LATVIJAS UNIVERSITĀTE

DATORIKAS FAKULTĀTE

Laika rindas datu analīze un pielietojums

BAKALAURA DARBS

Autors: **Ričards Kalniņš**

Studenta apliecības Nr.: rk15035

Darba vadītājs: profesors Dr.sc.comp. Māris Vītiņš

RĪGA 2019

ANOTĀCIJA

Darba mērķis ir laika rindas tehnoloģijas, analīzes, prognozēšanas un tās praktisko pielietojumu izpēte. Darba ietvaros tika veikta laika rindas analīzes un prognozēšanas metožu apskate, laika rindas problēmsituāciju izpēte, kā arī tika apskatītas nozares, kurās laika rindas tiek pielietotas. Papildus tika apskatītas un salīdzinātas divas vadošās laika rindas datubāzu pārvaldības sistēmas.

Pēc izpētes laikā iegūtajiem rezultātiem darba praktiskajā daļā tika veikta laika rindas datu prognozēšanas metožu pielietošana – Latvijas eksporta apjoma prognozēšana balsoties uz iegūtajiem datiem no 1995.gada līdz 2019.gadam.

Atslēgvārdi: Laika rindas, analīze, prognozēšana, datubāze.

ABSTRACT

TIME SERIES DATA ANALYSIS AND APPLICATION

The main goal of this paper is time series technology, analysis, forecasting and its practial application research. The research contains a review of time series analysis and forecasting methods, common issues in timeseries were investigated, as well as the sectors in which timeseries are applied. In addition, two leading time series database management systems were considered and compared.

In the practical part of the paper, according to the results obtained during the research, application of time series data forecasting methods were perfomed – forecasting the volume of Latvian exports with the data obtained form 1995 to 2019.

Keywords: Time series, analysis, forecasting, database.

SATURS

APZĪMĒJUMU SARAKSTS

Laika rinda – skaitļu virkne, kas raksturo kāda procesa vai parādības izmaiņas laikā.

Lietu internets – fizisku ierīču starptīklošana.

MAD – vidējā absolūtā novirze.

MSD – vidējā kvadrātiskā novirze.

MAPE – vidējā absolūtā procentuālā kļūda.

Moving Average – slīdošais vidējais.

Exponencial Smoothing – eksponenciālā izlīdzināšana.

ARIMA – automātiski regresējošs integrēts slīdošais vidējais.

Flux – skriptu valoda vaicājumu veidošanai.

SQL – strukturēto vaicājumu valoda.

Tagset – birku kopa.

Timestamp – laika zīmogs.

Python – interpretējama objektorientētā skriptu valoda.

IEVADS

Laika rindas mūsdienās aizvien vairāk gūst popularitāti nozarēs, kā, piemēram, autonomajos transportlīdzekļos, viedajās mājās un finanšu tirgū. Šajās nozarēs laika rindas datus apvienojot ar dažādām statistikas analīzes metodēm ir iespējams precīzāk izprast, kas norisinājies pagātnē, novērot, kas notiek pašlaik un prognozēt, kādas izmaiņas gaidāmas nākotnē.

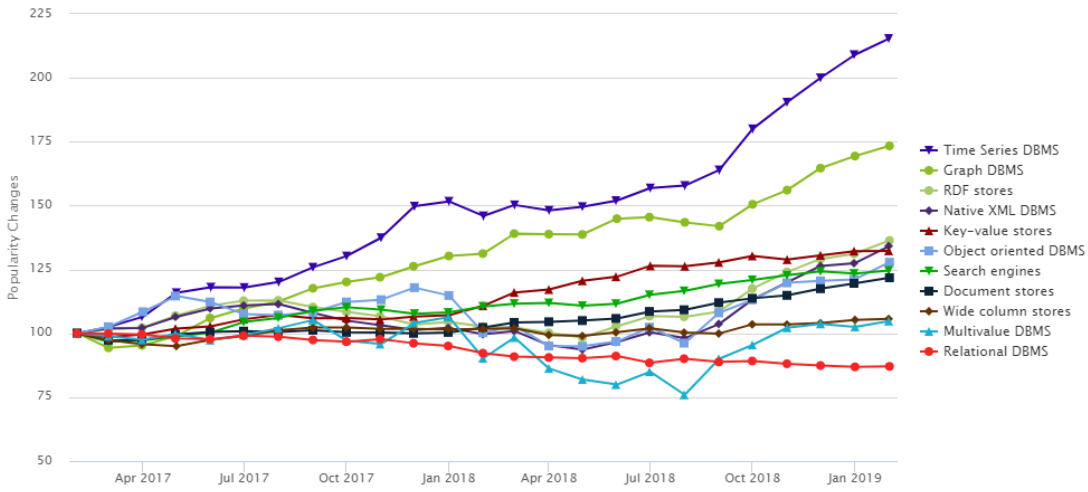
Laika rindas nav jauna tehnoloģija, taču tā savu popularītāti iegūst ar vien vairāk pateicoties citiem tehnoloģiskiem sasniegumiem, kā, piemēram, lietu internetam (Angliski – Internet of Things), kur vairākas lietu interneta veidojošas ierīces (viedierīces, sensori) veic mērījumus par kādām parādībam vai notikumiem. Laika rindas dati ir lietiskā interneta galvenā sastāvdaļa un tie palīdz ne tikai uzglabāt šos datus, bet arī tos apstrādāt, lai no tiem iegūtu noderīgus secinājumus un izvirzītu prognozes. Nozares eksperti uzskata, ka laika rindas datu apjoms pieaugs eksponenciāli tuvākajos 10 gados. [1]

Darba autors izvirzīja mērķi veikt laika rindas tehnoloģijas, pielietojuma un analīzes izpēti, un vadoties pēc izpētes veikt laika rindas datu prognozēšanu, kas pielietotu iegūtās zināšanas teorijas izpētes laikā praktiskā veidā.

Darba izstrādes laikā kā pētniecības metodes tika izmantotas pieejamās literatūras analīze.

1 LAIKA RINDAS

Interese un pieprasījums pēc laika rindas datubāzu pārvaldības sistēmām pieaug attīstoties tehnoloģijām un cilvēku vēlmei atvieglot savu ikdienu. Laika rindas datubāzu pārvaldības sistēmu popularitātes pieaugums ir redzams diagrammā (skat. att. 1.1).



att. 1.1 ***Laika rindas datubāzu pārvaldības sistēmu popularitātes pieaugums pēdējos 24 mēnešos.*** *[2]*

Grafikā violetā līnija attēlo laika rindas datubāzu pārvaldes sistēmu popularitātes pieaugumu (skat. att. 1.1) un kā redzams, pēdējo 24 mēnešu laikā tieši laika rindas datubāzu pārvaldes sistēmas ir guvušas straujāko popularitātes pieaugumu. [2]

Šajā nodaļā autors aprakstīs, kas tiek uzskatītas par laika rindām, kādās nozārēs tās tiek praktiski pielietotas un kādas ir galvenās problēmsituācijas laika rindās.

1.1 Laika rindas uzbūve un darbība

Laika rinda ir reālu skaitļu virkne, kas ir kāda mainīga lieluma novērojuma rezultāts, kuru iegūst veicot mērījumus kādā fiziskā sistēmā regulāros laika intervālos. Laika rindu veido dati, kuru identifikators ir laika vienība, kurā ir veikts mērījums. Katrs laika rindas ieraksts sastāv no diviem elementiem – laika norāde un līmenis, jeb mainīgā mērījuma rezultāts konkrētā laika vienībā. Laika rindas, galvenokārt, raksturo kādas parādības (mainīgā) pārmaiņas laikā. [3] Tās tiek pierakstītas šādā formā:

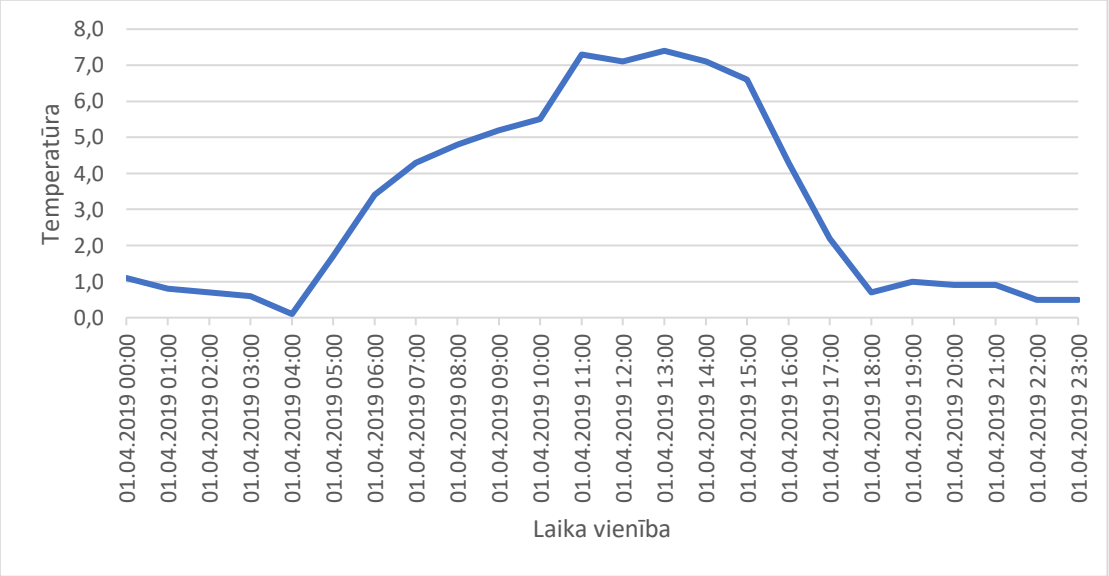
$$Y = \{Y_t : t \in T\}$$

Kur Y_t ir mērījuma vērtība konkrētā laika vienībā un T ir laika vienību kopa.

