

**RĪGAS TEHNISKĀ UNIVERSITĀTE**

**Sergejs PARŠUTINS**

**INTELEKTUĀLU AĢENTU TEHNOLOĢIJĀ UN DATU IEGUVES  
METODĒS BĀZĒTAS PRODUKTA DZĪVES CIKLA VADĪBAS  
ATBALSTA SISTĒMAS IZSTRĀDE**

**Promocijas darba kopsavilkums**

**Rīga 2011**

**RĪGAS TEHNISKĀ UNIVERSITĀTE**  
Datorzinātnes un informācijas tehnoloģijas fakultāte  
Informācijas tehnoloģijas institūts

**Sergejs PARŠUTINS**

Doktora studiju programmas „Vadības informācijas tehnoloģija” doktorants

**INTELEKTUĀLU AGENTU TEHNOLOĢIJĀ UN DATU IEGUVES  
METODĒS BĀZĒTAS PRODUKTA DZĪVES CIKLA VADĪBAS  
ATBALSTA SISTĒMAS IZSTRĀDE**

**Promocijas darba kopsavilkums**

Zinātniskais vadītājs  
Dr.habil.sc.comp., profesors  
**A.BORISOVS**

**Rīga 2011**

UDK 004.89(043.2)

Pa 607 i

Paršutins S. Intelektuālu aģentu tehnoloģijā un datu ieguves metodēs bāzētas produkta dzīves cikla vadības atbalsta sistēmas izstrāde. Promocijas darba kopsavilkums.-R.: RTU, 2011.-36 lpp.

Iespiests saskaņā ar RTU promocijas padomes P-07 2011. gada 26. maija lēmumu Nr. 11-02.

**ISBN: 978-9934-10-169-4**



Šis darbs izstrādāts ar Eiropas Sociālā fonda atbalstu projektā „Atbalsts RTU doktora studiju īstenošanai”

## PROMOCIJAS DARBS

### IZVIRZĪTS RĪGAS TEHNISKAJĀ UNIVERSITĀTĒ INŽENIERZINĀTŅU DOKTORA GRĀDA IEGŪŠANAI

Promocijas darbs inženierzinātņu doktora grāda iegūšanai tiek publiski aizstāvēts 2011. gada 5. septembrī Rīgas Tehniskās universitātes Datorzinātnes un informācijas tehnoloģijas fakultātē, Meža ielā 1, 3. korpusā, 202. auditorijā.

#### OFICIĀLIE RECENZENTI

Profesors, Dr.habil.sc.ing. Jānis Grundspenķis  
Rīgas Tehniskā universitāte, Latvija

Profesors, Dr.sc.ing. Oļegs Užga-Rebrovs  
Rēzeknes augstskola, Latvija

Profesors, DSc Mihails Kovaļov  
Baltkrievijas universitātes Lietišķas matemātikas un informātikas fakultāte,  
Informātikas problēmu apvienotais institūts, Baltkrievija

#### APSTIPRINĀJUMS

Es apstiprinu, ka esmu izstrādājis doto promocijas darbu, kas iesniegts izskatīšanai Rīgas Tehniskajā universitātē inženierzinātņu doktora grāda iegūšanai. Promocijas darbs nav iesniegts nevienā citā universitātē zinātniskā grāda iegūšanai.

Sergejs Paršutins .....(Paraksts)

Datums: .....

Promocijas darbs ir uzrakstīts latviešu valodā, satur ievadu, 5 nodaļas, rezultātu analīzi un secinājumus, bibliogrāfisko sarakstu, 5 pielikumus, 47 attēlus, kopā 131 lappuse. Bibliogrāfiskajā sarakstā ir 61 nosaukums.

# VISPĀRĒJS DARBA RAKSTUROJUMS

## Tēmas aktualitāte

Nepārtraukti attīstoties mūsdienīgas informācijas tehnoloģijas piedāvā plašākas iespējas vadības sfērā. Uzņēmumi tiecas ieviest dažādas vadības sistēmas, cenšoties palielināt plānošanas un ražošanas efektivitāti kā arī samazināt izdevumus. Viena no svarīgām problēmām uzņēmumos, kas ražo un pozicionē tirgū dažādas preces ir plānošana, kas ir produkta dzīves cikla vadības neatņemama daļa. Šī problēma paliek aktuāla, jo produkta dzīves cikls ietekmē ražošanas plānošanas stratēģijas izvēli un uzņēmuma izdevumus, tajā skaitā reklāmas kompānijām atsevišķos laika periodos.

