neatitiko 95% pasiklivimo intervalų ribų, tad galimai šių savivaldybių prognozės nėra iki galo tikslios.

Iš didžiausių prognozuotų ir tikrų reikšmių pagal LSTM modelį palyginimų, pateiktų 25 lentelėje, matome, jog pagal prognozuotas ir tikras reikšmes, pagal aukščiausią GKI, savivaldybių išsidėstymas atrodo taip: Vilniaus m. sav., Kauno m. sav., Kauno r. sav., Birštono sav. bei Klaipėdos m. sav. O pagal tikras reikšmes atrodo taip: Vilniaus m. sav., Kauno m. sav., Klaipėdos m. sav., Kauno r. sav. bei Birštono sav. Taigi, pagal išsidėstymą nuo didžiausią GKI turinčių savivaldybių, sutapo Vilniaus ir Kauno miestų



5 pav.: Mažiausių prognozuotų ir tikrų reikšmių pagal LSTM modelį palyginimas

Savivaldybės	Prognozuotas GKI	Tikras GKI	95% P.I.
Birštono sav.	0,611588	0,7394860	[0,6110; 0,6201]
Kauno m. sav.	0,778940	0,7960150	[0,7749; 0,7827]
Kauno r. sav.	0,767285	0,7740490	[0,7600; 0,7715]
Klaipėdos m. sav.	0,579085	0,7774145	[0,5734; 0,6131]
Vilniaus m. sav.	1,009772	1,00141005	[1,0091; 1,0192]

25 lentelė: Didžiausių prognozuotų ir tikrų reikšmių pagal LSTM modelį palyginimas

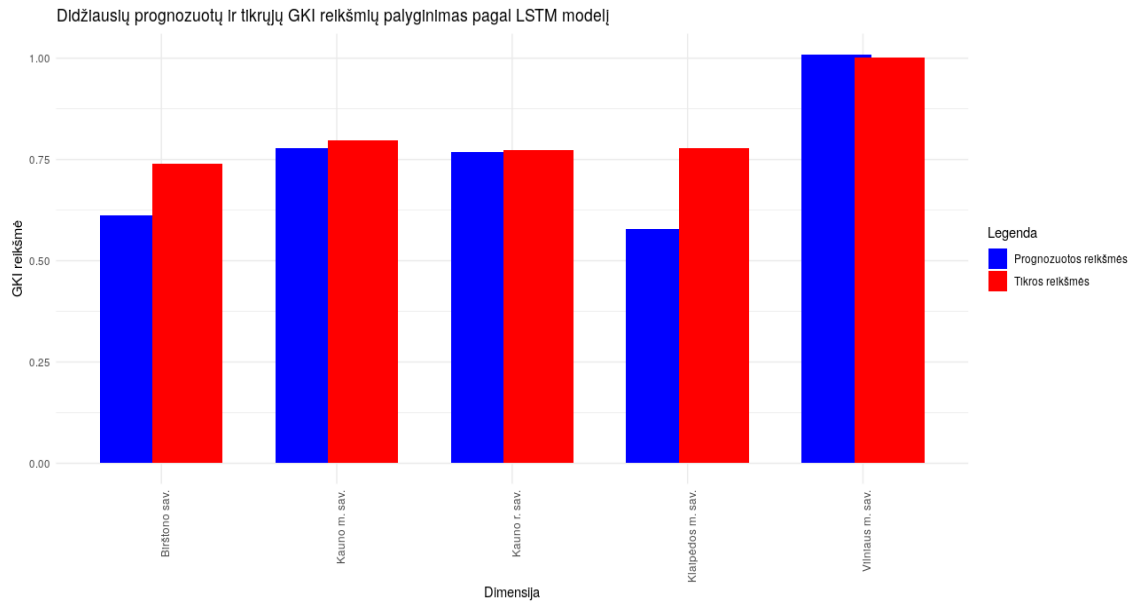
Čia:

95% P.I. – Monte Karlo simuliacijomis gautas GKI 95% pasiklivimo intervalas; žr. 2.2.9 skyrelį.

savivaldybės. Pagal LSTM modelį prognozuoti GKI likusioms 5 savivaldybėms atitiko 95% pasiklivimo intervalų ribas. Tad galime teigti, jog šių savivaldybių prognozės LSTM modeliu gavosi pakankamai tikslios. Tikrų ir prognozuotų GKI reikšmių skirtumus galima pamatyti ir 6 paveiksle.

3.3.4 BiLSTM modelio prognozės

Ketvirtasis modelis, kuriuo buvo atliekamas savivaldybių subindeksų prognozavimas, buvo Dvikryptis ilgalaikės trumpalaikės atminties (BiLSTM) modelis. Šis modelis yra geresnė LSTM modelio versija. Prognozavimas BiLSTM modeliu buvo atliktas visų 10-ies savivaldybių subindeksų reikšmėms, kurių iš viso buvo 6. Gavus subindeksų prognozes, jos buvo padaugintos iš savo svorių bei sudėtos, ir taip buvo gautos savivaldybių Gyvenimo kokybės indekso prognozės metams į priekį. BiLSTM modelio prognozės buvo atliktos beveik identiška, kaip ir su LSTM modeliu. Buvo taikyta vieno žingsnio prognozė (1-step ahead forecast) bei naudojami 3 laiko žingsniai, kurie padėjo gauti 2022 metų prognozių reikšmes. Prognozavime naudoti trys laiko žingsniai reiškė, jog modelis naudodavo trejų ankstesnių metų duomenų reikšmes kaip įvestį, kiekvienai prognozei gauti. Norint prognozuoti 2022



6 pav.: Didžiausių prognozuotų ir tikrųjų reikšmių pagal LSTM modelį palyginimas

metų subindeksų reikšmės, modelis naudojo tik 2019, 2020 ir 2021 metų duomenis. Prieš sudarant BiLSTM modelį, duomenys buvo normalizuoti ir kai kurie iš jų diferencijuoti, tad gavus prognozes, reikėjo jas paversti į originalias prognozuotas reikšmes. Prognozių reikšmės, kurioms buvo taikytas normalizavimas, naudojant *MinMaxScaler* funkciją, buvo konvertuotos atgal į originalų mastelį, taikant atvirkštinę transformaciją tuo pačiu metodu. Šiai konvertacijai buvo naudojama *scaler.inverse_transform()* funkcija, kuri grąžino reikšmes iš normalizuotos skalės intervale $[0;1]$ į pradines reikšmes. Jei duomenims buvo atliktas diferencijavimas, prognozuotai originaliai reikšmei atstatyti reikėjo prie jos pridėti paskutinę originalios laiko eilutės reikšmę – prognozuojant 2022 metus, pridėjome prognozuotą skirtumą prie paskutinės originalios 2021 metų reikšmės. Šiame modelyje 2 lygio diferencijavimas nebuvo atliktas. Šios 2 transformacijos užtikrino, jog galutinės prognozės bus originalios.

Savivaldybė	Metai	Prognozė	95% P.I.
A Subindeksas			
Kalvarijos sav.	2022	0,7134	[0,7066; 0,7187]
Kelmės r. sav.	2022	0,3112	[0,2799; 0,3404]
Lazdijų r. sav.	2022	0,6808	[0,6725; 0,6881]
Skuodo r. sav.	2022	0,3689	[0,3256; 0,4007]
Vilkaviškio r. sav.	2022	0,7904	[0,7871; 0,7938]
B Subindeksas			
Kalvarijos sav.	2022	0,1353	[0,1342; 0,1364]
Kelmės r. sav.	2022	0,0898	[0,0879; 0,0918]
Lazdijų r. sav.	2022	0,0486	[0,0459; 0,0510]
Skuodo r. sav.	2022	0,0219	[0,0195; 0,0245]
Vilkaviškio r. sav.	2022	0,0941	[0,0931; 0,0954]
C Subindeksas			
Kalvarijos sav.	2022	0,3619	[0,3559; 0,3668]
Kelmės r. sav.	2022	0,4529	[0,4466; 0,4571]
Lazdijų r. sav.	2022	0,3938	[0,3901; 0,3968]
Skuodo r. sav.	2022	0,3024	[0,2955; 0,3085]
Vilkaviškio r. sav.	2022	0,3705	[0,3651; 0,3762]
D Subindeksas			
Kalvarijos sav.	2022	0,2877	[0,2827; 0,2929]
Kelmės r. sav.	2022	0,3486	[0,3436; 0,3545]
Lazdijų r. sav.	2022	0,2971	[0,2848; 0,3067]
Skuodo r. sav.	2022	0,4806	[0,4771; 0,4833]
Vilkaviškio r. sav.	2022	0,3448	[0,3431; 0,3461]
E Subindeksas			
Kalvarijos sav.	2022	0,3822	[0,3793; 0,3855]
Kelmės r. sav.	2022	0,1766	[0,1690; 0,1831]
Lazdijų r. sav.	2022	0,3129	[0,3075; 0,3190]
Skuodo r. sav.	2022	0,1334	[0,1241; 0,1432]
Vilkaviškio r. sav.	2022	0,2718	[0,2667; 0,2760]
F Subindeksas			
Kalvarijos sav.	2022	0,4607	[0,4568; 0,4646]
Kelmės r. sav.	2022	0,5253	[0,5188; 0,5311]
Lazdijų r. sav.	2022	0,5367	[0,5354; 0,5386]
Skuodo r. sav.	2022	0,5705	[0,5701; 0,5709]
Vilkaviškio r. sav.	2022	0,5010	[0,4992; 0,5027]

26 lentelė: BiLSTM modelių prognozės pagal mažiausią GKI turinčias savivaldybes ir jų 95% pasiklivimo intervalai

Čia:

95% P.I. – 95% Monte Karlo pasiklivimo intervalai (2.21).