Laika rindām parasti ir trīs kopīgas lietas:

1. Saņemtie dati tiek saglabāti kā jauni ieraksti;
2. Dati parasti pienāk laika kārtībā;
3. Laiks ir primārā ass.

Tieši šis iemesls, ka katrs saņemtais mērījums tiek saglabāts kā jauns ieraksts, nevis katreiz atjaunots, ir tas, kas laika rindām dod vērtību. Tās ļauj izmērīt izmaiņas sistēmā - analizēt, kā kāda parādība ir mainījusies pagātnē, novērot, kā tā mainās dotajā brīdī un prognozēt tās izmaiņas nākotnē.



att. 1.2 ***Gaisa temperatūra Igaunijā, Heltermā.*** *[4]*

Laika rindas grafiskais attēlojums, kurā 24 stundu posmā ir veikti gaisa temperatūras mērījumi ar 1 stundas intervālu starp tiem (skat. att. 1.2).

1.2 Laika rindas datu pielietojums

Pamatā, laika rindas izmanto nozarēs, kurās nepieciešams novērot vai paredzēt, kā dati mainās laikā, kur laiks nav tikai mērījums, bet primārā ass. Tās ir nozares, kā, piemēram, valūtu un akciju tirgus, autonomie transportlīdzekļi, viedās mājas, kā arī uzņēmumos, kuros nepieciešams prognozēt apgrozījumu.

1.2.1 Valūtu un akciju tirgus

Valūtu un akciju tirgus iespējams ir vissenākā nozare, kuras analīzes un prognozēšanas pamatā ir laika rindas, lai novērotu kādu valūtu vai akciju vērtību izmaiņas laikā. Dotājās nozarēs dati tiek analizēti un pētīti ļoti plaši, taču tos prognozēt ir visai sarežģīti, jo valūtu un akciju tirgus svārstības ietekmē daudzi arējie faktori, kurus nevienmēr ir iespējams paredzēt.

1.2.2 Autonomie transportlīdzekļi

Pēdējā laikā ar vien lielāki sasniegumi ir manāmi autonomo transportlīdzekļu nozarē, kuru darbības pamātā ir daudzu sensoru sadarbība un iegūto mērījumu analīze reālajā laikā, šo sensoru dati ir laika rindas dati, kuru analīze un prognozēšana ir pamats autonomo transportlīdzekļu darbībā. Autonomie transportlīdzekļi ir ļoti komplicēta tehnoloģija, taču, lai saprastu, kā laika rindas tiek pielietotas transportlīdzekļos, pietiek apskatīt, kā strādā transportlīdzkļu adaptīvās kruīza kontroles sistēmas. [5]

Veids, kā tiek veikti mērījumi atšķiras starp transportlīdzekļu izstrādātajiem, taču sistēmas galvenais mērķis ir iegūt informāciju par priekšā esošā transportlīdzekļa ātrumu un attālumu no dotā transportlīdzekļa, lai ievērotu drošu distanci starp transportlīdzekļiem un avārijas situācijā spētu transportlīdzekli apturēt bez fiziskas iejaukšanās.

Veiktie mērījumi tiek saņemti un saglabāti datubāzē, tālāk analīzējot iegūtos datus un prognozējot nākotni, tiek izdarīti secinājumi un attiecīgas darbības ar transportlīdzekli – ātruma palielināšana vai samazināšana.

1.2.3 Viedās mājas

Viedās mājās izmanto laika rindu tehnoloģiju un to analīzi, lai atvieglotu cilvēku ikdienu. Līdzīgi kā autonomajos transportlīdzekļos, arī viedajās mājās vairāku sensoru kopums veic mērījumus, kurus galvenā sistēmas komponente analizē, veicot secinājumus un darbības, lai uzturētu klimatu vai drošību mājā. Latvijā uzņēmums “MeshRobotics” piedāvā radiatoru kontrolieru sistēmas, kuras mēra un analizē istabas temperatūru un attiecīgi pielāgo radiatoru siltumatdevi. [6]

1.3 Laika rindu problēmsituācijas

Laika rindās ir arī zinamās problēmsituācijas, kā, piemēram, trūkstoši mērījumi, dublikāti mērījumi un aizkavēti mērījumi. Minētās problēmsituācijas var radīt problēmas nākotnē, kad saņemtie dati tiek analizēti un izmantojot tos, tiek veiktas prognozes.

1.3.1 Trūkstoši mērījumi

Trūkstošu mērījumu problēmsituācijas var iedalīt trīs apakšgrupās:

- mērījumi nav saņemti,
- nepilnīgi mērījumi saņemti,
- saņemti neizmantojami jeb kļūdaini mērījumi.

Gadījumā, ja kādas laika vienības mērijums nav saņemts, rodas laika rindas definīcijas pretruna, kas nosaka, ka intervāliem starp mērijumiem ir jābūt regulāriem, tāpēc mūsdienās "modernās" laika rindās, intervālu regularitāte nav tik stingri nosacīta, taču to esamība ir ieteicama precīzāku prognožu veikšanai. [7]

tabula. 1.1 ***Piemērs nesaņemtiem datiem***

Laiks	**Temperatūra (C°)**
01.04.2019 00:00	1,1
01.04.2019 01:00	0,8
01.04.2019 03:00	0,6
01.04.2019 04:00	0,1

Kā redzams dotajā piemērā (skat. tabulu. 1.1), iztrūkst dati ar laika norādi 01.04.2019 02:00, kas var radīt neprecizitātes laika rindas analīzē un nākamo datu prognozēšanā.

Nepilnīgu mērījumu gadījums var rasties vairaku dimensiju laika rindas gadījumā, kur vienā laika vienībā tiek veikti vairāki mērījumi un kāds no tiem nav ticis nolasīts vai saņemts.

tabula. 1.2 ***Piemērs nepilnīgiem mērījumiem***

Laiks	**Temperatūra (C°)**	**Relatīvais gaisa mitrums (%)**
01.04.2019 00:00	1,1	68
01.04.2019 01:00	0,8	72
01.04.2019 02:00		***75***
01.04.2019 03:00	0,6	77
01.04.2019 04:00	***0,1***	

Dotajā tabulā redzams (skat. tabulu. 1.2), ka iztrūkst temperatūras mērījuma rezultāts laika norādē 01.04.2019 02:00 un relatīvā gaisa mitruma mērījums ar laika norādi 01.04.2019 04:00. Līdzīgi kā nesaņemtu datu gadījumā, laika rindas ar nepilnīgiem datiem var radīt neprecizitātes laika rindas analīzē un nākamo datu prognozēšanā.

Neizmantojamu jeb kļūdainu mērījumu gadījumā mainīgā vērtība neatbilst mainīgā datu tipam, vai arī vērtība, ar neticamu starpību, atšķiras no pārējās laika rindas.

tabula. 1.3 ***Piemērs neizmantojamiem jeb kļūdaniem mērījumiem***

Laiks	**Temperatūra (C°)**	**Relatīvais gaisa mitrums (%)**
01.04.2019 00:00	1,1	68
01.04.2019 01:00	0,8	72
01.04.2019 02:00	***NULL***	***75***
01.04.2019 03:00	0,6	77
01.04.2019 04:00	***0,1***	***873***

Tabulā attēlots, ka laika norādē 01.04.2019 02:00 iegūtie mērījumi neatbilst laika rindas datu tipam un laika norādē 01.04.2019 04:00 iegūtie mērījumi drastiski atšķiras no iepriekš iegūtajiem mērījumiem (skat. tabulu 1.3), kas liecina, ka dati, iespējams, ir kļūdaini. Dotajā piemērā datu nekorektumu var viegli atpazīt. Pirmajā gadījumā vērtība ir neeksistējoša un otrajā vērtība pārsniedz maksimāli iespējamo vērtību, taču, lai atrisinātu problēmas ar kļūdainiem datiem, katra situācijai ir jāizvērtē individuāli, vienkāršākos gadījumos nosakot vērtības minimālās un maksimālās vērtības, un sarežģītākos gadījumos, iespējams, konstruējot algoritmu, kas kļūdas var identificēt.

1.3.2 Dublikāti mērījumi

Dublikātu mērījumu problēmsituācijas var iedalīt divās apakšgrupās:

- vienā laika vienībā saņemti vairāki vienādi mērījumi,
- vienā laika vienībā saņemti vairāki dažādi mērījumi.