Produkta dzīves cikla vadība ir stratēģiskā procesu vadības pieeja ar dažādu procesu izpildes nodrošināšanu produkta dzīves ciklā saistītu problēmu risināšanai un to novēršanai nākotnē [40, 43]. Produkta dzīves cikls (PDC) attēlo produkta pieprasījuma līkni laika periodā no produkta ieviešanas tirgū līdz produkta izņemšanai no ražošanas un no tirgus. PDC ietver vairākas fāzes, kas atšķiras ar pieprasījuma līmeni, tā izmaiņu tendenci un ar nepieciešamu ieguldījumu līmeni. Bieži pielietojamās metodes PDC fāžu robežu noteikšanai balstās uz pieprasījuma līknes prognozēšanas un eksperta viedokļa izejot no prognozētā pieprasījuma.

Produkta dzīves cikla fāžu robežu noteikšanas uzdevums nav jauns, bet joprojām paliek aktuāls, pasaules zinātnieki to pēta un piedāvā dažādus iespējamus risinājumus. Vairākums no šiem risinājumiem palīdz atbildēt uz jautājumu „Vai produkts mainīja PDC fāzi?“, un neatbild uz jautājumu „Kad notiks fāzes pārejas periods?“. Datu ieguves metodes var sniegt nepieciešamo funkcionalitāti zināšanu bāzes izveidošanai izmantojot statistikas datus, kura attēlos saites starp produkta pieprasījuma līkni un dzīves cikla fāzes pārejas periodu. Intelektuālu aģentu tehnoloģija piedāvā plašas iespējas sarežģītu sistēmu projektēšanai, un kopā ar datu ieguves metodēm var būt pielietota PDC vadības atbalsta multiaģentu sistēmas izstrādei, kas automātiskā režīmā veidos un atjaunos zināšanu bāzi, uzraudzīs vairāku produktu stāvokli tirgū un savlaicīgi sniegs menedžerim informāciju par PDC fāzes pārejas periodiem, arī gadījumos, kad ir pieejamas tikai dažas pirmās vērtības pieprasījuma līknē.

## Darba mērķis

**Promocijas darba mērķis** ir izpētīt īsu laika rindu klasterizācijas metodes un uz intelektuālu aģentu tehnoloģijas un datu ieguves metožu bāzes izstrādāt multiaģentu sistēmas modeli un metodes produkta dzīves cikla fāzes pārejas perioda prognozēšanai. Darba mērķa sasniegšanai nepieciešams atrisināt šādus uzdevumus:

1. Izpētīt intelektuālu aģentu tehnoloģijas idejas un pamatprincipus, un formulēt pētījuma priekšmetam piemērotas aģentu un multiaģentu sistēmas definīcijas.
2. Izstrādāt multiaģentu sistēmas modeli, kas automātiski izveidotu saišu modeli starp PDC ieviešanas fāzes pārejas periodu un produkta pieprasījuma līkni, un nodrošinātu pārejas perioda prognozēšanu jauniem datiem izmantojot saišu modeli.
3. Izstrādāt un aprakstīt darbības algoritmus multiaģentu sistēmā iekļautajiem aģentiem, nodrošinot aģentiem spēju apmainīties ar informāciju.
4. Izpētīt īsu laika rindu klasterizācijas metodes un izstrādāt modifikācijas atlasītajām klasterizācijas metodēm, kas nodrošinātu īsu laika rindu ar atšķirīgu periodu skaitu klasterizāciju.
5. Aprobēt izstrādātu multiaģentu sistēmu, izveidojot programmatūru produkta dzīves cikla ieviešanas fāzes pārejas perioda prognozēšanai, implementējot katru no izstrādātajām klasterizācijas metožu modifikācijām. Izpildīt iegūto praktisko rezultātu salīdzinošo analīzi.