BiLSTM modelių prognozės pagal mažiausią GKI turinčias savivaldybes, pateiktos 26 lentelėje, atskleidė, jog visos prognozės patenka į 95% pasiklivimo intervalų ribas, tad taip

pat galime daryti prielaidą, jog prognozės yra gan tikslios. BiLSTM modelio pasiklivimo intervalų gauti nebuvo galimybės, kadangi šis modelis taip pat nėra pritaikytas jiems skaičiuoti ir šio modelio prognozių tikslumas buvo vertinamas pagal prognozių pasiklivimo intervalus. Prognozių pasiklivimo intervalai buvo skaičiuojami pasitelkiant prieš tai aprašytą Monte Karlo metodą, kuomet turėdami subindeksų prognozių imtį apskaičiavome jų vidurkį, standartinę nuokrypį bei gavome pasiklivimo intervalus.

Savivaldybė	Metai	Prognozė	95% P.I.
A Subindeksas			
Birštono sav.	2022	0,9660	[0,9353; 0,9968]
Kauno m. sav.	2022	1,2290	[1,2265; 1,2315]
Kauno r. sav.	2022	1,2437	[1,2323; 1,2551]
Klaipėdos m. sav.	2022	0,6324	[0,5895; 0,6752]
Vilniaus m. sav.	2022	1,3814	[1,3751; 1,3877]
B Subindeksas			
Birštono sav.	2022	0,3543	[0,3534; 0,3552]
Kauno m. sav.	2022	0,7315	[0,7211; 0,7419]
Kauno r. sav.	2022	0,7057	[0,6992; 0,7122]
Klaipėdos m. sav.	2022	0,7627	[0,7378; 0,7877]
Vilniaus m. sav.	2022	1,6054	[1,5917; 1,6191]
C Subindeksas			
Birštono sav.	2022	0,3013	[0,2802; 0,3224]
Kauno m. sav.	2022	0,7928	[0,7862; 0,7995]
Kauno r. sav.	2022	0,3643	[0,3510; 0,3776]
Klaipėdos m. sav.	2022	0,7019	[0,6927; 0,7111]
Vilniaus m. sav.	2022	0,6860	[0,6792; 0,6928]
D Subindeksas			
Birštono sav.	2022	0,5009	[0,4887; 0,5132]
Kauno m. sav.	2022	0,6136	[0,6056; 0,6215]
Kauno r. sav.	2022	0,4193	[0,4169; 0,4216]
Klaipėdos m. sav.	2022	0,5264	[0,5221; 0,5307]
Vilniaus m. sav.	2022	0,5258	[0,4958; 0,5558]
E Subindeksas			
Birštono sav.	2022	0,4193	[0,3998; 0,4387]
Kauno m. sav.	2022	0,3205	[0,3072; 0,3337]
Kauno r. sav.	2022	0,6659	[0,6573; 0,6745]
Klaipėdos m. sav.	2022	0,3307	[0,3290; 0,3324]
Vilniaus m. sav.	2022	0,4755	[0,4654; 0,4856]
F Subindeksas			
Birštono sav.	2022	0,7958	[0,7884; 0,8032]
Kauno m. sav.	2022	0,5138	[0,5052; 0,5224]
Kauno r. sav.	2022	0,5102	[0,5038; 0,5166]
Klaipėdos m. sav.	2022	0,5845	[0,5787; 0,5903]
Vilniaus m. sav.	2022	0,5905	[0,5881; 0,5929]

27 lentelė: BiLSTM modelių prognozės pagal didžiausią GKI turinčias savivaldybes ir jų 95% pasiklovimo intervalai

Čia:

95% P.I. – 95% Monte Karlo pasiklovimo intervalai (2.21).

Iš BiLSTM modelių prognozių pagal didžiausią GKI turinčias savivaldybes rezultatų, pateiktų 27 lentelėje, matome BiLSTM modelio prognozes subindeksams likusioms 5 savival-

dybėms bei jų pasiklivimo intervalus. Čia visos prognozės atitinka pasiklivimo intervalų ribas ir tai parodo, jog galima daryti prielaidą apie pakankamai tikslias modelio prognozes. Prognozių 95% pasiklivimo intervalai taip pat buvo apskaičiuoti pasitelkiant Monte Karlo metodą, gavus subindeksų prognozių imties vidurkį bei standartinį nuokrypį. Turint visų subindeksų prognozes savivaldybėms, buvo atliekamas Gyvenimo kokybės indekso prognozavimas, susumuojant bei padauginant visus subindeksus iš jų svorių.

	BiLSTM_MIN	BiLSTM_MAX
RMSE	0,126871418	0,13660098
MSE	0,016096357	0,018659828
MAPE	17,4617448%	12,7335764%

28 lentelė: BiLSTM modelio metrikų rezultatai

Čia:

RMSE – Šaknis iš vidutinės kvadratinės paklaidos (2.24).

MSE – Vidutinis kvadratinis nuokrypis (2.25).

MAPE – Vidutinė absoliutinė procentinė paklaida (2.26).

BiLSTM modelio metrikų rezultatuose, pateiktuose 28 lentelėje, matome, jog BiLSTM modelis geriau prognozavo savivaldybes, pasirinktas pagal žemiausią GKI, tačiau metrikos atrodo panašiai ir labiausiai skiriasi tik vidutinė absoliutinė procentinė paklaida. Matome, jog pagal žemiausią GKI šaknies iš vidutinės kvadratinės paklaidos reikšmė buvo 0,12, vidutinis kvadratinis nuokrypis buvo tik 0,016, o vidutinė absoliutinė procentinė paklaida buvo apytiksliai 17,5%.

Savivaldybės	Prognozuotas GKI	Tikras GKI	95% P.I.
Kalvarijos sav.	0,439237	0,4537610	[0,4299; 0,4341]
Kelmės r. sav.	0,315167	0,4760570	[0,2869; 0,3045]
Lazdijų r. sav.	0,413636	0,4734530	[0,4072; 0,4131]
Skuodo r. sav.	0,285841	0,4527395	[0,2862; 0,3070]
Vilkaviškio r. sav.	0,445429	0,4648895	[0,4418; 0,4450]

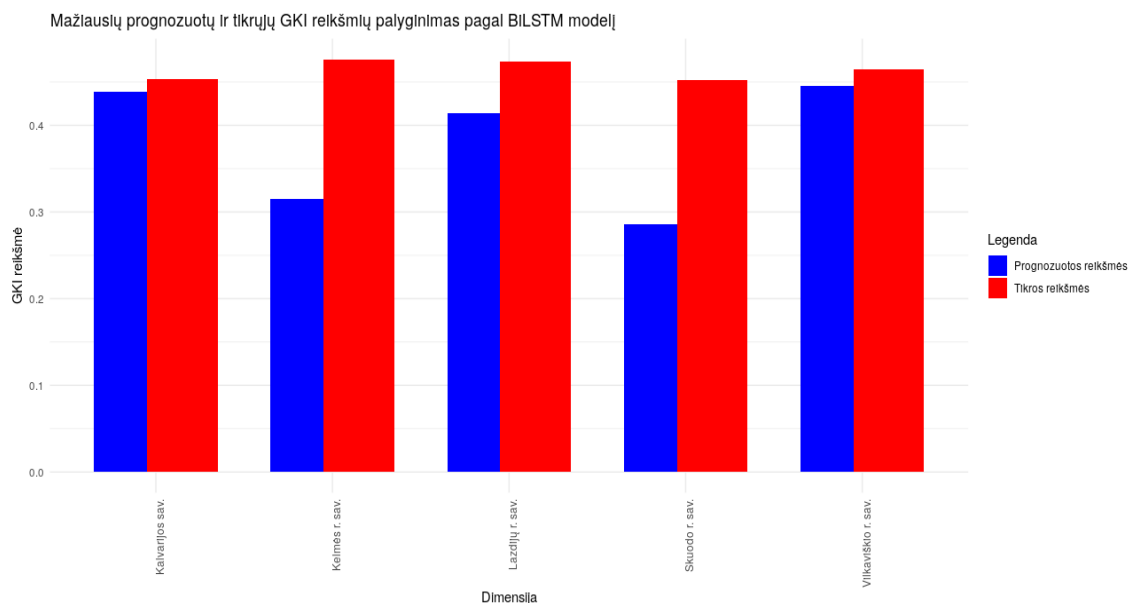
29 lentelė: Mažiausių prognozuotų ir tikrų reikšmių pagal BiLSTM modelį palyginimas

Čia:

95% P.I. – Monte Karlo simuliacijomis gautas GKI 95% pasiklivimo intervalas; žr. 2.2.9 skyrelį.

Šioje mažiausių prognozuotų ir tikrų reikšmių pagal BiLSTM modelį 29 lentelėje matome, jog pagal prognozuotas ir tikras reikšmes savivaldybių išsirikiavimas labai skiriasi. Pagal prognozuotas reikšmes, nuo žemiausią GKI turinčios savivaldybės, išsidėstymas atrodo taip: Skuodo r. sav., Kelmės r. sav., Lazdijų r. sav., Kalvarijos sav. ir Vilkaviškio r. sav. O pagal tikras reikšmes atrodo taip: Skuodo r. sav., Kalvarijos sav., Vilkaviškio r. sav., Lazdijų r. sav. ir Kelmės r. sav. Pagal tikrą ir prognozuotą savivaldybių išsidėstymą nuo mažiausią GKI turinčios savivaldybės sutapo tik vienintelė Skuodo r. sav. Šioje 29 lentelėje nei viena prognozuota GKI reikšmė neatitiko 95% pasiklivimo intervalų ribų. GKI reikšmių skirtumus galima palyginti ir 7 paveiksle.