Dublikātu mērījumu iemesli var būt dažādi, tos var izraisīt kāda kļūda sensorā, kas datus iegūst vai kļūda galvenajā komponentē, kas datus saņem no sensora.

tabula. 1.4 ***Piemērs duplikātiem mērījumiem ar vienādu rezultātu***

Laiks	**Temperatūra (C°)**
01.04.2019 00:00	1,1
01.04.2019 01:00	0,8
01.04.2019 02:00	0,7
01.04.2019 03:00	***0,6***
01.04.2019 03:00	***0,6***
01.04.2019 04:00	0,1

Viens no preventīvajiem risinājumiem šādām situācijām varētu būt datubāzes pareiza nokonfigurēšana, kas neļautu ievietot datus ar vienādiem indeksiem, šajā gadījumā laika vienību.

tabula. 1.5 ***Piemērs dublikātiem mērījumiem ar dažādiem rezultātiem***

Laiks	**Temperatūra (C°)**
01.04.2019 00:00	1,1
01.04.2019 01:00	0,8
01.04.2019 02:00	0,7
01.04.2019 03:00	***0,6***
01.04.2019 03:00	***0,5***
01.04.2019 04:00	0,1

1.3.3 Aizkavēti mērījumi

Aizkavētu mērījumu gadījums var atvasināties no nesaņemtu mērījumu problēmsituācijas. Tiek konstatēts, ka dati noteiktajā laikā nav nolasīti vai saņemti no kā rodas nesaņemtu mērījumu problēmsituācija, taču vēlāk, mērījumi tiek saņemti ar aizturi un datubāzē dati vairs nav sakārtoti pēc laika norādēm.

tabula. 1.6 ***Piemērs aizkavētiem mērījumiem***

Laiks	**Temperatūra (C°)**
01.04.2019 00:00	1,1
01.04.2019 01:00	0,8
01.04.2019 03:00	0,6
01.04.2019 04:00	0,1
01.04.2019 02:00	***0,7***

Tabulā redzams, ka mērījumi ar laika norādi 01.04.2019 02:00 ir saņemti ar aizturi, kas rada pretrunu ar laika rindas definīciju, kas nosaka, ka laika rindā datiem ir jābūt sakārtotiem pēc laika indeksācijas (skat. tabulu 1.6).

2 LAIKA RINDU ANALĪZES METODES

Laika rindu uzvedībā ir novērojamas 4 komponentes – sezonālās svārstības, tendence, cikliskās svārstības un nejaušas svārstības. [3] Sezonālās svārstības jeb periodiskās svārstības atkārtojas noteiktos laika intervālos, kā, piemēram, dienās, nedēļās, mēnešos utml. Tendence ir laika rindas pamatbūtība, kura ir salīdzinoši viegli prognozējama, jo laika rinda parasti svārstās ap tendenci un nobīde no tās ir tipiska parādība. [7] Cikliskās svārstības ir svārstības, kas atkārtojas līdzigi kā sezonālās svārstības, taču tām nav noteikts laika intervāls pirms notikuma. [8] Nejaušas svārstības ir svārstības, kas neatspoguļo savu darbību nevienā no iepriekš minētajām komponentēm. Šādas svārstības nav prognozējamas un to cēloņi var būt dažādi.

Šajā nodaļā autors aprakstīs, kādas ir laika rindu analīzes un prognozēšanas metodes. Laika rindu analīze tiek veikta divu mērķu sasniegšanai – noderīgas informācijas iegūšanai (secinājumiem, novērojumiem) un nākamās laika rindas vienības vērtības prognozēšanai.

2.1 Stacionaritāte

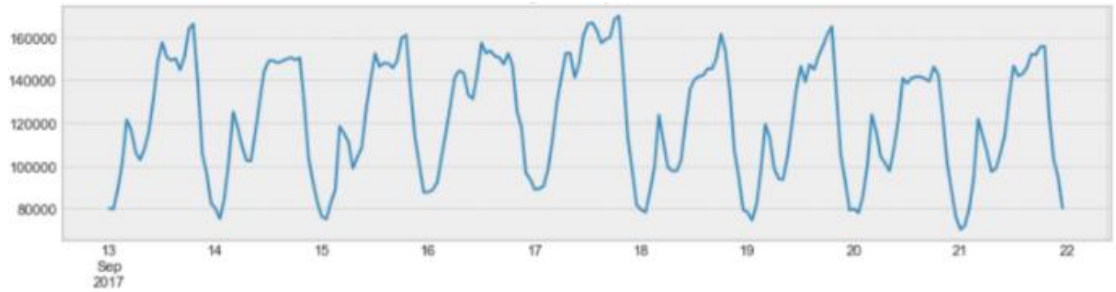
Stacionaritāte ir svarīga raksturiezīme laika rindās, kas nenozīmē to, ka laika rinda ir nemainīga, bet to, ka laika rindas statistiskās īpašības nemainās laikā, citos vārdos, veids, kā laika rinda mainās ir konstants. Dotais nosacījums ir svarīgs, jo stacionāras laika rindas prognozēt ir vieglāk, kā arī tikai stacionāras laika rindas var prognozēt ar kādu pieļaujamu precizitāti, jo nav iespējams paredzēt izmaiņas, kuras nav konstantas laikā. Dotā raksturiezīme jau no apraksta liecina par to, ka laika rindai vajadzētu būt paredzamai, ja veids, kā tā mainās ir paredzams. Viens no veidiem, ka noteikt vai laika rindas stacionaritāti ir izmantojot paplašināto Dickey-Fuller testu, kura nulles hipotēze balstās uz to, ka laika rinda nav stacionāra.

Reālajā dzīvē ne vienmēr laika rindas būs stacionāras, taču tās ir iespējams pārveidot par stacionārām izmantojot dažādas transformācijas un diferencēšanu.

2.1.1 Diferencēšana

Viens no veidiem, kā ne-stacionāru laika rindu pārveidot par stacionāru ir veicot diferencēšanu, kas tiek veikta no nākamās laika rindas vērtības atņemot iepriekšējo.

Piemēram laika rindai, kas sastāv no X_1, X_2, X_3, ..., X_n, diferencēšana izpaustos kā $X_2 - X_1$, $X_3 - X_2$, ..., $X_n - X_{n-1}$. Pirmās kārtas diferenci var aprēķināt tikai *n-1* (n apzīme laika rindas mērījumu skaitu) laika rindas mērījumiem, jo nav iespējams aprēķināt diferenci pirmajam mērījumam. Iespēja pastāv, ka veicot pirmās kārtas diferencēšanu laika rinda netiek pārveidota par stacionāru, tādā gadījumā to atkārto. Vairakkārtēja diferencēšana laika rindas analīzi sarežģī, tāpēc nepieciešams atrast vismazāko diferencēšanas kārtu ar kuru laika rinda ir veiksmīgi pārveidota par stacionāru. Otrās kārtas diferencēšanu veic izmantojot pēc pirmās kārtas diferencēšanas iegūtās laika rindas mērījumus un to var aprēķināt tikai *n-2* laika rindas mērījumiem. [9] Diferencēšana tiek pielietota ARIMA modelī, kas ir aprakstīts 2.4 nodaļā.



att. 2.1 ***Stacionāras laika rindas grafiskais attēlojums.***

Attēlā redzama stacionāra laika rinda, kurā novērojama dienu sezonalitāte.

2.2 Slīdošā vidējā modelis

Slīdošā vidējā (Angliski – Moving Average) modelis ir visvieglāk veicamais paņēmiens, taču arī *visnaivākais*, jo tā prognoze balstās uz to, ka katrs nākamais laika rindas mērījums būs iepriekšējo mērījumu, konkrēta intervāla, vidējais aritmētiskais.

Neskatoties uz to, ka dotais paņēmies ir viegli izdarāms, dažādos gadījumos tas var izrādīties salīdzinoši precīzs un var kalpot kā labs sākuma punkts turpmākajai analīzei, novērojot konkrētās laika rindas tendences. Slīdošā vidējā modeļa prognozes ir iespējams veikt tikai netālā nākotnē, jo kā tika minēts, dotais modelis vairāk attēlo laika rindas tendenci.

Matemātiski slīdošā vidējā prognozes aprēķina formula ar 3 laika vienību intervālu ir sekojoša:

$$F_n = \frac{A_{n-3} + A_{n-2} + A_{n-1}}{3}$$

Formulā F_n ir prognozētā vērtība un A_{n-*} ir reālā vērtība konkrētajā laika vienībā.

Izmantojot slīdošā vidējā analīzi ir iespējams laika rindu *nolīdzināt* no trokšņiem, tādā veidā uzskatāmāk redzot tās izmaiņu pamattendenci, jo to rēķinot tiek izslēgtas atsevišķas sākotnējās rindas līmeņa gadījuma svārstības. Izmantojot doto analīzi izvēlas datu intervālu, kuru izvēloties lielāku, tendence kļūs līdzenāka, taču intervāla garums nedrīkst pārsniegt kopējo laika rindas garumu.

tabula. 2.1 ***Gaisa temperatūra Igaunijā, Heltermā****. [4]*

Laiks	01.04.2019 00:00	01.04.2019 01:00	01.04.2019 02:00	01.04.2019 03:00	01.04.2019 04:00	01.04.2019 05:00	01.04.2019 06:00	01.04.2019 07:00	01.04.2019 08:00	01.04.2019 09:00	01.04.2019 10:00	01.04.2019 11:00	01.04.2019 12:00
Temperatūra (C°)	1,1	0,8	0,7	0,6	0,1	1,7	3,4	4,3	4,8	5,2	5,5	7,3	?