6. Izpētīt datu pirmapstrādes pamatprincipus un izpildīt darbā izmantojamo datu sagatavošanu analīzei.

### **Pētījuma priekšmets**

Pētījuma priekšmets ir intelektuālu aģentu tehnoloģijas un datu ieguves metožu apvienošanas un modifikācijas stratēģijas, virzītas uz uzdevuma risināšanu apstākļos, kur tehnoloģijas vai metodes atsevišķa pielietošana nebūtu pietiekoši efektīva.

### **Pētījuma hipotēzes**

1. Intelektuālu aģentu tehnoloģijas izmantošana var vienkāršot sarežģītu vadības un vadības atbalsta sistēmu projektēšanu, piedāvājot plašas iespējas struktūras, procesu un informācijas plūsmu formalizācijai.
2. Pielietojot datu ieguves metodes, ir iespējams izveidot produkta dzīves cikla paraugu modeli, paredzētu jaunu produktu analīzei.

### **Pētījumu metodes**

Promocijas darbā ir izmantotas intelektuālu aģentu tehnoloģija, datu ieguves (*Data Mining*) un nestingras skaitļošanas (*Soft Computing*) metodes un to modifikācijas, informācijas teorija, ieskaitot pašorganizējošus neironu tīklus, hierarhisko klasterizācijas algoritmu uz gravitācijas likuma pamata, datu pirmapstrādes metodes un attāluma mērus: *Dynamic Time Warping (DTW)* un darbā piedāvātais *MEuclidean* attāluma mērs.

### **Zinātniskais jaunieguvums**

1. Izstrādāta PDC fāzes pārejas perioda prognozēšanas pieeja uz intelektuālu aģentu tehnoloģijas un datu ieguves metožu bāzes, kura tika realizēta darbā piedāvātajā multiaģentu sistēmā.
2. Piedāvāta pašorganizējošo neironu tīklu algoritma modifikācija, kas ļauj ar vienu neironu tīklu klasterizēt īsas laika rindas ar atšķirīgu periodu skaitu.
3. Izstrādāta hierarhiskā gravitācijas klasterizācijas algoritma modifikācija, kas ļauj klasterizēt īsas laika rindas ar atšķirīgu periodu skaitu.
4. Piedāvāta *Eiklīda* attāluma mēra modifikācija – *MEuclidean* attāluma mērs, pielietojot kuru ir iespējams aprēķināt attālumu starp laika rindām ar atšķirīgu garumu.

### **Darba praktiskais pielietojums**

Darbā tika izpētītas intelektuālo aģentu tehnoloģiju idejas un pamatprincipi, un kombinācijā ar modificētām klasterizācijas metodēm tika pielietotas produkta dzīves cikla pārvaldīšanas atbalsta multiaģentu sistēmas izstrādei. Izstrādātā sistēma tika pielietota PDC fāzes pārejas perioda prognozēšanas uzdevuma risināšanai, pielietojot datus no starptautiskas ķīmiskas ražošanas kompānijas *HUNTSMAN*, iegūtus starptautiskā projekta *ECLIPS* ietvaros. Sistēmas veiktspēja tika pārbaudīta, izmantojot katru no izstrādātajām klasterizācijas algoritmu modifikācijām.

Sistēmas darbības rezultāts ir automātiski ģenerējama zināšanu bāze, kas satur saites starp produkta pieprasījuma līkni un pārejas periodu. Izmantojot automātiski ģenerēto zināšanu bāzi, jauniem produktiem tika veikta PDC fāzes pārejas perioda prognoze. Labākie rezultāti tika iegūti, pielietojot multiaģentu sistēmu ar modificēto hierarhisko gravitācijas klasterizācijas algoritmu. Praktisko eksperimentu realizācijai ar mērķi novērtēt sistēmas veiktspēju tika izstrādāts specializēts programnodrošinājums *MS Visual Basic 2010* vidē.