Šioje didžiausių prognozuotų ir tikrų reikšmių pagal BiLSTM modelį 30 lentelėje matome, jog pagal prognozuotas ir tikras reikšmes, pagal aukščiausią GKI, savivaldybių išsidėstymas atrodo taip: Vilniaus m. sav., Kauno m. sav., Kauno r. sav., Birštono sav. bei Klaipėdos m. sav. O pagal tikras reikšmes atrodo taip: Vilniaus m. sav., Kauno m. sav., Klaipėdos



7 pav.: Mažiausių prognozuotų ir tikrųjų reikšmių pagal BiLSTM modelį palyginimas

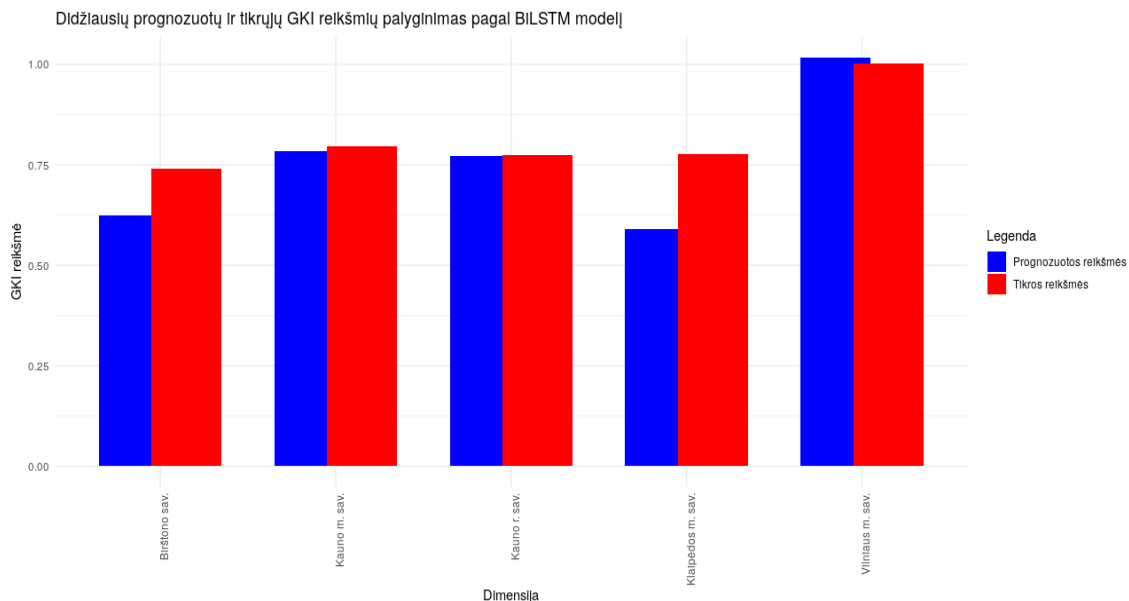
Savivaldybės	Prognozuotas GKI	Tikras GKI	95% P.I.
Birštono sav.	0,624114	0,7394860	[0,6126; 0,6326]
Kauno m. sav.	0,783171	0,7960150	[0,7772; 0,7849]
Kauno r. sav.	0,772244	0,7740490	[0,7646; 0,7739]
Klaipėdos m. sav.	0,590795	0,7774145	[0,5889; 0,6171]
Vilniaus m. sav.	1,017690	1,00141005	[1,0119; 1,0217]

30 lentelė: Didžiausių prognozuotų ir tikrųjų reikšmių pagal BiLSTM modelį palyginimas

Čia:

95% P.I. – Monte Karlo simuliacijomis gautas GKI 95% pasiklivimo intervalas; žr. 2.2.9 skyrelį.

m. sav., Kauno r. sav. bei Birštono sav. Taigi, pagal išsidėstymą nuo didžiausią GKI turinčių savivaldybių, sutapo Vilniaus ir Kauno miestų savivaldybės ir šis išsidėstymas yra identiškas, kaip ir su LSTM modeliu pagal didžiausias prognozuotas ir tikras GKI reikšmes. Čia taip pat matome, jog visos prognozės atitiko 95% pasiklivimo intervalų ribas, kurios buvo gautos Monte Karlo simuliacijų būdu. Tikrųjų ir prognozuotųjų GKI reikšmių skirtumus galima pamatyti ir 8 paveiksle.



8 pav.: Didžiausių prognozuotų ir tikrų reikšmių pagal BiLSTM modelį palyginimas

4 Išvados bei rekomendacijos

Šiame darbe buvo atliktas tyrimas apie 10-ies Lietuvos savivaldybių Gyvenimo kokybės indeksą ir jį sudarančius subindeksus. Buvo išanalizuotos penkios žemiausią ir penkios aukščiausią Gyvenimo kokybės indeksą Lietuvoje turinčios savivaldybės, sudaryti modeliai ir atliktos 2022 metų GKI prognozės, kurios buvo lyginamos su tikromis reikšmėmis. Modeliai buvo atrinkti pagal aktualią bei išnagrinėtą mokslinę literatūrą, laiko eilučių duomenims. Savivaldybės buvo parinktos pagal 2022 metų duomenis. Penkios savivaldybės pagal žemiausias GKI reikšmes (rikiuojant nuo žemiausios reikšmės) buvo: Skuodo rajono, Kalvarijos, Vilkaviškio rajono, Lazdijų rajono ir Kelmės rajono. Penkios savivaldybės pagal aukščiausias GKI reikšmes (rikiuojant nuo didžiausios reikšmės) buvo: Vilniaus miesto, Kauno miesto, Klaipėdos miesto, Kauno rajono bei Birštono. Modeliams sudaryti bei prognozėms atlikti buvo naudojami 2013 – 2021 metų paneliniai Lietuvos savivaldybių subindeksų duomenys. Sudėjus visų subindeksų prognozes, kurias reikėjo dar papildomai padauginti iš jų svorių, buvo gautas Gyvenimo kokybės indeksas. 2022 metų duomenys tiek subindeksų, tiek GKI buvo naudojami prognozuotų ir tikrų reikšmių palyginimams. Iš viso buvo sudaryti 4 laiko eilučių modeliai, tokie kaip Autoregresinis integruotas slenkančio vidurkio modelis (ARIMA), Apibendrintasis adityvusis modelis (GAM), Ilgalaikės trumpalaikės atminties modelis (LSTM) bei patobulinta šio modelio versija – Dvikryptis Ilgalaikės trumpalaikės atminties modelis (BiLSTM). Kiekvieno subindekso duomenų rinkiniai buvo transformuojami į panelinę struktūrą, tikrinamos duomenų prielaidos, sudarinėjami modeliai, tuomet tikrinamos tam tikros modelių prielaidos. Daug kur buvo susiduriama su duomenų nestacionarumo problema, todėl buvo atliktas diferencijavimas ir imti kintamųjų skirtumai laike. Darbe buvo aptarti visų modelių tam tikrų koeficientų statistiniai reikšmingumai kiekvienam iš subindeksų. Sekantis žingsnis buvo subindeksų prognozavimas visoms savivaldybėms ir šios prognozės buvo tikrinamos pagal 95% pasiklivimo intervalų ribas. Kadangi visos prognozės atitiko pasiklivimo intervalus, buvo