Pēc dotajiem datiem (skat. tabulu 2.1) var veikt slīdošā vidējā analīzi un prognozēt nākošās laika vienības temperatūras vērtību. Izvēloties intervālu ar 3 laika vienībām, prognozes aprēķins ir sekojošs:

T4 = (0,7 + 0,8 + 1,1) / 3 = 0,87

T5 = (0,6 + 0,7 + 0,8) / 3 = 0,70

T6 = (0,1 + 0,6 + 0,7) / 3 = 0,47

T7 = (1,7 + 0,1 + 0,6) / 3 = 0,80

T8 = (3,4 + 1,7 + 0,1) / 3 = 1,73

T9 = (4,3 + 3,4 + 1,7) / 3 = 3,13

T10 = (4,8 + 4,3 + 3,4) / 3 = 4,17

T11 = (5,2 + 4,8 + 3,4) / 3 = 4,77

T12 = (5,5 + 5,2 + 4,8) / 3 = 5,17

T13 = (7,3 + 5,5 + 5,2) / 3 = 6

Prognoze nākamajai laika vienībai ir vienāda slīdošā vidējā vērtību iepriekšējai, tātad 13. stundai jeb laika vienībai "01.04.2019 12:00" prognozētā vērtība ir 6. Pēc prognožu datiem var veikt to precizitātes analīzi salīdzinot prognozes ar novērotajiem datiem. Precizitātes analīzi vair veikt vairākos veidos, taču autors izpētīs 3 pazīstamākos:

- Vidējā absolūtā novirze,
- Vidējā kvadrātiskā novirze,
- Vidējā absolūtā procentuālā kļūda.

Vidējā absolūtā novirze (Angliski – Mean Absolute Deviation) atspoguļo kļūdu stap novērotajiem datiem un prognozētajiem datiem, un, jo mazāka novirze, jo precīzāka prognoze. Vidējās absolūtās novirzes matemātiskā formula:

$$MAD = \frac{1}{n}\sum_{i=1}^{n}|A_i - F_i|$$

Formulā n ir prognozēto vērtību skaits, A_i ir reālā vērtība konkrētajā laika vienībā no laika rindas un F_i ir prognozētā vērtība.

MAD = (|0,6 – 0,87| + |0,1 – 0,7| + |1,7 – 0,47| + |3,4 – 0,8| + |4,3 – 1,73| + |4,8 – 3,13| + |5,2 – 4,17| + |5,5 – 4,77| + |7,3 – 5,17|) / 9 = 1,43

Vīdējo kvadrātisko novirzi (Angliski – Mean Squared Error vai Mean Squared Deviation), līdzīgi kā vidējā absolūtā novirze, atspoguļo kļūdu starp novērotajiem datiem un prognozētajiem datiem, taču, atšķirībā no absolūtās novirzes, kvadrātiskā novirze lielākām kļūdām piešķir lielāku kļūdas koeficientu un mazākām kļūdām – mazāku koeficientu. To aprēķina pēc matemātiskās formulas:

$$MSD = \frac{1}{n}\sum_{i=1}^{n}(A_i - F_i)^2$$

Formulā n ir prognozēto vērtību skaits, A_i ir reālā vērtība konkrētajā laika vienībā no laika rindas un F_i ir prognozētā vērtība.

MSE = ((0,6 – 0,87)^2 + (0,1 – 0,7)^2 + (1,7 – 0,47)^2 + (3,4 – 0,8)^2 + (4,3 – 1,73)^2 + (4,8 – 3,13)^2 + (5,2 – 4,17)^2 + (5,5 – 4,77)^2 + (7,3 – 5,17)^2) / 9 = 2,69

Vidējā absolūtā procentuālā kļūda (Angliski – Mean Absolute Percentage Error) tiek bieži izmantota, jo kļūdas procentuālais attēlojums ir uzskatāmāks un bieži vien saprotamāks.

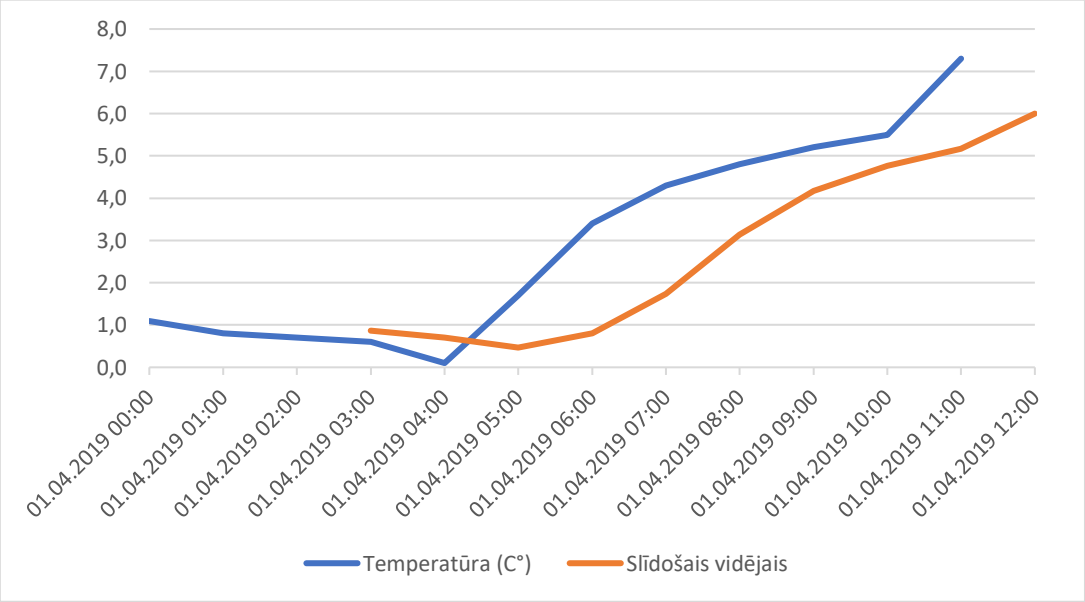
Neskatoties uz to, ka dotais analīzes paņēmiens tiek izmantots, tas rada problēmas situācijās, kad dati ir negatīvi vai pietuvināti nullei. Vidējās absolūtās procentuālās kļūdas aprēķina trūkums ir tāds, ka metode piešķir lielāku kļūdas koeficientu pozitīvām vērtībām, nekā negatīvām. [10] Matemātiskā formula dotās analīzes veikšanai ir sekojoša:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{A_i - F_i}{A_i}\right|$$

Formulā n ir prognozēto vērtību skaits, A_i ir reālā vērtība konkrētajā laika vienībā no laika rindas un F_i ir prognozētā vērtība.

MAPE = ((|0,6 – 0,87|) / 0,6 + (|0,1 – 0,7|) / 0,1 + (|1,7 – 0,47|) / 1,7 + (|3,4 – 0,8|) / 3,4 + (|4,3 – 1,73|) / 4,3 + (|4,8 – 3,13|) / 4,8 + (|5,2 – 4,17|) / 5,2 + (|5,5 – 4,77|) / 5,5 + (|7,3 – 5,17|) / 7,3) / 9 = 106%

Pēc veiktās analīzes iespējams secināt, ka laika rindas prognoze vairāk atspugoļo laika rindas tendenci un mazāk reālos datus, jo aprēķināto kļūdu vērtības ir salīzinoši lielas, priekš visu kopējo vērtību intervāla.



att. 2.2 ***Slīdošā vidējā analīzes prognožu grafiskais attēlojums.***

Grafikā redzams (skat. att. 2.2), kā slīdošā vidējā analīzes grafiskais attēlojums labāk attēlo laika rindas pamattendenci un nolīdzina trokšņus, kā, piemēram, pie laika vienības "01.04.2019 04:00".

tabula. 2.2 ***Slīdošā vidējā analīzes kļūdu rezultāti***

Laiks	Temperatūra (C°)	Slīdošais vidējais	MAD	MSE	MAPE
01.04.2019 00:00	1,10				
01.04.2019 01:00	0,80				
01.04.2019 02:00	0,70				
01.04.2019 03:00	0,60	0,87	0,27	0,07	44%
01.04.2019 04:00	0,10	0,70	0,60	0,36	600%
01.04.2019 05:00	1,70	0,47	1,23	1,52	73%
01.04.2019 06:00	3,40	0,80	2,60	6,76	76%
01.04.2019 07:00	4,30	1,73	2,57	6,59	60%
01.04.2019 08:00	4,80	3,13	1,67	2,78	35%
01.04.2019 09:00	5,20	4,17	1,03	1,07	20%
01.04.2019 10:00	5,50	4,77	0,73	0,54	13%
01.04.2019 11:00	7,30	5,17	2,13	4,55	29%
01.04.2019 12:00		6,00			
			1,43	**2,69**	**106%**

Tabulā (skat. tabulu 2.2) attēlotas slīdošā vidēja prognozes kļūdu aprēķins, kurā redzams, ka izmantojot slīdošā vidējā modeli nav iespējams aprēķināt prognozes datiem, kas atrodas pirmajā intervāla kopā.