Par zinātniskiem rezultātiem tika ziņots šādās konferencēs:

1. Parshutin S. *Managing product life cycle with multiagent data mining system // Industrial Conference on Data Mining, ICDM'10 Germany, Berlin, July 12-14, 2010.*
2. Parshutin S., Borisov A. *Data Mining Driven Decision Support // 16<sup>th</sup> International Multi-Conference ADVANCED COMPUTER SYSTEMS, ACS'2009, Poland, Miedzydroje, October 14-16, 2009.*
3. Parshutin S., Aleksejeva L., Borisov A. *Forecasting Product Life Cycle Phase Transition Points with Modular Neural Networks Based System // Industrial Conference on Data Mining, ICDM'09, Germany, Leipzig, July 20-22, 2009.*
4. Parshutin S., Aleksejeva L., Borisov A. *Time Series Analysis with Modular Neural Networks // RTU 49<sup>th</sup> International Scientific Conference, Latvia, Riga, October 13, 2008.*
5. Parshutin S., Kuleshova G. *Time Warping Techniques in Clustering Time Series // 14th International Conference on Soft Computing, MENDEL 2008, Czech Republic, Brno, June 18-20, 2008.*
6. Parshutin S. *Clustering Time Series of Different Length Using Self-Organising Maps // RTU 48<sup>th</sup> International Scientific Conference, Latvia, Riga, October, 2007.*
7. Паршутин С.В. *Кластеризация временных рядов с применением карт самоорганизации // Международная конференция «Интегрированные модели и мягкие вычисления в искусственном интеллекте» (IMSCAI'2007) – 28-30 мая, Коломна, Россия, 2007.*

### **Promocijas darba galvenie rezultāti**

Promocijas darbā izpētītas intelektuālu aģentu tehnoloģija un datu ieguves metodes, izstrādāta un aprobēta produkta dzīves cikla vadības atbalsta multiaģentu sistēma. Detalizēti darba galvenie rezultāti:

1. Izstrādāta produktu dzīves cikla vadības atbalsta multiaģentu sistēma, kas nodrošina sistēmas lietotāju ar papildus informāciju par produkcijas stāvokli tirgū un par situācijas iespējamiem attīstības virzieniem, kas veicina argumentētu lēmumu pieņemšanu par ražošanas plānošanas, tirdzniecības attīstīšanas un reklāmas stratēģijām. No iegūtiem rezultātiem var secināt, ka apvienojot intelektuālo aģentu tehnoloģiju un datu ieguves metodes ir iespējams izstrādāt sarežģītas vadības un vadības atbalsta sistēmas.
2. Izpildīta intelektuālu aģentu tehnoloģijas pamatprincipu analīze, kuras rezultāti norāda uz to, ka intelektuālu aģentu tehnoloģijas pielietošana var vienkāršot sarežģītu, sadalītu sistēmu modeļa projektēšanas procesu.
3. Balstoties uz promocijas darbā definētiem uzdevumiem un izteiktām prasībām multiaģentu sistēmai, izstrādāti un aprakstīti funkcionēšanas algoritmi piedāvātās sistēmas aģentiem – datu vadības aģents, datu ieguves aģents un lēmumu atbalsta aģents. Rezultāti ļauj secināt, ka intelektuālo aģentu tehnoloģija veicina sistēmas iekšējo procesu precīzāku formalizāciju un strukturēšanu.
4. Izpildīta vairāku klasterizācijas algoritmu piemērotības analīze darbā risināmajam uzdevumam. Rezultātā atlasīti un aprakstīti pašorganizējošo neironu tīklu un hierarhiskā gravitācijas klasterizācijas algoritms. Izpildīta minēto algoritmu pielietojamības analīze īsu laika rindu ar atšķirīgu periodu skaitu klasterizācijai. No iegūtiem rezultātiem var secināt, ka pašorganizējošo neironu tīklu un hierarhiskā

gravitācijas klasterizācijas algoritma klasiskie varianti nenodrošina iespēju klasterizēt laika rindas ar atšķirīgu garumu.