galima daryti prielaidą, jog jos yra tikslios ir tuomet jau buvo atliktas Gyvenimo kokybės indekso prognozavimas savivaldybėms. Visos subindeksų prognozės buvo padaugintos iš savo atitinkamų svorių bei susumavus ir buvo gautas GKI pagal keturis skirtingus laiko eilučių modelius. Čia buvo pastebėta, jog vieni modeliai tiksliau prognozavo GKI reikšmes, o kiti prasčiau. GKI prognozuotų reikšmių tikslumui vertinti buvo pasitelktas Monte Karlo simuliacijų metodas. ARIMA modeliu prognozuotos GKI reikšmės visur atitiko 95% pasiklivimo intervalų ribas. GAM modelyje pasiklivimo intervalų ribas atitiko tik pagal mažiausią GKI turinčių savivaldybių prognozuotos reikšmės. Tuomet pagal LSTM modelį prognozuotos GKI reikšmės didžiaja dalimi atitiko pasiklivimo intervalus – Kelmės bei Skuodo rajonų savivaldybių GKI prognozės nepateko į 95% pasiklivimo intervalų ribą. O pagal paskutinįjį taikytą BiLSTM modelį, pasiklivimo intervalų ribas atitiko tik tos GKI reikšmės, kurios buvo pagal didžiausią GKI turinčias savivaldybes. Tad galime teigti, jog ARIMA modeliu gautos prognozės buvo labiausiai tikslios, taip pat ir LSTM modeliu gautos prognozės didžiaja dalimi irgi buvo tikslios. GAM ir BiLSTM modeliais gautos GKI prognozės turėjo tik dalinį tikslumą. Monte Karlo simuliacijų metodui prognozuotų reikšmių pasiklivimo intervalams buvo pasirinktas 95% lygmuo, kadangi tai standartinis bei dažniausiai moksliniuose tyrimuose praktikoje taikomas lygmuo. Likusių 5% ekstremumų reikšmėms analizė nebuvo atlikta, kadangi tai yra sudėtingesnės analizės reikalaujantis uždavinys ir, to pasekoje, darbo analizėje gauti rezultatai galėjo būti ne iki galo tikslūs. Taip pat tikslumui įvertinti buvo naudojamos 3 metrikos: Šaknis iš vidutinės kvadratinės paklaidos, Vidutinis kvadratinis nuokrypis ir Vidutinė absoliutinė procentinė paklaida. Atsižvelgiant į metrikų rezultatus, buvo pastebėta, jog geriausiai prognozavo Autoregresinis integruotas slenkančio vidurkio modelis (ARIMA) bei Dvikryptis Ilgalaikės trumpalaikės atminties (BiLSTM) modelis. Šių modelių metrikų rezultatai buvo mažiausi. Apibendrintasis adityvusis modelis (GAM) bei Ilgalaikės trumpalaikės atminties (LSTM) modelis prognozavo kur kas prasčiau, nes jų metrikos turėjo aukščiausias reikšmes, o tai rodo prastą prognozių atitikimą realioms duomenims. Taip pat tikrų ir prognozuotų reikšmių, pagal mažiausią GKI turinčias savivaldybes, metrikų rezultatai beveik visur buvo mažesni. Taigi, ARIMA modelio prognozės tiek pagal pasiklivimo intervalus, tiek pagal metrikų rezultatus buvo tiksliausios. Lyginant po 5 mažiausias ir 5 didžiausias GKI reikšmes turinčias savivaldybes pagal tikrus ir prognozuotus 2022 metų duomenis – atitikimų pagal savivaldybių išsiriavimą buvo nedaug, nors reikšmės labai stipriai nesiskyrė. Prognozuotos ir tikros Vilniaus bei Kauno miestų savivaldybių GKI reikšmės pagal išsidėstymą sutapo ARIMA, LSTM bei BiLSTM modeliuose. Birštono savivaldybė sutapo ARIMA ir GAM modeliuose. Pastarajame GAM modelyje irgi sutapo Vilniaus miesto savivaldybė ir Kauno rajono savivaldybė. Taip pat ARIMA bei GAM modeliuose, pagal prognozuotų ir tikrų mažiausią GKI turinčių savivaldybių išsidėstymą, sutapo Kelmės rajono savivaldybė. BiLSTM modelyje sutapo Skuodo rajono savivaldybė pagal tikrų ir prognozuotų savivaldybių GKI reikšmių išsidėstymą. Taigi, apibendrinus galima teigti, jog prognozuotos 2022 metų GKI reikšmės ne iki galo atitiko originalius to meto duomenis. Prognozuotų subindeksų reikšmės tobulai neatitiko realių jų reikšmių ir, to pasekoje, gavome ne iki galo realias duomenų reikšmes atitinkantį Gyvenimo kokybės indeksą, kuris buvo gautas iš subindeksų, padaugintų pagal atitinkamus savo svorius, sumos. Iš darbe pateiktų lentelių matome, jog prognozuotos ir tikros reikšmės skyrėsi ir ne iki galo atitiko savivaldybių išsiriavimą. Reikšmių neatitikimams daug įtakos galėjo turėti modeliavimo rezultatai, trumpa laiko eilutė, ne iki galo tinkamai parinkti modeliai.

Rekomendacijos tolimesniems bei panašios tematikos tyrimams būtų tokios, jog reikėtų

analizuoti ilgesnę laiko eilutę, kadangi tuomet galimai prognozės labiau atitiktų realius duomenis. Taip pat būtų naudinga išmėginti ir kitus modelius bei kuo labiau įsigilinti į jų struktūrą, prieš sudarinėjant prognozes. Rekomenduotina į analizę įtraukti kuo daugiau subindeksus sudarančių komponentų, nes tai gali pagerinti modelių charakteristikas ir prognozavimą. Kadangi Monte Karlo simuliacijų metodo metu pasiklovimo intervalų lygmuo buvo pasirinktas 95%, likusios 5% ekstremumų reikšmės liko neišnagrinėtos. Taigi, dar viena rekomendacija panašios tematikos tyrimams būtų tokia, jog reikėtų išnagrinėti ir 5% ekstremumų poveikį pasiklovimo intervalams bei prognozių tikslumui. Ši analizė yra svarbi modeliuojant ir interpretuojant kiekvieno subindekso reikšmes, kurios nulemia ir paties Gyvenimo kokybės indekso prognozuotą reikšmę bei jos patikimumą. Taip pat tokia analizė padėtų atskleisti galimus modelių taikymo intervalus, kuomet laiko eilutė yra pakankamai trumpa.

Lietuvos savivaldybės pasigerinti ir/arba išlaikyti savo Gyvenimo kokybės indeksą gali keliais būdais. Savivaldybės galėtų skatinti ekonominę plėtrą per mokesčines lengvatas smulkiajam ir vidutiniam verslui. Tai taip pat gal net padėtų pritraukti naujų investicijų į tam tikras savivaldybes. Smulkusis bei vidutinis klestintis verslas pritrauktų ir naujų turistų, kurie taip pat prisidėtų prie savivaldybių ekonomikos gerinimo. Kita svarbi sritis yra švietimas – savivaldybės turėtų stengtis išlaikyti kuo didesnę mokyklų, darželių ir kitų neformaliojo ugdymo įstaigų skaičių, norint, jog visi šalies gyventojai gautų kokybišką išsilavinimą, nepriklausomai nuo savo gyvenamosios aplinkos. Sveikatos paslaugų prieinamumas – taip pat svarbi sritis, į kurią savivaldybėms reikėtų susitelkti, norint išlaikyti ne tik aukštą šių paslaugų lygį, bet ir sukurti palankias darbo sąlygas sveikatos specialistams, kuriuos vis sunkiau pritraukti dirbti į mažuosius šalies miestus bei savivaldybes. Norint pritraukti jaunos bei profesionalios sveikatos paslaugų specialistus, reikia stengtis kurti palankias darbo sąlygas bei gerinti kiekvienos savivaldybės infrastruktūrą, pavyzdžiui, skatinti žaliųjų erdvių didinimą bei didesnę subsidijų kiekį būstui įsigyti. Infrastruktūros gerinimas prisidėtų ne tik prie geresnių sąlygų savivaldybių gyventojams, bet ir didintų kiekvienos iš jų demografinį lygmenį.

Literatūra

- [1] S. Jackson, *Machine Learning*, Durham University, England, 2024, 10.1 ir 10.2 skyriai, nuoroda tikrinta 2024 m. gruodžio 5 d. <https://bookdown.org/ssjackson300/Machine-Learning-Lecture-Notes/generalised-additive-models.html>
- [2] Savivaldybių Gyvenimo kokybės indeksas, nuoroda tikrinta 2024 m. gruodžio 5 d. <https://lietuvosfinansai.lt/gki/gki-struktura/>
- [3] S. Siami-Namini, N. Tavakoli, A. Siami Namin, *A Comparison of ARIMA and LSTM in Forecasting Time Series*, 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 2018, 1394–1401, <https://ieeexplore.ieee.org/document/8614252>
- [4] Peter T. Yamak, Li Yujian, Pius K. Gadosey, *A Comparison between ARIMA, LSTM, and GRU for Time Series Forecasting*, ACAI'19: Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence 49–55, <https://dl.acm.org/doi/10.1145/3377713.3377722>
- [5] Abdelhafid Zeroual, Fouzi Harrou, Abdelkader Dairi, Ying Sun *Deep learning methods for forecasting COVID-19 time-Series data: A Comparative study*, Chaos, Solitons & Fractals, vol. 140, Nov. 2020, <https://www.sciencedirect.com/science/article/pii/S096007792030518X>
- [6] Raúl López-Blanco, Juan Herranz Martín, Ricardo S. Alonso ir Javier Prieto *Time Series Forecasting for Improving Quality of Life and Ecosystem Services in Smart Cities*, Ambient Intelligence—Software and Applications—13th International Symposium on Ambient Intelligence, Springer International Publishing, Cham, 2023, 74–85 https://link.springer.com/chapter/10.1007/978-3-031-22356-3_8
- [7] A. Račkauskas, *Atsitiktiniai procesai*, Paskaitų konspektas, 2014, 121 p., <https://web.vu.lt/mif/a.medziunas/wp-content/uploads/2021/02/Rackauskas-1.pdf>
- [8] Peter J. Brockwell, Richard A. Davis, *Introduction to Time Series and Forecasting*, 2016, 14 p., <https://link.springer.com/book/10.1007/978-3-319-29854-2>