2.3 Eksponenciālās izlīdzināšanas modelis

Eksponenciālās izlīdzināšanas (Angliski – Exponential Smoothing) modelis atšķirībā no slīdošā vidējā modeļa neizmanto tikai iepriekš novērotos mērījumus, bet arī iepriekšējo mērījumu prognozes, tādā veidā cenšoties izlabot novirzi no iepriekšējā mērījuma un prognozes. Dotajā modelī tiek izmantots *izlīdzināšanas koeficients*, kurš aktuālajiem datiem piešķir lielāku nozīmi prognozes aprēķināšanai. Līdzīgi kā slīdošā vidējā modelī, arī izmantojot eksponenciālās izlīdzināšanas modeli, laika rindu iespējams prognozēt tikai netālā nākotnē. Eksponenciālās izlīdzināšanas prognozes aprēķina pēc matemātiskās formulas:

$$F_n = F_{n-1} + \alpha(A_{n-1} - F_{n-1})$$

Formulā F_n ir prognozētā vērtība, F_{n-1} ir prognozētā vērtība iepriekšējai laika vienībai un A_{n-1} ir reālā vērtība iepriekšlaika vienībā.

Dotajā funkcijā α ir izlīdzināšanas koeficients, kurš atrodas robežās $0 < \alpha < 1$, kas nosaka prognozētās laika rindas līdzenumu un aktuālo datu ietekmi uz prognozi. Pietuvinot α vērtību tuvāk 0, laika rinda tiks vairāk izlīdzināta un aktuālo datu ietekme uz prognozi kļūs mazāka, pretēji, pietuvinot α vērtību tuvāk 1, aktuālo datu ietekme uz prognozi palielināsies. Prognozes precizitāti ietekmē izlīdzināšanas koeficienta izvēle un optimāla koeficienta atrašanu veic izpildot eksponenciālo izlīdzināšanu ar vairākiem koeficientiem, un tad salīdzinot to vidējās absolūtās, kvadrātiskās, un absolūtās procentuālas novirzes rezultātus, attiecīgi, izvēloties koeficientu, kura kļūda apstiprinājās kā mazākā. [11]

Izmantojot tabulas datus (skat. tabulu 2.1) un piemērojot 0.8 izlīdzināšanas koeficientu tiek veikta eksponenciālās izlīdzināšanas analīze un nākamās laika vienības prognozes aprēķins:

T1 = 1.1

T2 = 1.1 + 0.8 * (1.1 – 1.1) = 1.1

T3 = 1.1 + 0.8 * (0.8 – 1.1) = 0.86

T4 = 0.86 + 0.8 * (0.7 – 0.86) = 0.73

T5 = 0.73 + 0.8 * (0.6 – 0.73) = 0.63

T6 = 0.63 + 0.8 * (0.1 – 0.63) = 0.21

T7 = 0.21 + 0.8 * (1.7 – 0.21) = 1.4

T8 = 1.4 + 0.8 * (3.4 – 1.4) = 3

T9 = 3 + 0.8 * (4.3 – 3) = 4.04

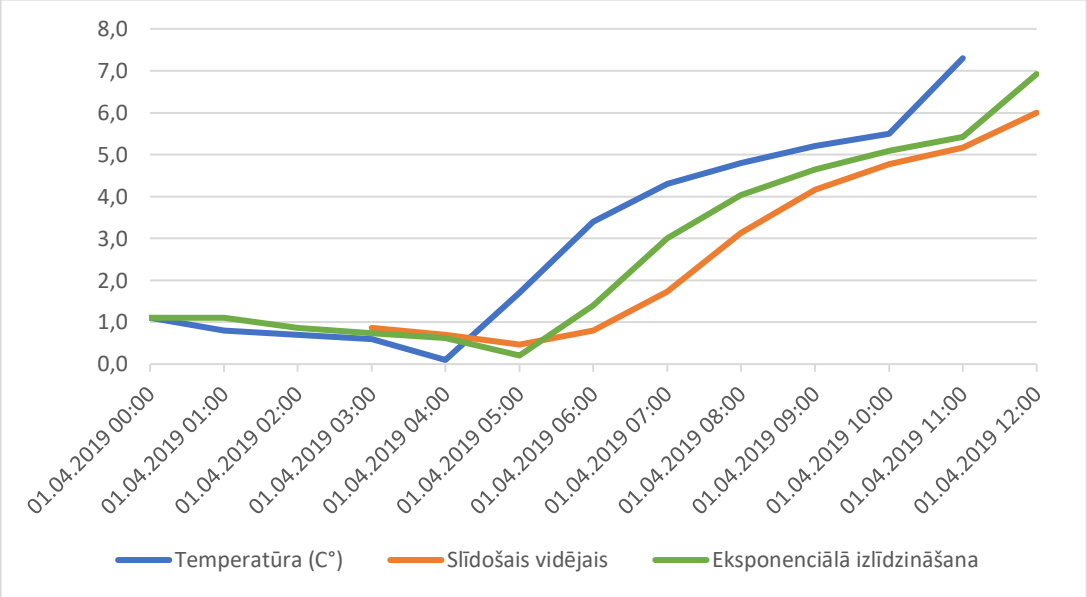
T10 = 4.04 + 0.8 * (4.8 – 4.04) = 4.65

T11 = 4.65 + 0.8 * (5.2 – 4.65) = 5.09

T12 = 5.09 + 0.8 * (5.5 – 5.09) = 5.42

T13 = 5.42 + 0.8 * (7.2 – 5.42) = 6.92

Prognozētā vērtība pēc eksponenciālās izlīdzināšanas aprēķina laika vienībai "01.04.2019 12:00" ir 6.92, kas salīdzinot ar slīdošā vidējā prognozi ir precīzāka. Arī aprēķina datus attēlojot diagrammā redzams, ka eksponenciālās izlīdzināšanas metode labāk reaģē uz temperatūras izmaiņām (skat. att. 2.3).



att. 2.3 ***Eksponenciālās izlīdzināšanas analīzes prognožu grafiskais attēlojums.***

Pec prognozes aprēķiniem tiek izmantotas iepriekš minēto prognožu kļūdu analīzes metodes.

MAD = (|1,1 – 1,1| + |0,8 – 1,1| + |0,7 – 0,86| + |0,6 – 0,73| + |0,1 – 0,63| + |1,7 – 0,21| + |3,4 – 1,4| + |4,3 – 3| + |4,8 – 4,04| + |5,2 – 4,65| + |5,5 – 5,09| + |7,3 – 5,42|) / 12 = 0,79

MSD = ((1,1 – 1,1)^2 + (0,8 – 1,1)^2 + (0,7 – 0,86)^2 + (0,6 – 0,73)^2 + (0,1 – 0,63)^2 + (1,7 – 0,21)^2 + (3,4 – 1,4)^2 + (4,3 – 3)^2 + (4,8 – 4,04)^2 + (5,2 – 4,65)^2 + (5,5 – 5,09)^2 + (7,3 – 5,42)^2) / 12 = 1,08

MAPE = (|1,1 – 1,1| / 1,1 + |0,8 – 1,1| / 0,8 + |0,7 – 0,86| / 0,7 + |0,6 – 0,73| / 0,6 + |0,1 – 0,63| / 0,1 + |1,7 – 0,21| / 1,7 + |3,4 – 1,4| / 3,4 + |4,3 – 3| / 4,3 + |4,8 – 4,04| / 4,8 + |5,2 – 4,65| / 5,2 + |5,5 – 5,09| / 5,5 + |7,3 – 5,42| / 7,3) / 12 = 70%

Pēc veiktajiem aprēķiniem, var secināt, ka dotās metodes prognozes ir precīzākas, ko pierāda visi trīs kļūdu analīzes rezultāti.

tabula. 2.3 ***Eksponenciālās izlīdzināšanas analīzes kļūdu rezultāti***

Laiks	Temperatūra (C°)	Eksponenciālā izlīdzināšana	MAD	MSE	MAPE
01.04.2019 00:00	1,10	1,10	0,00	0,00	0%
01.04.2019 01:00	0,80	1,10	0,30	0,09	38%
01.04.2019 02:00	0,70	0,86	0,16	0,03	23%
01.04.2019 03:00	0,60	0,73	0,13	0,02	22%
01.04.2019 04:00	0,10	0,63	0,53	0,28	526%
01.04.2019 05:00	1,70	0,21	1,49	2,23	88%
01.04.2019 06:00	3,40	1,40	2,00	4,00	59%
01.04.2019 07:00	4,30	3,00	1,30	1,69	30%
01.04.2019 08:00	4,80	4,04	0,76	0,58	16%
01.04.2019 09:00	5,20	4,65	0,55	0,30	11%
01.04.2019 10:00	5,50	5,09	0,41	0,17	7%
01.04.2019 11:00	7,30	5,42	1,88	3,54	26%
01.04.2019 12:00		6,92			
Koeficients	0,8		**0,79**	**1,08**	**70%**

Tabulā (skat. tabulu. 2.3) redzams, ka eksponenciālās izlīdzināšanas prognozes ir izdevušās precīzākās, salīdzinot ar slidošā vidējā prognozēm.

2.4 ARIMA modelis

Automātiski regresējošā integrētā slīdošā vidējā (Angliski – Autoregressive Integrated Moving Avarage) analīzes metode ir krietni komplicētāka metode, kuru izmanto laikrindas datu analīze un nākotnes vērtību prognozēšanā.

Atšķirībā no iepriekš apskatītajām analīzes metodēm, ARIMA metode ir efektīva tikai ar stacionārām laika rindām, tāpēc integrācijas daļā dotā metode laika rindu pārveido par stacionāru. ARIMA analīzes metode sastāv no trīs komponentēm:

- autoregresīvā analīze;
- integrācijas kārta, jeb differences kārta;
- slīdošā vidējā analīze.