5. Izstrādātas pašorganizējošo neironu tīklu algoritma un hierarhiskā gravitācijas klasterizācijas algoritma modifikācijas, kas nodrošina īsu laika rindu ar atšķirīgu periodu skaitu klasterizācijas iespēju.
6. Piedāvāts *MEuclidean* attāluma mērs, pielietojot kuru ir iespējams aprēķināt attālumu starp laika rindām ar atšķirīgu periodu skaitu.
7. Izstrādāts programmnodrošinājums, kas realizē promocijas darbā piedāvātās multiaģentu sistēmas konceptuālo un procesuālo struktūru, un implementē izstrādātās klasterizācijas algoritmu modificētās versijas.
8. Ar izstrādātā programmnodrošinājuma palīdzību izpildīta piedāvātās multiaģentu sistēmas aprobācija un salīdzinošā analīze, salīdzinot izstrādātās klasterizācijas algoritmu modifikācijas un attālumu mērus. Sistēmas veiktspējas novērtēšanas rezultāti parādīja, ka piedāvātā multiaģentu sistēma ir spējīga automātiskā režīmā izveidot datus pastāvošo saišu modeli un pielietot to mērķa atribūtu vērtības prognozēšanai jauniem datiem ar salīdzinoši augstu precizitāti.
9. Balstoties uz promocijas darbā piedāvātās multiaģentu sistēmas aprobācijas un veiktspējas novērtēšanas rezultātu analīzes secināts, ka pielietojot datu ieguves metodes ir iespējams izveidot produkta dzīves cikla paraugu modeli, paredzētu jaunu produktu dzīves cikla analīzei.

Promocijas darba izstrādes gaitā iegūtie zinātniskie un praktiskie rezultāti tika izmantoti šādos projektos:

1. Latvijas-Baltkrievijas sadarbības programma zinātnē un tehnikā, līgums Nr. L7631, „Medicīnisko un bioloģisko datu intelektuālo metožu un apstrādes algoritmu kompleksa izstrāde onkoloģisko slimību diagnostikas pilnveidošanai” (2010. – 2011. gadi).
2. LZP grants Nr. 09.1564, “Imitācijas modelēšanas un skaitļošanas intelekta metodes loģistikas un elektronisko pakalpojumu optimizācijai” (2010. – 2012. gadi).
3. LZP grants Nr. 09.1201, “Imitācijas modelēšanā bāzēta optimizācija, pielietojot skaitļošanas intelektu” (2009.g.).
4. RTU pētniecības projekts Nr. ZP-2008/7 „Mākslīgā intelekta skaitliskās metodes klasifikācijas uzdevumos” (2008. – 2009. gadi).
5. RTU pētniecības projekts Nr. ZP-2007/05 „Mākslīgā intelekta skaitliskās metodes datu ieguves uzdevumos” (2007. – 2008. gadi).
6. *European Project ECLIPS* (Extended Collaborative Integrated Life Cycle Supply Chain Planning System) of the European Community Sixth Framework Programme (2006. - 2008. gadi).
7. LZP grants nr. 05.1639, “Intelektuālās datortehnoloģijas slikti formalizējamiem lēmumu pieņemšanas uzdevumiem” (2005. – 2008. gadi).
8. IZM - RTU pētniecības projekts R 7085 “Mākslīgais intelekts prognozēšanas un vadības uzdevumos” (2006. gads).

## **Publikācijas**

Pētījumu rezultāti ir publicēti 11 zinātniskajos rakstos. Publikāciju saraksts ir iekļauts kopējā promocijas darbā izmantojamās literatūras sarakstā, kas atrodas promocijas darba

kopsavilkuma beigās. Pētījumi veikti ar Eiropas Sociālā fonda atbalstu Nacionālās programmas „Atbalsts doktorantūras programmas īstenošanai un pēcdoktorantūras pētījumiem” projekta „Atbalsts RTU doktorantūras attīstībai” ietvaros.

## **Darba struktūra un apjoms**

Promocijas darbs satur ievadu, 5 nodaļas, secinājumus, bibliogrāfisko sarakstu, 5 pielikumus, 47 attēlus, kopā 131 lappuse. Literatūras sarakstā ir 61 nosaukums. Promocijas darba struktūra ir šāda:

IEVADS – satur vispārējo informāciju par pētījuma priekšmetu, definē darba mērķi un saistītos uzdevumus.