5 Priedai

Šiame priede yra sudėti visi 4 modeliai su prognozėmis bei metrikų palyginimais. Kitų subindeksų koduose keitėsi tik atitinkami duomenų rinkiniai ir analizė buvo atliekama identiškai, tad priede pateiktas tik vieno subindekso analizės kodas. ARIMA ir GAM modeliai buvo rašyti su Rstudio programa. LSTM ir BiLSTM modeliai buvo rašyti su „Python“ programa. Paskutinis kodas yra Monte Karlo simuliacijų taikymas 95% pasiklovimo intervalams gauti.

```
# Load necessary libraries
library(forecast)
library(readxl)
library(dplyr)
library(writexl)
library(tseries)
library(lmtest)
library(FinTS)

# Load the data
A_MAX <- read_excel("/mif/stud3/2019/ugli6286/reshaped_A_MAX.xlsx")

# Convert columns to appropriate data types
A_MAX$Year <- as.numeric(A_MAX$Year)
A_MAX$Value <- as.numeric(A_MAX$Value)

A_MAX <- na.omit(A_MAX)

cat("Structure of the data:\n")
print(str(A_MAX))

unique_municipalities <- unique(A_MAX$Dimension)
cat("\nUnique municipalities found:\n")
print(unique_municipalities)

# Create an empty data frame to store predictions and diagnostics
predictions <- data.frame(
  Municipality = character(),
  Year = numeric(),
  Predicted_Value = numeric(),
  Lower_95_CI = numeric(),
  Upper_95_CI = numeric(),
  ADF_P_Value = numeric(),
  Jarque_Bera_P_Value = numeric(),
  Breusch_Pagan_P_Value = numeric(),
  ARCH_LM_P_Value = numeric(),
  stringsAsFactors = FALSE
)

# Loop through each municipality
for (municipality in unique_municipalities) {
```

```

cat("\n#####\n")
cat("Processing:", municipality, "\n")

municipality_data <- A_MAX %>% filter(Dimension == municipality)

# Ensure enough data points exist
if (nrow(municipality_data) < 3) {
  cat("Insufficient data for analysis. Skipping...\n")
  next
}

# Fit a linear trend model
trend_model <- lm(Value ~ Year, data = municipality_data)
bp_test <- bptest(trend_model)
bp_test_p_value <- bp_test$p.value

# Detrend the data
trend_values <- predict(trend_model, newdata = municipality_data)
detrended_data <- municipality_data$Value - trend_values
ts_data_orig <- ts(detrended_data, start = min(municipality_data$Year), frequency = 1)

adf_test_orig <- adf.test(ts_data_orig)
adf_test_p_value <- adf_test_orig$p.value

ts_data <- ts_data_orig
differencing_applied <- FALSE
if (adf_test_p_value > 0.05) {
  ts_data <- diff(ts_data_orig, differences = 1)
  differencing_applied <- TRUE
  adf_test_diff <- adf.test(ts_data)
  adf_test_p_value <- adf_test_diff$p.value
}

arima_model <- tryCatch({
  auto.arima(ts_data)
}, error = function(e) {
  cat("ARIMA model failed for", municipality, ": ", e$message, "\n")
  return(NULL)
})

if (is.null(arima_model)) next

# Forecasting with ARIMA
arima_forecast <- tryCatch({
  forecast_result <- forecast(arima_model, h = 1)
  list(mean = forecast_result$mean,
        lower = forecast_result$lower[, 2],
        upper = forecast_result$upper[, 2])
}, error = function(e) {
  cat("ARIMA prediction failed for", municipality, ": ", e$message, "\n")
  return(list(mean = 0, lower = 0, upper = 0))
})

# Restore forecasts to original scale if differencing was applied

```

```

if (differencing_applied) {
  last_value <- tail(municipality_data$Value, 1)
  predicted_value <- last_value + as.numeric(arima_forecast$mean)
  predicted_lower <- last_value + as.numeric(arima_forecast$lower)
  predicted_upper <- last_value + as.numeric(arima_forecast$upper)
} else {
  predicted_value <- as.numeric(arima_forecast$mean)
  predicted_lower <- as.numeric(arima_forecast$lower)
  predicted_upper <- as.numeric(arima_forecast$upper)
}

# Perform diagnostics
jb_test <- jarque.bera.test(residuals(arima_model))
jb_test_p_value <- jb_test$p.value

arch_test <- tryCatch({
  FinTS::ArchTest(residuals(arima_model))
}, error = function(e) {
  return(NULL)
})
arch_lm_p_value <- if (!is.null(arch_test)) arch_test$p.value else NA

# Plot ACF and PACF for residuals
cat("Plotting ACF and PACF for", municipality, "...\\n")
par(mfrow = c(1, 2)) # Arrange plots side-by-side
acf(residuals(arima_model), main = paste("ACF for", municipality))
pacf(residuals(arima_model), main = paste("PACF for", municipality))

# Append results to the predictions data frame
predictions <- rbind(predictions, data.frame(
  Municipality = municipality,
  Year = max(municipality_data$Year) + 1,
  Predicted_Value = as.numeric(predicted_value),
  Lower_95_CI = as.numeric(predicted_lower),
  Upper_95_CI = as.numeric(predicted_upper),
  ADF_P_Value = adf_test_p_value,
  Jarque_Bera_P_Value = jb_test_p_value,
  Breusch_Pagan_P_Value = bp_test_p_value,
  ARCH_LM_P_Value = arch_lm_p_value
))
}

cat("\\nPredictions for All Municipalities:\\n")
print(predictions)

output_file <- "Predictions_with_Original_Scale.xlsx"
write_xlsx(predictions, output_file)

cat("\\nPrediction process completed. Results saved to", output_file, "\\n")

library(Metrics)
VISI_MAX_FORECASTAI_2022 <-
\\

```

```

read_excel("/mif/stud3/2019/ugli6286/A_MAX_PROGNOZES.xlsx")
TIKROS_REIKSMES_2022 <-
\\
read_excel("/mif/stud3/2019/ugli6286/TIKROS_REIKSMES_MAX_2022.xlsx")
prognozes <-
\\
VISI_MAX_FORECASTAI_2022[, c("A", "B", "C", "D", "E", "F", "GKI")]
tikros_reiksmes
\\
<- TIKROS_REIKSMES_2022[, c("A", "B", "C", "D", "E", "F", "GKI")]
tikros_reiksmes <- as.numeric(tikros_reiksmes)
prognozes <- as.numeric(prognozes)
valid_indices <- !is.na(tikros_reiksmes) & !is.na(prognozes)
tikros_reiksmes <- tikros_reiksmes[valid_indices]
prognozes <- prognozes[valid_indices]
if (length(tikros_reiksmes) != length(prognozes)) {
  stop("The actual and forecast vectors must have the same length.")
}

# RMSE: Root Mean Square Error
rmse_value <- rmse(tikros_reiksmes, prognozes)
print(paste("RMSE:", rmse_value))

# MSE: Mean Squared Error
mse_value <- mse(tikros_reiksmes, prognozes)
print(paste("MSE:", mse_value))

# MAPE: Mean Absolute Percentage Error
mape_value <- mape(tikros_reiksmes, prognozes)
print(paste("MAPE:", mape_value))

library(readxl)
library(dplyr)

VISI_MAX_FORECASTAI_2022 <-

read_excel("/mif/stud3/2019/ugli6286/A_MAX_PROGNOZES.xlsx")

# Rename the "Municipality" column to "Dimensija"
VISI_MAX_FORECASTAI_2022 <- VISI_MAX_FORECASTAI_2022 %>%

rename(Dimensija = Municipality)

# Check if the column has been renamed successfully
print(colnames(VISI_MAX_FORECASTAI_2022))

library(ggplot2)

prognozes <- VISI_MAX_FORECASTAI_2022[, c("Dimensija", "GKI")]
tikros_reiksmes <- TIKROS_REIKSMES_2022[, c("Dimensija", "GKI")]

```

```

merged_data <- merge(proгноzes, tikros_reiksmes,
by = "Dimensija", suffixes = c("_prognoze", "_tikra"))

library(ggplot2)

merged_data <- merge(proгноzes, tikros_reiksmes, by = "Dimensija", suffixes =
c("_prognoze", "_tikra"))

merged_data <- merge(proгноzes, tikros_reiksmes, by = "Dimensija", suffixes =
c("_prognoze", "_tikra"))

ggplot(merged_data, aes(x = Dimensija)) +
  geom_bar(aes(y = GKI_tikra, fill = "Actual Values"), stat = "identity",
width = 0.6,

position = position_nudge(x = -0.15)) +
  geom_bar(aes(y = GKI_prognoze, fill = "Forecasted Values"),

stat = "identity", width = 0.4, position = position_nudge(x = 0.15)) +
  labs(title = "Didžiausių prognozuotų ir tikrų reikšmių pagal
ARIMA modelį palyginimas",
x = "Dimensija",
y = "GKI reikšmė") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  scale_fill_manual(name = "Legenda",
values = c("Actual Values" = "red",
\\
"Forecasted Values" = "blue"))

#####GAM#####

library(mgcv) # For fitting GAM models
library(dplyr) # For data manipulation
library(ggplot2) # For plotting
library(readxl) # For reading Excel files
library(lmtest) # For Durbin-Watson test
library(car) # For testing heteroscedasticity
library(corrplot)
library(writexl)
library(tseries) # For the Augmented Dickey-Fuller Test

# Load the dataset (update the path accordingly if not in the same directory)
data <- read_excel("/mif/stud3/2019/ugli6286/cleaned_df_A_long.xlsx")

data$Dimensija <- as.factor(data$Dimensija)

```

```

str(data)

# Augmented Dickey-Fuller test
is_stationary <- function(x) {
  adf_test <- adf.test(x, alternative = "stationary")
  return(adf_test$p.value < 0.05)
}

# Check if the data is stationary
stationary_check <- data %>%
  group_by(Dimensija) %>%
  summarize(is_stationary = is_stationary(Value_A))

print("Stationarity Check:")
print(stationary_check)

if (all(stationary_check$is_stationary)) {
  # If all time series are stationary, use original data
  print("All series are stationary. No differencing required.")
  data_diff <- data
  use_diff <- FALSE
} else {
  # If any time series is non-stationary, apply differencing
  print("Non-stationary series detected. Applying differencing.")
  data_diff <- data %>%
    group_by(Dimensija) %>%
    arrange(Year) %>%
    mutate(Value_A_diff = c(NA, diff(Value_A))) %>%
    ungroup() %>%
    na.omit() # Remove NA values created by differencing
  use_diff <- TRUE
}

# Check for correlation between numerical predictors
cor_matrix <- cor(data_diff %>% select_if(is.numeric))
print("Correlation Matrix:")
print(cor_matrix)
corrplot(cor_matrix)

# Fit a GAM model using appropriate data
response_var <- if (use_diff) "Value_A_diff" else "Value_A"
formula <- as.formula(paste(response_var, "~ s(Year, k = min(10, num_unique_years))
+ Dimensija"))
num_unique_years <- length(unique(data_diff$Year))
gam_model <- gam(formula, data = data_diff)

# Display a summary of the GAM model
summary(gam_model)

fitted_values <- fitted(gam_model)
residuals <- resid(gam_model)