Katra no šīm komponentēm modelī ir specificēta ar parametru un standarta pieraksts dotajam modelim ir ARIMA(p, d, q), kur

- p ir autoregresīvā modeļa kārta;
- d ir integrācijas kārta, kas pārveido laika rindu stacionāru;
- q ir slīdošā vidējā modeļā kārta.

Viens no veidiem, kā izvēlēties atbilsošas parametru vērtības, ir izmantojot Akaike informācijas kritēriju (Angliski – Akaike Information Criterion) un Bejiesa informācijas kritēriju (Angliski – Bayesian Information Criterion), kas tiek aprēķināts izvēlētajām modeļu vērtībām – izvēloties modeli, kam dotās vērtības apstiprinās kā mazākās. Otrs veids ir izmantojot daļējo autokorelāciju grafiku, kas attēlo korelāciju starp laika rindu un tās nobīdi.

3 LAIKA RINDU DATUBĀZES

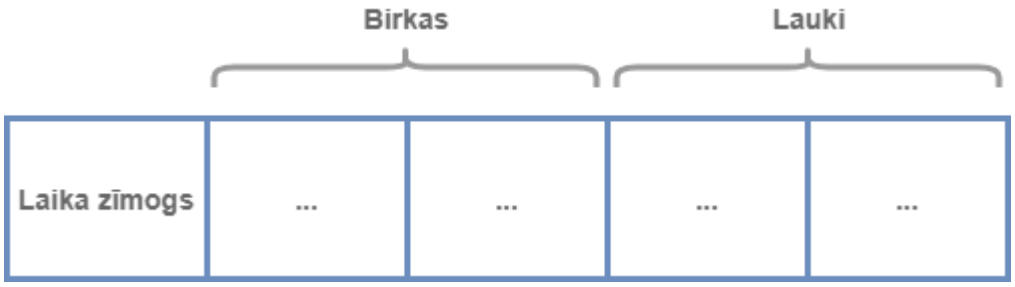
Pieejamo laika rindu datubāzu klāsts ir visai liels, taču, lai izvēlētos atbilstošu datubāzi laika rindas datiem nepieciešams paredzēt 3 lietas:

- Kāda ir lasīšanas/rakstīšanas attiecība un apjoms?
- Kada veida lasīšanas un rakstīšanas operācijas tiks izmantotas?
- Kāds būs sākotnējais datu apjoms un cik ātri tas varētu pieaugt? [12]

Šajā nodaļā autors apskatīs divas laika rindu datubāzes, kuras savā starpā atšķiras ar izmantoto modeli. Viena no tām ir InfluxDB, kas savā risinājumā izmanto birku kopas modeli un otra ir TimescaleDB, kas izmanto relāciju modeli.

3.1 InfluxDB

InfluxDB ir atvērtā pirmkoda laika rindu datubāze, kura ir optimizēta strādāšanai ar lielu rakstīšanas apjomu. [13] InfluxDB pašlaik ir vispopulārākā laika rindu datubāzu pārvaldības sistēma. [14] InfluxDB izmanto birku kopas (Angliski – Tagset) modeli, kur katram mērījumam ir laika zīmogs (Angliski – Timestamp), saistītās birkas, kuras apzīmē mērījumu metadatus un lauku kopa, kas satur mērījumu rezultātus.



att. 3.1 ***InfluxDB birku kopas modelis***

Lauku datu tipi ir limitēti ar peldošā punkta, veselo skaitļu, simbolu virkņu un loģiskā tipa mainīgajiem. [15] Birku kopas vērtības ir indeksētas un tās vērtības vienmēr tiek aptspoguļotas kā simbolu virkne, kas nevar tikt atjaunota. [16] InfluxDB vaicājumu veidošanai izmanto Flux skriptēšanas valodu.

InfluxDB piekršrocības:

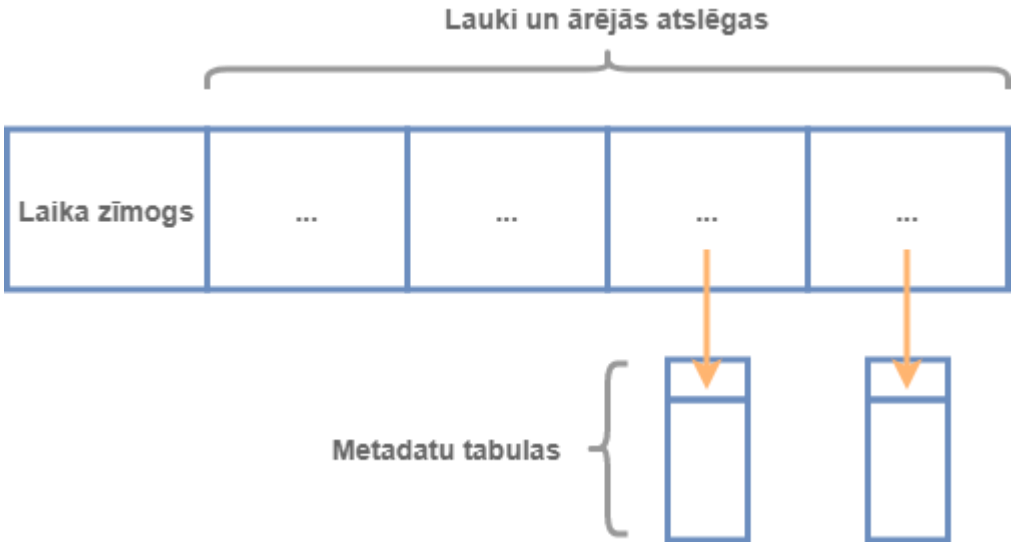
- viegla uzstādīšana, ja ievietojamie dati atbilst birku kopas modelim;
- ātrāka datu ievietošana pie mazas kardinalitātes, tas ir, katrā laika zīmogā tiek saglabāti tikai daži mērījumi;

3.2 TimescaleDB

TimescaleDB arī ir atvērtā pirmkoda laika rindu datubāze, taču atšķirībā no InfluxDB, dotā datubāze balstās uz SQL principiem. TimescaleDB ir izveidota kā papildinājums PostgreSQL, tā papildina zināmo relāciju datubāzi ar unikālu laika rindu operāciju kopumu, kas orientēts uz ātru datu saglabāšanu. TimescaleDB relāciju modelī katrs mērījums tiek ievietots savā rindā, kam seko laika vienība ar nenoteiktu skaitu laukiem, kuros saglabāt veiktos mērījumus.

TimescaleDB priekšrocības:

- iespējams ārējo atslēgu (Angliski – Foreign Key) pievienot jebkuram laukam, kura saista doto lauku ar citas tabulas datiem – papildus metadatiem;
- iespējams indeksēt jebkuru lauku;
- izmanto jau plaši zināmu strukturētu vaicājumvalodu (Angliski – Structured Query Language). [16]

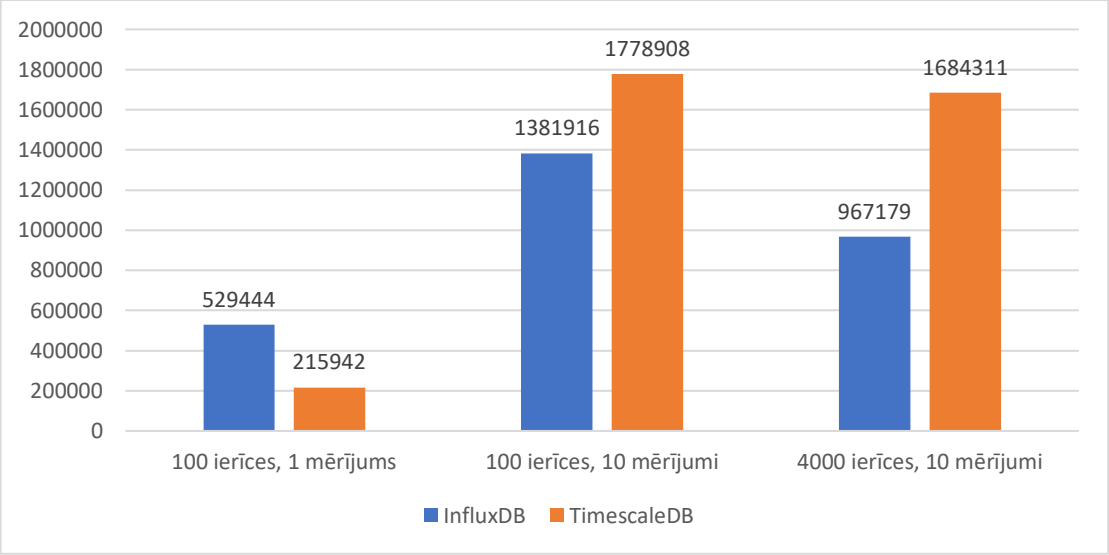


att. 3.2 ***TimescaleDB relāciju modelis***

Atšķirībā no InfluxDB, TimescaleDB atbalsta daudz plašāku klāstu ar datu tipiem. Dotajā datubāzē lauki var tikt definēti kā peldošā punkta, veselo skaitļu, simbolu virkņu, loģiskā tipa, masīvu, JSON lielo bināro objektu (Angliski – JavaScript Object Notation Binary Large Object), ģeotelpiskās dimensijas, datumu, laika, valūtu, bināro un citi datu tipi. [17]

3.3 Salīdzinājums

Neskatoties uz to, ka InfluxDB datubāze pašlaik ir vispopulārākā laika rindu datubāze, balstoties uz pieejamajiem avotiem autors uzskata, ka TimescaleDB datubāzei ir vairāk priekšrocību un tā ir pārāka par InfluxDB. TimescaleDB tiek izvirzīta kā piemērotākā datubāze, jo tā ir veidota uz PostgreSQL, kas ir zināma un pārbaudīta datubāze, un tas varētu liecināt par TimescaleDB datubāzes uzticamību. Kā arī dotā datubāze izmanto SQL vaicājumu veidošanai, kas ir plaši pazīstama vaicājumu valoda un tās lietotājiem nebūtu nepieciešams apgūt citu valodu, kā Flux, lai tos veidotu.



att. 3.3 ***Veiktspējas salīdzinājumā attēlots mērījumu ievietošanas skaits sekundē*** *[16]*

Datu ievietošanas veiktspējas salīdzinājumā pierādās tas, ka InfluxDB daudz labāk tiek galā ar lielu datu apjomu, kuriem ir maza kardinalitāte, taču palielinoties kardinalitātei, TimescaleDB iegūst pārsvaru (skat. att. 3.3).