1. NODAĻĀ ir sniegts ievads pētījumu nozarē, nozīmēta pētījumu sfērā risināmā problēma un tiek pētīti un aprakstīti intelektuālo aģentu tehnoloģijas pamatprincipi. Ir sniegts ieskats par aģentu īpašībām, multiaģentu sistēmas tiek salīdzinātas ar tādām zinātnēm, kā „Sadalītas un paralēlas sistēmas”, „Mākslīgais intelekts”, „Ekonomika un Spēļu teorija” un „Sociālā zinātne”. Noslēgumā definēta uzdevuma formalizēta nostādne.
2. NODAĻĀ tiek motivēta intelektuālo aģentu tehnoloģijas izmantošana promocijas darbā definēto uzdevumu risināšanai. Tiek piedāvāts produkta dzīves cikla vadības atbalsta multiaģentu sistēmas modelis, ir aprakstīta tā struktūra un īpašības, katram aģentam ir aprakstītas tā koncepcijas un funkcionēšanas algoritms.
3. NODAĻĀ ir analizēti darbā izmantotie klasterizācijas algoritmi – pašorganizējošie neironu tīkli un hierarhiskais gravitācijas klasterizācijas algoritms. Ir aprakstītas promocijas darbā izstrādātas klasterizācijas algoritmu modifikācijas, kas paplašina klasisko algoritmu iespējas ar pielietojumu īsu laika rindu ar atšķirīgu periodu skaitu klasterizācijai. Nodaļā ir sniegta informācija par izmantotiem un piedāvātiem attāluma mēriem un par datu pirmapstrādes procesu.
4. NODAĻĀ ir veltīta piedāvātās sistēmas aprobācijas un eksperimentu realizācijas nolūkam izstrādātam programmnodrošinājumam, aprakstīta tā funkcionalitāte, lietotāja saskarne.
5. NODAĻĀ notiek izstrādātās multiaģentu sistēmas eksperimentālā pārbaude. Sākumā tiek aprakstītas izmantotās datu kopas. Tam seko sistēmas veikspējas analīze, balstoties uz iegūtās apmācības kļūdas. Balstoties uz iegūtajiem rezultātiem, tiek izteikta un aprakstīta sistēmas drošas pārbaudes stratēģija. Tiek izstrādāts nepieciešamo eksperimentu plāns, kura izpildes rezultāti ir apkopoti un analizēti nodaļas nobeigumā. Nodaļā ir sniegti kopējie sistēmas veikspējas novērtēšanas rezultāti.