```

```

par(mfrow = c(1, 1))
gam.check(gam_model)

# Check for autocorrelation in residuals using Durbin-Watson test
dw_test <- dwtest(gam_model)
print("Durbin-Watson Test for Autocorrelation:")
print(dw_test)

# If autocorrelation is significant, fit a GMM model with an autoregressive structure
if (dw_test$p.value < 0.05) {
  print("Autocorrelation detected. Fitting a GMM model with AR(1) structure.")

  gamm_model <-

  gamm(as.formula(paste(response_var, "~ s(Year,

    k = min(10, num_unique_years)) + Dimensija")),
  correlation = corAR1(form = ~ Year | Dimensija),
    data = data_diff)

  # Extract and summarize the GAM part of the model
  gam_model <- gamm_model$gam
  summary(gam_model)

  plot(gam_model, pages = 1)

  par(mfrow = c(1, 1))
  gam.check(gam_model)

  fitted_values <- fitted(gam_model)
  residuals <- resid(gam_model)
}

# Check for homoscedasticity using residuals vs fitted values plot
plot(fitted_values, residuals,
  main = "Residuals vs Fitted Values",
  xlab = "Fitted Values",
  ylab = "Residuals",
  pch = 20, col = "blue")
abline(h = 0, col = "red")

# Breusch-Pagan test approximation using linear model
lm_model <- lm(residuals ~ fitted_values)
bp_test <- bptest(lm_model)
print("Breusch-Pagan Test for Homoscedasticity:")
print(bp_test)

# ARCH LM Test for heteroscedasticity
arch_lm_test <- ArchTest(residuals, lags = 1)
print("ARCH LM Test for Conditional Heteroscedasticity:")
print(arch_lm_test)

# Check for normality of residuals using Q-Q plot and Shapiro-Wilk test

```

```

qqnorm(gam_model$residuals)
qqline(gam_model$residuals, col = "red")
shapiro_test <- shapiro.test(gam_model$residuals)
print("Shapiro-Wilk Test for Normality of Residuals:")
print(shapiro_test)

data_diff$pred_diff <- predict(gam_model, newdata = data_diff)

last_year <- max(data$Year)
future_data <- data.frame(
  Dimensija = unique(data$Dimensija),
  Year = last_year + 1
)

future_data$pred_diff <- predict(gam_model, newdata = future_data)

combined_data_diff <- bind_rows(data_diff, future_data)

# Back-transform predictions to original values
if (use_diff) {
  # For future data, add differenced predictions to the last actual value
  combined_data <- combined_data_diff %>%
    group_by(Dimensija) %>%
    mutate(pred = ifelse(is.na(Value_A), lag(Value_A) + pred_diff, Value_A)) %>%
    ungroup()
} else {
  # If no differencing was used, predictions are already on original scale
  combined_data <- combined_data_diff %>%
    mutate(pred = pred_diff)
}

# Plot actual vs predicted values
ggplot(combined_data, aes(x = Year, y = Value_A, color = Dimensija)) +
  geom_line() +
  geom_line(aes(y = pred), linetype = "dashed") +
  labs(title = "Actual vs Predicted Values (Including Future Prediction)",
       x = "Year",
       y = "Value",
       color = "Dimensija") +
  theme_minimal()

future_predictions <- combined_data %>%
  filter(Year == last_year + 1) %>%
  select(Dimensija, Year, pred)

print("Future Predictions (Original Scale):")
print(future_predictions)

write_xlsx(future_predictions, "A_MIN_fr_GAM_OriginalScale.xlsx")

```

```

gam_summary <- summary(gam_model)
param_coef_estimates <- gam_summary$p.coef
param_coef_se <- gam_summary$se[1:length(param_coef_estimates)]
param_coef_lower_95 <- param_coef_estimates - 1.96 * param_coef_se
param_coef_upper_95 <- param_coef_estimates + 1.96 * param_coef_se
param_coef_ci <- data.frame(
  Estimate = param_coef_estimates,
  Std.Error = param_coef_se,
  Lower_95 = param_coef_lower_95,
  Upper_95 = param_coef_upper_95
)
print("Confidence Intervals for Parametric Coefficients:")
print(param_coef_ci)

last_year <- max(data$Year)
future_data <- data.frame(
  Dimensija = unique(data$Dimensija),
  Year = last_year + 1
)

predictions <- predict(gam_model, newdata = future_data, se.fit = TRUE)

# Calculate 95% confidence intervals for predictions (on differenced scale)
future_data$pred_diff <- predictions$fit
future_data$se <- predictions$se.fit
future_data$lower_95_diff <- future_data$pred_diff - 1.96 * future_data$se
future_data$upper_95_diff <- future_data$pred_diff + 1.96 * future_data$se

last_values <- data %>%
  group_by(Dimensija) %>%
  summarize(last_Value_A = last(Value_A))

# Merge with future_data to back-transform
future_data <- merge(future_data, last_values, by = "Dimensija", all.x = TRUE)

# Back-transform to original values
future_data$pred <- future_data$last_Value_A + future_data$pred_diff
future_data$lower_95 <- future_data$last_Value_A + future_data$lower_95_diff
future_data$upper_95 <- future_data$last_Value_A + future_data$upper_95_diff

# Display back-transformed results
print("Predictions with Confidence Intervals for 2022 (Original Scale):")
print(future_data[, c("Dimensija", "Year", "pred", "lower_95", "upper_95")])

library(writexl)
write_xlsx(future_data, "predictions_with_ci_2022_original_scale.xlsx")

install.packages("Metrics")
library(Metrics)

```

```

VISI_MIN_FORECASTAI_2022 <-

read_excel("/mif/stud3/2019/ugli6286/GAM_FR_MAX.xlsx")
TIKROS_REIKSMES_2022 <-

read_excel("/mif/stud3/2019/ugli6286/TIKROS_REIKSMES_MAX_2022.xlsx")
proгноzes <- VISI_MIN_FORECASTAI_2022[, c("A", "B", "C", "D", "E", "F", "GKI")]
tikros_reiksmes <- TIKROS_REIKSMES_2022[, c("A", "B", "C", "D", "E", "F", "GKI")]
tikros_reiksmes <- as.numeric(tikros_reiksmes)
proгноzes <- as.numeric(proгноzes)
valid_indices <- !is.na(tikros_reiksmes) & !is.na(proгноzes)
tikros_reiksmes <- tikros_reiksmes[valid_indices]
proгноzes <- prognoses[valid_indices]
if (length(tikros_reiksmes) != length(proгноzes)) {
  stop("The actual and forecast vectors must have the same length.")
}

# RMSE: Root Mean Square Error
rmse_value <- rmse(tikros_reiksmes, prognoses)
print(paste("RMSE:", rmse_value))

# MSE: Mean Squared Error
mse_value <- mse(tikros_reiksmes, prognoses)
print(paste("MSE:", mse_value))

# MAPE: Mean Absolute Percentage Error
mape_value <- mape(tikros_reiksmes, prognoses)
print(paste("MAPE:", mape_value))

library(ggplot2)
library(dplyr)
library(readxl)

VISI_MIN_FORECASTAI_2022 <-

read_excel("/mif/stud3/2019/ugli6286/GAM_FR_MAX.xlsx")
TIKROS_REIKSMES_2022 <-

read_excel("/mif/stud3/2019/ugli6286/TIKROS_REIKSMES_MAX_2022.xlsx")
VISI_MIN_FORECASTAI_2022 <- VISI_MIN_FORECASTAI_2022 %>%
  rename(Dimensija = Dimension)
# Ensure prognoses and tikros_reiksmes have "Dimensija" and "GKI" columns
# (Make sure the column names "Dimensija" and "GKI" exist in the data)
proгноzes <- VISI_MIN_FORECASTAI_2022[, c("Dimensija", "GKI")]
tikros_reiksmes <- TIKROS_REIKSMES_2022[, c("Dimensija", "GKI")]

merged_data <- merge(proгноzes,

tikros_reiksmes, by = "Dimensija",

suffixes = c("_proгноze", "_tikra"))

merged_data$GKI_proгноze <- as.numeric(merged_data$GKI_proгноze)

```



```

        'Vilkaviškio r. sav.', 'Vilkaviškio r. sav.', 'Vilkaviškio r. sav.'
    ],
    'Year': [
        2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021,
        2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021,
        2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021,
        2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021,
        2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021
    ],
    'Value_A': [
        0.20556, 0.28418, 0.30282, 0.38959, 0.45862, 0.43443, 0.55733, 0.65501, 0.72768,
        0.30558, 0.39871, 0.41876, 0.53798, 0.59643, 0.62586, 0.6996, 0.76196, 0.79596,
        0.28714, 0.28933, 0.38787, 0.50116, 0.57036, 0.59479, 0.75885, 0.70464, 0.72289,
        0.28593, 0.42171, 0.51963, 0.61808, 0.68883, 0.71142, 0.78536, 0.83856, 0.81978,
        0.23772, 0.30878, 0.41908, 0.4818, 0.57453, 0.67193, 0.73805, 0.75046, 0.76828
    ]
}