4 PRAKTISKĀ DAĻA

Vadoties pēc darbā veiktā laika rindas pētījuma, darba ietvaros tika veikta Latvijas eksporta apjoma prognozēšana izmantojot iegūtos datus no 1995.gada janvāra līdz 2019.gada janvārim, kuri sadalīti ceturkšņos. Dati tika iegūti no Centrālās statistikas pārvaldes datubāzes. [18]

4.1 Tehniskais apraksts

Prognozes veikšanai tika izmantota Python programmēšanas valoda ar kuras palīdzību iespējams veikt ARIMA modeļa prognozes un tās grafiski attēlot. Risinājuma pamatā Python programmēšanas valoda tika izvēlēta tāpēc, ka tajā ir plašs klāst ar pieejamām statistikas bibliotēkām.

Izstrādātās programmatūras pirmkods ir bāzēts uz "Microsoft Azure Notebooks", kas ir lielisks tiešsaistes rīks ar kura palīdzību izstrādāto Python pirmkodu iespējams izpildīt pārlūkprogrammā. Izstrādātais programmatūras pirmkods ir pieejams tiešsaistē: https://exportanalysis-ricardskalnins.notebooks.azure.com/j/notebooks/Analysis.ipynb

4.2 Realizācijas apraksts

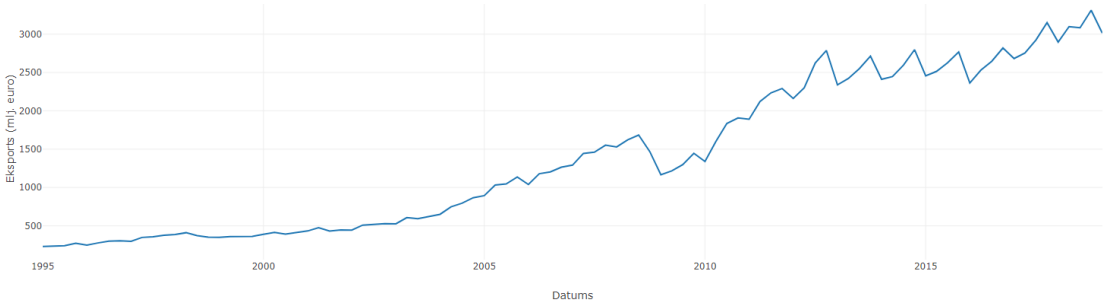
Iegūtā datu faila nolasīšanai tika izmantota Python "pandas" bibliotēka, ar kuras palīdzību tika importēti dati no "csv" tipa faila izmantojot "read_csv" funkciju. Pēc datu importēšanas to laika vienības kolona tika pārveidota "datetime" formātā un dotā kolona tika indeksēta.

tabula. 4.1 ***Datu pārveidojuma attēlojums***

GadsCeturksnis	Eksports
1995/1.ceturksnis	230.8
1995/2.ceturksnis	235.5
1995/3.ceturksnis	240.5
1995/4.ceturksnis	272.8
1996/1.ceturksnis	249.1
1996/2.ceturksnis	276.9
...	...

date	Eksports
1995-01-01	230.8
1995-04-01	235.5
1995-07-01	240.5
1995-10-01	272.8
1996-01-01	249.1
1996-04-01	276.9
...	...

Pārveidotie dati tika attēloti grafiski izmantojot “plotly” bibliotēku.

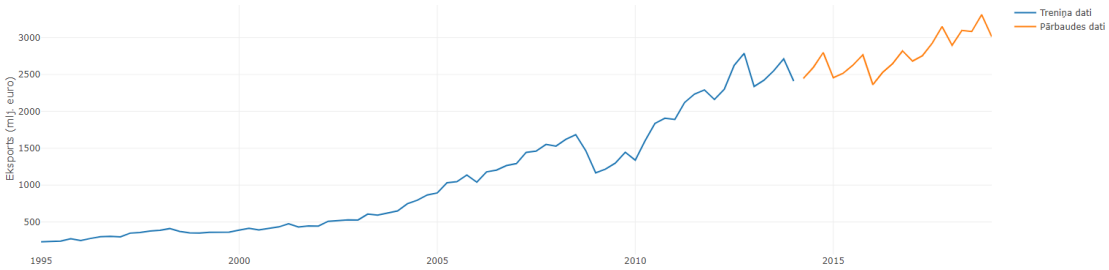


att. 4.1 ***Latvijas eksporta grafiskais attēlojums (1995-2019)***

Pēc grafika redzams (skat. att. 4.1), ka ir novērojama augšupejoša tendence. Iegūtie dati tika sadalīti divās daļās:

- modeļa trenēšanas datu kopa;
- prognozes pecizitātes pārbaudes datu kopa.

Modeļa trenēšanas datu kopa tiek izmantota prognozes veikšanā. Dotie dati tiek ievietoti “auto_arima” funkcijā, kurā pēc to analīzes tiek iegūtas labākās ARIMA(p, q, d) mainīgo vērtības pēc kurām tiek veiktas prognozes.

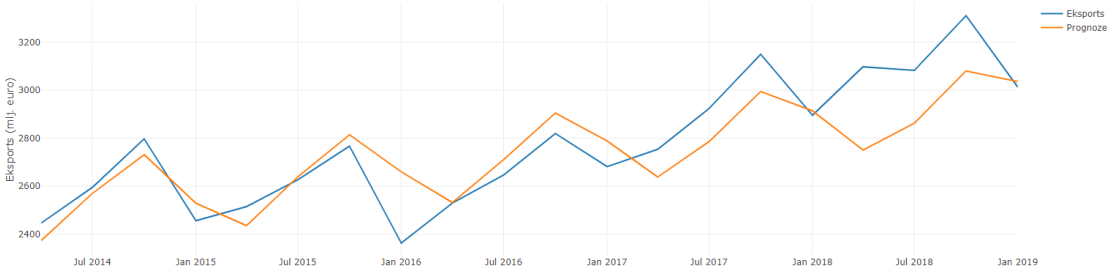


att. 4.2 ***Trenēšanas un pārbaudes datu kopu sadalījums***

Grafikā redzams modeļa trenēšanas datu kopas attēlojums ar zilo līniju un prognozes precizitātes pārbaudes datu kopa ar oranžo līniju (skat. att. 4.2). ARIMA modeļa uzdevums ir pēc iespējas precīzāk pietuvināties prognozes precizitātes pārbaudes datu kopai.

Izmantojot “auto_arima” funkciju, tiek inicializēts arima modelis, kurā iegūtas piemērotākās ARIMA modeļa mainīgo vērtības. Konkrētajiem datiem tās tiek atrastas kā p = 4, d = 1, q = 2. Dotās vērtības tālāk tiek izmantotas prognozes veikšanā.

Prognozes veikšanai tiek izmantota “predict” funkcija, kurai tiek padots prognozētā perioda garums, kas ir vienāds ar prognozes precizitātes pārbaudes datu kopas garumu. Iegūtie prognozes dati tiek attēloti grafikā kopā ar precizitātes pārbaudes datu kopu.



att. 4.3 ***Prognozes un pārbaudes datu kopu salīdzinājums***

Grafikā (skat. att. 4.3) zilā līnija attēlo precizitātes pārbaudes datu kopu un oranžā – prognožu kopu. Redzams, ka izveidotā ARIMA modeļa prognozes ir pietuvinātas reālajiem datiem, pēc kā jau varētu secināt, ka modelis ir izdevies veiksmīgs. Iegūtajiem prognozes datiem tiek aprēķināta arī vidējā absolūtā novirze, vidējā kvadrātiskā novirze un videjā absolūtā procentuālā kļūda.