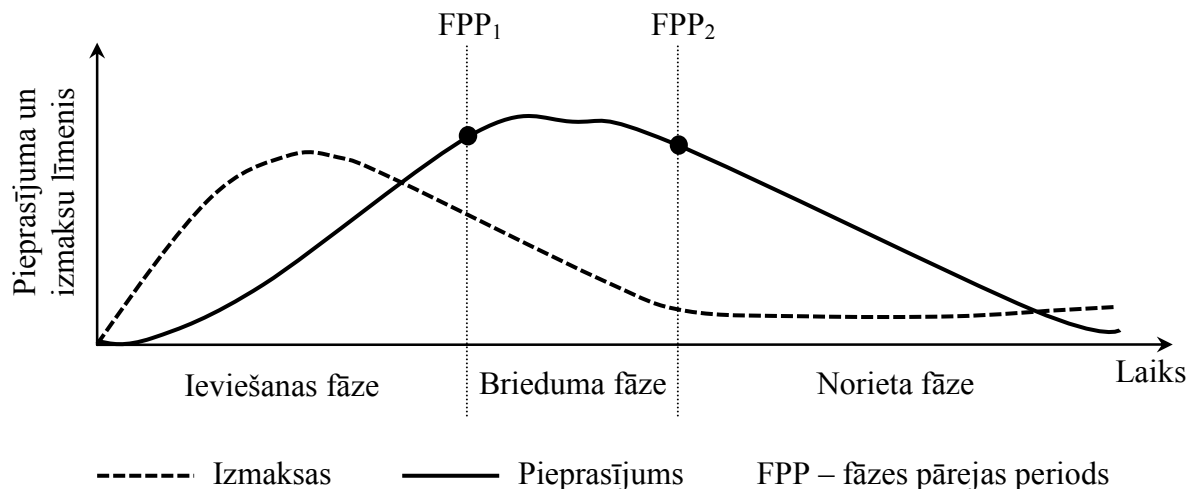
## **DARBA ATSEVIŠĶU NODAĻU IZKLĀSTS**

### **Pirmā nodaļa**

#### **1.1. Ievads pētījumu nozarē**

Produkta dzīves cikla vadība var būt definēta kā stratēģiskā procesu vadības pieeja ar dažādu procesu izpildes nodrošināšanu produkta dzīves ciklā saistītu problēmu risināšanai un to novēršanai nākotnē [11, 25, 41, 44]. Produkta dzīves cikla vadība apvieno sevī vairākus uzdevumus, kuru saraksts ir cieši saistīts ar uzņēmuma prasībām, kurā izmanto vai plāno ieviest produkta dzīves cikla vadības sistēmu [41, 44]. Promocijas darbā piedāvātā multiaģentu sistēma ir virzīta tieši uz prognozēšanas un plānošanas procesa atbalstu, kura pamata uzdevums ir informācijas pieejamības par produkta stāvokli tirgū savlaicīga nodrošināšana un situācijas attīstības virzienu noteikšana. Šādas informācijas pieejamība veicina ražošanas, pārvadāšanas, produkta popularizācijas u.taml. precīzāku stratēģiju izstrādi, kas samazina izdevumus un uzlabo uzņēmuma efektivitāti.

Produkta dzīves cikls (PDC) attēlo produkta pieprasījuma līkni laika periodā no produkta ieviešanas tirgū līdz produkta izņemšanai no ražošanas un no tirgus (sk. 1.1. attēlu). Produkta dzīves ciklu var attēlot kā virkni no zināmām fāzēm, kas atspoguļo produkta pieprasījuma līmeni noteiktā laika intervālā.



**1.1. att. Produkta dzīves cikla fāzes un izmaksu līmenis**

Produkta dzīves cikla fāžu pārejas periodu noteikšanai bieži pielietotās pieejas balstās uz dinamiskiem lineāriem modeļiem [53, 54] un Bassa izplatīšanās modeli vai arī uz tās modifikācijām [9, 10, 46, 52, 53]. Šādu pieeju pamatideja paredz produkta pieprasījuma līknes prognozēšanu, kurai seko pārejas perioda noteikšana, balstoties uz eksperta individuālās pieredzes. Efektivitāte sistēmām, kuras izmanto minētas pieejas, var samazināties, pārvaldītu produktu skaitam palielinoties, jo ekspertu komandas ilgtermiņa uzturēšanas izmaksas var pārsniegt vienreizējas izmaksas automātiskas pārejas punktu noteikšanas sistēmas ieviešanai.

## 1.2. Intelektuālu aģentu tehnoloģijas koncepcijas

Aģents ir uz aparatūras vai, biežāk, uz programmatūras bāzes izstrādāta datoru sistēma [8, 50]. Tai piemīt šādas īpašības:

- 1) **autonomija** – aģenti spēj darboties bez tiešas citu sistēmu vai cilvēku iejaukšanās, un spēj kontrolēt savas darbības un stāvokli;
- 2) **sociālās spējas** – aģenti sadarbojas ar citiem aģentiem un, iespējams, arī ar cilvēku, izmantojot konkrētam gadījumam izstrādātu valodu;
- 3) **reaktivitāte** – aģenti uztver vidi, kas varētu būt fiziska telpa, lietotājs, citu aģentu kopa, internets, datubāze u.c. vai arī visu minēto kombinācijas; un reaģēt uz izmaiņām vidē, kurā aģenti ir novietoti;
- 4) **proaktivitāte** – aģents, virzoties uz definēto mērķi, spēj pārņemt iniciatīvu, nevis vienkārši izpildīt darbības, reaģējot uz izmaiņām sistēmā.

Aģents eksistē kādā vidē, kurā tas bija novietots. Vide var ietekmēt aģentu, mainot savu stāvokli, ka arī aģents var ietekmēt vidi ar savu aktivitāti, izpildot definētas darbības.