```

```

data_df = pd.DataFrame(data)

import tensorflow as tf

results with confidence intervals

for all municipalities
forecast_results = pd.DataFrame()

# Loop through each unique municipality in the dataset
for dimensija in data_df['Dimensija'].unique():
    print(f"Processing Dimensija: {dimensija}")

    # Filter data for the current municipality
    data_filtered = data_df[data_df['Dimensija'] == dimensija].sort_values('Year')
    data_filtered = data_filtered[['Year', 'Value_A']].dropna()

    # Check stationarity using the Augmented Dickey-Fuller test
    adf_test = adfuller(data_filtered['Value_A'])
    is_stationary = adf_test[1] < 0.05
    print(f"ADF Test p-value for {dimensija}: {adf_test[1]}")
    print(f"Is the series stationary? {'Yes' if is_stationary else 'No'}")

    # Apply differencing if non-stationary
    if not is_stationary:
        data_filtered['Value_A_diff'] = data_filtered['Value_A'].diff().dropna()
        use_diff = True
    else:
        data_filtered['Value_A_diff'] = data_filtered['Value_A']
        use_diff = False

    data_filtered = data_filtered.dropna().reset_index(drop=True)

    scaler = MinMaxScaler()

```

```

data_filtered['Value_A_scaled'] = scaler.fit_transform(data_filtered

[['Value_A_diff']])

# Prepare data for LSTM
def prepare_lstm_data(data, time_steps=3):
    X, y = [], []
    for i in range(len(data) - time_steps):
        X.append(data[i:(i + time_steps)])
        y.append(data[i + time_steps])
    return np.array(X), np.array(y)

# Prepare data with a time step of 3
time_steps = 3
X, y = prepare_lstm_data(data_filtered['Value_A_scaled'].values,
\\
time_steps=time_steps)
X = X.reshape(X.shape[0], X.shape[1], 1)

# Define the LSTM model with Monte Carlo Dropout
model = Sequential([
    Input(shape=(X.shape[1], 1)),
    LSTM(units=50, return_sequences=False),
    Dropout(0.2), # Add dropout for Monte Carlo Dropout
    Dense(units=1)
])

model.compile(optimizer='adam', loss='mean_squared_error')

# Early stopping to prevent overfitting
early_stopping = EarlyStopping(monitor='loss', patience=2)

# Train the model with 5 epochs
model.fit(X, y, epochs=5, batch_size=16, verbose=1, callbacks=[early_stopping])

# Prepare input for 1-step ahead forecast (2022)
last_sequence = X[-1].reshape(1, X.shape[1], 1)

confidence intervals
n_simulations = 100
forecasts = []

for _ in range(n_simulations):
    # Enable training mode for dropout
    forecast = model(last_sequence, training=True)
    forecast = scaler.inverse_transform(forecast)
    if use_diff:
        forecast = forecast + data_filtered['Value_A'].iloc[-1]
    forecasts.append(forecast[0][0])

mean_forecast = np.mean(forecasts)
lower_bound = np.percentile(forecasts, 2.5)
upper_bound = np.percentile(forecasts, 97.5)

```

```

forecast_results = pd.concat([forecast_results, pd.DataFrame({
    'Dimensija': [dimensija],
    'Year': [2022],
    'Forecasted Value': [mean_forecast],
    'Lower Bound (95%)': [lower_bound],
    'Upper Bound (95%)': [upper_bound]
})], ignore_index=True)

print("Forecasted Values for 2022 with Confidence Intervals for

all municipalities:")
print(forecast_results)

output_file = "LSTM_forecasts_2022_with_confidence_intervals.xlsx"
forecast_results.to_excel(output_file, index=False)
files.download(output_file)

print("All municipalities processed and results with confidence intervals saved.")

from sklearn.metrics import mean_squared_error, mean_absolute_percentage_error
import numpy as np

tikros_reiksmes_data = {
    "A": [0.74140, 0.80879, 0.79039, 0.80556, 0.76796],
    "B": [0.13774, 0.10744, 0.06810, 0.05530, 0.09606],
    "C": [0.35373, 0.44110, 0.42171, 0.28825, 0.45036],
    "D": [0.34077, 0.43364, 0.40508, 0.54266, 0.36401],
    "E": [0.48025, 0.25536, 0.41586, 0.20636, 0.36403],
    "F": [0.41537, 0.57436, 0.51772, 0.57311, 0.52832],
    "GKI": [0.453761, 0.476057, 0.473453, 0.452740, 0.464890]
}

proгноzes_data = {
    "A": [0.717336, 0.315453, 0.692689, 0.372487, 0.785212],
    "B": [0.134374, 0.088556, 0.045192, 0.021080, 0.090844],
    "C": [0.352636, 0.456985, 0.389510, 0.309845, 0.357391],
    "D": [0.286674, 0.345632, 0.267010, 0.480986, 0.339668],
    "E": [0.373410, 0.162068, 0.317972, 0.123349, 0.269467],
    "F": [0.460638, 0.523043, 0.536098, 0.570110, 0.499181],
    "GKI": [0.431114, 0.295376, 0.410607, 0.299064, 0.438736]
}

actual_numeric_values = np.hstack([tikros_reiksmes_data[key] for
key in tikros_reiksmes_data.keys()])
forecasted_numeric_values = np.hstack([proгноzes_data[key] for
key in prognazes_data.keys()])

# Calculate RMSE for the numeric data
rmse = np.sqrt(mean_squared_error(actual_numeric_values, forecasted_numeric_values))

```

```

# Calculate MSE for the numeric data
mse = mean_squared_error(actual_numeric_values, forecasted_numeric_values)

# Calculate MAPE for the numeric data
mape = mean_absolute_percentage_error(actual_numeric_values, forecasted_numeric_values)

# Print the metrics
print("RMSE:", rmse)
print("MSE:", mse)
print("MAPE:", mape)
#LSTM

```

#####BiLSTM (PYTHON)#####

```

import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from statsmodels.tsa.stattools import adfuller
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Bidirectional, LSTM, Dense, Input, Dropout
from tensorflow.keras.callbacks import EarlyStopping
import matplotlib.pyplot as plt
from google.colab import files

data = {
    'Dimensija': [
        'Kalvarijos sav.', 'Kalvarijos sav.', 'Kalvarijos sav.', 'Kalvarijos sav.',
        'Kalvarijos sav.', 'Kalvarijos sav.', 'Kalvarijos sav.', 'Kalvarijos sav.',
        'Kalvarijos sav.', 'Kelmės r. sav.', 'Kelmės r. sav.', 'Kelmės r. sav.',
        'Kelmės r. sav.', 'Kelmės r. sav.', 'Kelmės r. sav.', 'Kelmės r. sav.',
        'Kelmės r. sav.', 'Kelmės r. sav.', 'Lazdijų r. sav.', 'Lazdijų r. sav.',
        'Lazdijų r. sav.', 'Lazdijų r. sav.', 'Lazdijų r. sav.', 'Lazdijų r. sav.',
        'Lazdijų r. sav.', 'Lazdijų r. sav.', 'Lazdijų r. sav.', 'Lazdijų r. sav.',
        'Lazdijų r. sav.', 'Lazdijų r. sav.', 'Lazdijų r. sav.', 'Skuodo r. sav.',
        'Skuodo r. sav.', 'Skuodo r. sav.', 'Skuodo r. sav.', 'Skuodo r. sav.',
        'Skuodo r. sav.', 'Skuodo r. sav.', 'Skuodo r. sav.', 'Skuodo r. sav.',
        'Vilkaviškio r. sav.', 'Vilkaviškio r. sav.', 'Vilkaviškio r. sav.',
        'Vilkaviškio r. sav.', 'Vilkaviškio r. sav.', 'Vilkaviškio r. sav.',
        'Vilkaviškio r. sav.', 'Vilkaviškio r. sav.', 'Vilkaviškio r. sav.'
    ],
    'Year': [
        2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021,
        2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021,
        2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021,
        2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021,
        2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021
    ],
    'Value_A': [
        0.20556, 0.28418, 0.30282, 0.38959, 0.45862, 0.43443, 0.55733, 0.65501,
        0.72768, 0.30558, 0.39871, 0.41876, 0.53798, 0.59643, 0.62586, 0.6996,

```

```

        0.76196, 0.79596, 0.28714, 0.28933, 0.38787, 0.50116, 0.57036, 0.59479,
        0.75885, 0.70464, 0.72289, 0.28593, 0.42171, 0.51963, 0.61808, 0.68883,
        0.71142, 0.78536, 0.83856, 0.81978, 0.23772, 0.30878, 0.41908, 0.4818,
        0.57453, 0.67193, 0.73805, 0.75046, 0.76828
    ]
}

data_df = pd.DataFrame(data)
forecast_results = pd.DataFrame()

for dimensija in data_df['Dimensija'].unique():
    print(f"Processing Dimensija: {dimensija}")

    data_filtered = data_df[data_df['Dimensija'] == dimensija].
    \
    sort_values('Year')
    data_filtered = data_filtered[['Year', 'Value_A']].dropna()

    adf_test = adfuller(data_filtered['Value_A'])
    is_stationary = adf_test[1] < 0.05

    if not is_stationary:
        data_filtered['Value_A_diff'] = data_filtered['Value_A'].diff().dropna()
        use_diff = True
    else:
        data_filtered['Value_A_diff'] = data_filtered['Value_A']
        use_diff = False

    data_filtered = data_filtered.dropna().reset_index(drop=True)

    scaler = MinMaxScaler()
    data_filtered['Value_A_scaled'] = scaler.fit_transform(data_filtered

    [['Value_A_diff']])

def prepare_lstm_data(data, time_steps=3):
    X, y = [], []
    for i in range(len(data) - time_steps):
        X.append(data[i:(i + time_steps)])
        y.append(data[i + time_steps])
    return np.array(X), np.array(y)

time_steps = 3
X, y = prepare_lstm_data(data_filtered['Value_A_scaled'].values,
\
time_steps=time_steps)
X = X.reshape(X.shape[0], X.shape[1], 1)

model = Sequential([
    Input(shape=(X.shape[1], 1)),
    Bidirectional(LSTM(units=50, return_sequences=False)),
    Dropout(0.2),
    Dense(units=1)
])