Vidējā absolūtā un vidējā kvadrātiskā novirze datiem tika aprēķināta izmantojot "sklearn" bibliotēkas "mean_absolute_error", un "mean_squared_error" funkcijas. Vidējās absolūtās procentuālas kļūdas funkcija tika implementēta. Prognozes datu kļūdu analīzes rezultāti:

- MAD = 108.86
- MSE = 20751.33
- MAPE = 3.86%

REZULTĀTI

Bakalaura darbā tika veikta laika rindas datu analīzes un pielietojuma izpēte. Tika apskatītas tādas tehnoloģijas problemsituācijas, kā trūkstoši mērījumi, dublikāti mērījumi un aizkavēti mērījumi, un šo problēmsituāciju cēloņi, kā arī iespējamās darbības, kā no tām izvairīties. Tika apskatītas un praktiski pielietotas 3 klasiskas laika rindas datu analīzes un prognozēšanas metodes. Izpētītas arī divas vadošās laika rindu datubāzes, to tehniskās atsķirības un salīdzinājums, kā arī tika izvirzīta, pēc autora domām, piemērotākā datubāze laika rindu datiem.

Balstoties uz darba pētnieciskās daļas iegūtajiem rezultātiem tika veikta laika rindas datu prognozēšana, izmantojot Latvijas Eksporta apjoma datus un pielietojot petniecīskajā daļā aprakstītos laika rindu prognozēšānas metodes, kā arī veiktajām prognozēm tika pielietotas aprakstītas kļūdu analīzes metodes.

SECINĀJUMI

Darba mērķis bija izpētīt un noskaidrot laika rindas datu analīzes un prognozēšanas metodes, kā arī nozares, kurās šī tehnoloģija tiek veiksmīgi pielietota. Pēc pētījumā iegūto rezultātu analīzes, tika secināts, ka laika rindas dati gūst strauju popularitāti informācijas un tehnoloģijas nozarē, kurās laika rindas ir kā pamatkomponente to darbībā.

Laika rindu tehnoloģija ir jau sen plaši pielietota statistikā, taču mūsdienās laika rindas tiek pielietotas arī citās nozarēs, kā, piemēram, viedajās mājās un autonomajos transportlīdzekļos. Pateicoties šīm nozarēm, laika rindas ir guvušas lielu interesi par tās darbību un analīzi.

Darbā tika apskatītas 3 klasiskas laika rindu analīzes un prognozēšanas metodes, taču pēc pieejamās literatūras izpētes var secināt, ka mūsdienās tiek izstrādātas jaunas laika rindas prognozēšanas metodes, kā arī jau esošās metodes tiek papildinātas un atjaunotas, lai prognozes varētu veikt vēl efektīvāk un precīzāk.

Apskatot divas vadošās laika rindu datubāzes, autors izvīrzīja tieši TimescaleDB datubāzi, kā līderi šajā sfērā, pateicoties tās uzticamībai un izmantotajiem tehnoloģiskajiem risinājumiem.

Pēc autora domām, nozaru skaits, kurās tiek pielietotas laika rindas, turpinās augt, kā arī tiks veidotas jaunas prognozēšanas metodes, kuras spēs laika rindas prognozēt vēl precīzāk.

IZMANTOTĀ LITERATŪRA UN AVOTI

[1] «Whats is Time Series Data?,» MEMSQL, [Tiešsaiste]. Available: https://www.memsql.com/blog/what-is-time-series-data/.

[2] «DBMS popularity broken down by database model,» [Tiešsaiste]. Available: https://db-engines.com/en/ranking_categories.

[3] «Dinamikas rindas,» [Tiešsaiste]. Available: http://ezis.appspot.com/Statistika/d.14.htm.

[4] «Observation data,» Estonian Weather Service, [Tiešsaiste]. Available: https://www.ilmateenistus.ee/ilm/ilmavaatlused/vaatlusandmed/tunniandmed/?lang=en.

[5] «What is adaptive cruise control, and how does it work?,» ExtremeTech, [Tiešsaiste]. Available: https://www.extremetech.com/extreme/157172-what-is-adaptive-cruise-control-and-how-does-it-work.

[6] «MESH siltuma vadības platforma,» MESH, [Tiešsaiste]. Available: https://mesh.lv/.

[7] I. Zoratti, «Time series from collection to analysis - examples and use cases,» Percona Live, [Tiešsaiste]. Available: https://www.percona.com/live/data-performance-conference-2016/sessions/time-series-collection-analysis-examples-and-use-cases.

[8] «Cyclic and seasonal time series,» Rob J. Hyndman, [Tiešsaiste]. Available: https://robjhyndman.com/hyndsight/cyclicts/.

[9] R. J. Hyndman, «Forecasting: Principles and Practice - 8.1 Stationarity and differencing,» otexts.com, [Tiešsaiste]. Available: https://otexts.com/fpp2/stationarity.html.

[10] A. B. K. Rob J Hyndman, «Rob J. Hyndman,» [Tiešsaiste]. Available: https://robjhyndman.com/papers/mase.pdf.

[11] S. K. Paul, «Global Journals,» [Tiešsaiste]. Available: https://globaljournals.org/GJRE_Volume11/7-Determination-of-Exponential-Smoothing-Constant-to.pdf.

[12] T. Pifferi, «How to efficiently store and query time-series data,» Medium, [Tiešsaiste]. Available: https://medium.com/@neslinesli93/how-to-efficiently-store-and-query-time-series-data-90313ff0ec20.

[13] A. Solnichkin, «4 Best Time Series Databases To Watch in 2019,» Medium, [Tiešsaiste]. Available: https://medium.com/schkn/4-best-time-series-databases-to-watch-in-2019-ef1e89a72377.

[14] «DB-Engines Ranking of Time Series DBMS,» DB-Engines, [Tiešsaiste]. Available: https://db-engines.com/en/ranking/time+series+dbms.

[15] «InfluxDB Line Protocol reference,» influxdata, [Tiešsaiste]. Available: https://docs.influxdata.com/influxdb/v1.7/write_protocols/line_protocol_reference/.

[16] «TimescaleDB vs. InfluxDB: Purpose built differently for time-series data,» Timescale, [Tiešsaiste]. Available: https://blog.timescale.com/timescaledb-vs-influxdb-for-time-series-data-timescale-influx-sql-nosql-36489299877/.

[17] «Chapeter 8. Data Types,» PostgreSQL, [Tiešsaiste]. Available: https://www.postgresql.org/docs/current/datatype.html.

[18] «Eksports un imports pa valstu grupām pa ceturkšņiem (mlj. euro),» Centrālās statistikas pārvaldes datubāzes, [Tiešsaiste]. Available: http://data1.csb.gov.lv/pxweb/lv/atirdz/atirdz__atirdz__isterm/AT020c.px/.

PIELIKUMI

1 pielikums. Arima modeļa pirmkods.

```
# Izveido auto_arima modeli
model = auto_arima(train, error_action='ignore', suppress_warnings=True)

# Atrod vispiemērotākās p, q un d mainīgo vērtības
model.fit(train)

# Aprēķina prognozes vērtības testa kopas garumā
forecast = model.predict(n_periods=len(test))

# Precizitātes pārbaudes un prognozēto datu diagramma
def plot_comparision_data(data, test_data):
    export = go.Scatter(
        x = test_data.index,
        y = data["Eksports"],
        mode = 'lines',
        name = 'Eksports'
    )

    prediction = go.Scatter(
        x = test_data.index,
        y = data["Prediction"],
        mode = 'lines',
        name = 'Prognoze'
    )

    layout = dict(
        title = "Reālo un prognozēto datu salīdzinājums",
        xaxis = dict(title = 'Datums'),
        yaxis = dict(title = 'Eksports (mlj. euro)'),
    )

    fig = dict(data=[export, prediction], layout=layout)

    iplot(fig, filename='line-mode')

plot_comparision_data(comparision, test)
```

2 pielikums. Kļūdu analīzes pirmkods.

```
# Funkcija, kas aprēķina MAPE vērtību
def mean_absolute_percentage_error(actual, prediction):
    actual, prediction = np.array(actual), np.array(prediction)
    return np.mean(np.abs((actual - prediction) / actual)) * 100

forecast = pd.DataFrame(forecast, index = test.index, columns=["Prediction"])
comparision = pd.concat([test, forecast], axis=1)

# Aprēķina MAD, MSE un MAPE vērtības
def performance_measure(test, forecast):
    forecast_errors = test.copy()["Eksports"].values.tolist()

    mad = mean_absolute_error(test["Eksports"].values.tolist(),
forecast["Prediction"].values.tolist())
    mse = mean_squared_error(test["Eksports"].values.tolist(),
forecast["Prediction"].values.tolist())
    mape = mean_absolute_percentage_error(test["Eksports"].values.tolist(),
forecast["Prediction"].values.tolist())

    dataframe = pd.DataFrame(data = {
        "MAD": [mad],
        "MSE": [mse],
        "MAPE": [mape]
    })

    dataframe.reset_index(drop=True, inplace=True)

    display(dataframe)
```

Bakalaura darbs „Laika rindas datu analīze un pielietojums” izstrādāts LU Datorikas fakultātē.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: _____ Ričards Kalniņš __.__.2019

Rekomendēju/nerekomendēju darbu aizstāvēšanai *(nederīgo svītro vadītājs)*

Vadītājs: Dr. Sc. Comp. profesors Māris Vītiņš _____ __.__.2019.

Recenzents: Dr. Sc. Comp. profesors Ģirts Karnītis

Darbs iesniegts Datorikas fakultātē __.__.2019.

Dekāna pilnvarotā persona: vecākā metodiķe Ārija Sproģe ____ _____

Darbs aizstāvēts bakalaura gala pārbaudījuma komisijas sēdē

__.__.2019. prot. Nr. _____

Komisijas sekretārs(-e): _____