```

```

model.compile(optimizer='adam', loss='mean_squared_error')
early_stopping = EarlyStopping(monitor='loss', patience=2)

model.fit(X, y, epochs=5, batch_size=16, verbose=1, callbacks=[early_stopping])

last_sequence = X[-1].reshape(1, X.shape[1], 1)
n_simulations = 100
forecasts = []

for _ in range(n_simulations):
    forecast = model(last_sequence, training=True)
    forecast = scaler.inverse_transform(forecast)
    if use_diff:
        forecast = forecast + data_filtered['Value_A'].iloc[-1]
    forecasts.append(forecast[0][0])

mean_forecast = np.mean(forecasts)
lower_bound = np.percentile(forecasts, 2.5)
upper_bound = np.percentile(forecasts, 97.5)

forecast_results = pd.concat([forecast_results, pd.DataFrame({
    'Dimensija': [dimensija],
    'Year': [2022],
    'Forecasted Value': [mean_forecast],
    'Lower Bound (95%)': [lower_bound],
    'Upper Bound (95%)': [upper_bound]
})], ignore_index=True)

print("Forecasted Values for 2022 with Confidence Intervals for all municipalities:")
print(forecast_results)

output_file = "BiLSTM_forecasts_2022_with_confidence_intervals.xlsx"
forecast_results.to_excel(output_file, index=False)
files.download(output_file)

print("All municipalities processed and results with confidence intervals saved.")

from sklearn.metrics import mean_squared_error, mean_absolute_percentage_error
import numpy as np

tikros_reiksmes_data = {
    "A": [0.74140, 0.80879, 0.79039, 0.80556, 0.76796],
    "B": [0.13774, 0.10744, 0.06810, 0.05530, 0.09606],
    "C": [0.35373, 0.44110, 0.42171, 0.28825, 0.45036],
    "D": [0.34077, 0.43364, 0.40508, 0.54266, 0.36401],
    "E": [0.48025, 0.25536, 0.41586, 0.20636, 0.36403],
    "F": [0.41537, 0.57436, 0.51772, 0.57311, 0.52832],
    "GKI": [0.453761, 0.476057, 0.473453, 0.452740, 0.464890]
}

proгноzes_data = {
    "A": [0.729870, 0.376812, 0.689834, 0.328980, 0.793293],

```

```

      "B": [0.138884, 0.087790, 0.047068, 0.021926, 0.093110],
      "C": [0.359308, 0.450323, 0.386282, 0.298581, 0.366773],
      "D": [0.297661, 0.345876, 0.288346, 0.485979, 0.344706],
      "E": [0.378030, 0.173031, 0.331055, 0.125038, 0.282342],
      "F": [0.467318, 0.526605, 0.534342, 0.570335, 0.502134],
      "GKI": [0.439237, 0.315167, 0.413636, 0.285841, 0.445429]
    }

    actual_numeric_values = np.hstack([tikros_reiksmes_data[key] for
    key in tikros_reiksmes_data.keys()])
    forecasted_numeric_values = np.hstack([proгноzes_data[key] for
    key in prognозes_data.keys()])

    # Calculate RMSE for the numeric data
    rmse = np.sqrt(mean_squared_error(actual_numeric_values, forecasted_numeric_values))

    # Calculate MSE for the numeric data
    mse = mean_squared_error(actual_numeric_values, forecasted_numeric_values)

    # Calculate MAPE for the numeric data
    mape = mean_absolute_percentage_error(actual_numeric_values,
    forecasted_numeric_values)

    # Print the metrics
    print("RMSE:", rmse)
    print("MSE:", mse)
    print("MAPE:", mape)
    #BiLSTM

#####Monte Karlo SIMULIACIJU KODAS#####

# Load necessary libraries
library(readxl)
library(openxlsx)
library(dplyr)

arima_file <- "/mif/stud3/2019/ugli6286/ARIMA_CI.xlsx"
gam_file <- "/mif/stud3/2019/ugli6286/GAM_CI.xlsx"
lstm_file <- "/mif/stud3/2019/ugli6286/LSTM_CI.xlsx"
bilstm_file <- "/mif/stud3/2019/ugli6286/BiLSTM_CI.xlsx"

arima_data <- read_excel(arima_file) %>%
  select(Savivaldybė, Subindeksas, Prognozė, PI_Lower, PI_Upper)

gam_data <- read_excel(gam_file) %>%
  select(Savivaldybė, Subindeksas, Prognozė, PI_Lower, PI_Upper)

lstm_data <- read_excel(lstm_file) %>%
  select(Savivaldybė, Subindeksas, Prognozė, PI_Lower, PI_Upper)

```

```

bilstm_data <- read_excel(bilstm_file) %>%
  select(Savivaldybė, Subindeksas, Prognozė, PI_Lower, PI_Upper)

arima_data <- arima_data %>%
  mutate(Prognozė = as.numeric(Prognozė),
         PI_Lower = as.numeric(PI_Lower),
         PI_Upper = as.numeric(PI_Upper))

gam_data <- gam_data %>%
  mutate(Prognozė = as.numeric(Prognozė),
         PI_Lower = as.numeric(PI_Lower),
         PI_Upper = as.numeric(PI_Upper))

lstm_data <- lstm_data %>%
  mutate(Prognozė = as.numeric(Prognozė),
         PI_Lower = as.numeric(PI_Lower),
         PI_Upper = as.numeric(PI_Upper))

bilstm_data <- bilstm_data %>%
  mutate(Prognozė = as.numeric(Prognozė),
         PI_Lower = as.numeric(PI_Lower),
         PI_Upper = as.numeric(PI_Upper))

arima_data <- arima_data %>%
  filter(!is.na(PI_Lower) & !is.na(PI_Upper) & PI_Lower < PI_Upper)

gam_data <- gam_data %>%
  filter(!is.na(PI_Lower) & !is.na(PI_Upper) & PI_Lower < PI_Upper)

lstm_data <- lstm_data %>%
  filter(!is.na(PI_Lower) & !is.na(PI_Upper) & PI_Lower < PI_Upper)

bilstm_data <- bilstm_data %>%
  filter(!is.na(PI_Lower) & !is.na(PI_Upper) & PI_Lower < PI_Upper)

# Combine data into a single list for processing
model_data <- list(ARIMA = arima_data, GAM = gam_data,
                  LSTM = lstm_data, BiLSTM = bilstm_data)

subindex_weights <- c(0.3, 0.2, 0.1, 0.1, 0.15, 0.15)

# Monte Carlo simulation parameters
n_sim <- 100 # Number of simulations
set.seed(123)

calculate_gki_ci <- function(data, weights, n_sim) {
  municipalities <- unique(data$Savivaldybė)
  gki_results <- data.frame(Municipality = character(),

```

```

        GKI_Mean = numeric(),
        GKI_Lower = numeric(),
        GKI_Upper = numeric())

for (mun in municipalities) {

  mun_data <- subset(data, Savivaldybė == mun)

  mun_data <- mun_data[order(mun_data$Subindeksas), ]

  mun_data <- mun_data[!is.na(mun_data$PI_Lower)
                        & !is.na(mun_data$PI_Upper), ]
  mun_data <- mun_data[mun_data$PI_Lower < mun_data$PI_Upper, ]

  if (nrow(mun_data) < length(weights)) {
    warning(paste("Skipping municipality:",
                  mun, "- insufficient data"))
    next
  }

  # Monte Carlo simulations for each subindex
  simulations <- replicate(n_sim, {
    subindex_values <- mapply(function(lower, upper) {
      runif(1, min = lower, max = upper)
    }, mun_data$PI_Lower, mun_data$PI_Upper)

    # Calculate GKI for this simulation
    sum(subindex_values * weights)
  })

  # Calculate mean and 95% confidence intervals for GKI
  gki_mean <- mean(simulations)
  gki_ci <- quantile(simulations, probs = c(0.025, 0.975))

  gki_results <- rbind(gki_results, data.frame(
    Municipality = mun,
    GKI_Mean = gki_mean,
    GKI_Lower = gki_ci[1],
    GKI_Upper = gki_ci[2]
  ))
}

return(gki_results)
}

# Process each model's data
results_list <- lapply(model_data, calculate_gki_ci,
                        weights = subindex_weights, n_sim = n_sim)

# Save results to Excel

```

```
output_file <- "GKI_CI_Results.xlsx"
wb <- createWorkbook()

for (model_name in names(results_list)) {
  addWorksheet(wb, model_name)
  writeData(wb, model_name, results_list[[model_name]])
}

saveWorkbook(wb, output_file, overwrite = TRUE)

cat("GKI confidence intervals saved to:", output_file)
```