Starp objektiem un aģentiem ir daudz līdzības, tomēr ir arī svarīgas atšķirības; kā vienu no tām var norādīt autonomijas līmeni. Objektorientētā gadījumā lēmums par darbības (metodes) izpildīšanu paliek objekta ziņā, kas izsauc konkrēto metodi. Aģentu gadījumā, lēmums par darbības izpildīšanu paliek aģenta ziņā, kas saņem pieprasījumu par darbības izpildi. Vispārīgi to var aprakstīt ar citātu no grāmatas [50]: „*Objekti to dara par brīvu; aģenti dara to tāpēc, ka viņi grib to darīt*” („*Objects do it for free; agents do it because they want to*”). Savukārt, ir iespējams realizēt aģentu ar objektorientētām tehnikām, ieviešot procedūru

lēmumu pieņemšanai par kādas metodes izpildīšanas palaišanu, taču apskatītā veida autonomija nav daļa no klasiskā objektorientētā modeļa [8, 50]. Nākamā svarīgākā atšķirība starp aģentiem un objektiem ir elastīga (reaktīva, proaktīva, sociāla) autonoma uzvedība. Standarta objektorientētā modelī nav nekas paredzēts šo uzvedības tipu realizācijai [50]. Cita atšķirība starp standarta objektu modeli un aģentu sistēmām ir tas, ka katrs aģents sistēmā kontrolē savu procesu plūsmu (*thread of control*). Standarta objektu modelī sistēmai pastāv tikai viena kopēja procesu plūsma (*single thread*) [50]. Grāmatā [50] M. Wooldridge publicēja multiaģentu sistēmu salīdzināšanas vispārīgos rezultātus par tādām nozarēm, kā „Sadalītās un paralēlās sistēmas”, „Mākslīgais intelekts”, „Ekonomika un spēļu teorija” un „Sociālā zinātne”. Wooldridge norādīja kā līdzības, tā arī svarīgas atšķirības.

### 1.3. Formalizētā uzdevumu nostādne

Analizējot produkta dzīves cikla vadības procesus un izskatot produkta dzīves cikla fāžu robežu noteikšanas uzdevumu ir iespējams formalizēt uzdevumus, kurus ir nepieciešams atrisināt promocijas darba mērķa sasniegšanai.

Produkta dzīves cikla fāzes pārejas perioda prognozēšanas procesa automatizācija var būt izpildīta pielietojot stratēģiju, kad no statistiskiem datiem iegūtās zināšanas tiek pielietotas jaunu datu analīzei. Šāds uzdevums paredz autonomas sistēmas izstrādi, kas realizē datu iegūšanu no glabātuves, to apstrādi, zināšanu bāzes automatisku izveidošanu un atjaunošanu un tās pielietošanu, prognozējot PDC pārejas periodu jauniem produktiem. Svarīgi ir nodrošināt pārejas perioda prognozēšanu datu trūkuma apstākļos – izmantojot tikai dažas pirmās vērtības produkta pieprasījuma līknē.

Pārejas perioda prognozēšanas sistēmas struktūrai jābūt viegli uztveramai, jo kopējais vispārinātais sistēmas modelis atvieglo sistēmas detalizētu projektēšanu, algoritmu un procesu plūsmu formalizāciju. To ir iespējams panākt sistēmas projektēšanai, pielietojot intelektuālu aģentu tehnoloģiju (IAT), kura sniedz pietiekošu funkcionalitāti sarežģītas sistēmas kopējā modeļa izveidošanai un procesu sadalīšanu moduļos – aģentos, kas kopā veido multiaģentu sistēmu. Izmantojot IAT, kļūst iespējams atdalīt datu vadības, datu ieguves un lēmumu pieņemšanas procesus, atsevišķi projektējot autonomus aģentus minētiem procesiem. Multiaģentu sistēmā katram aģentam ir savs mērķis, tomēr tie ir saistīti savā starpā, kas nodrošina informācijas apmaiņu starp aģentiem. Intelektuālu aģentu tehnoloģijas ieviešanai ir nepieciešams izpētīt tās idejas un pamatprincipus, kurus turpmāk pielietot projektējot multiaģentu sistēmu.

Zināšanas bāzes izveidošanas procesam ir iespējams pielietot datu ieguves metodes, jo tās nodrošina pietiekošu funkcionalitāti likumsakarību modeļa izveidošanai. Produkta pieprasījuma dati ir secīgi dati, kuri attēloti īsu laika rindu formātā. Ir nepieciešams izpildīt laika rindu klasterizācijas metožu salīdzinošu analīzi un atlasīt pieprasījuma datu klasterizācijai piemērotas metodes. Bieži vien datu ieguves metodes tikai daļēji nodrošina nepieciešamu funkcionalitāti, līdz ar to ir nepieciešams izstrādāt to modifikācijas īsu laika rindu ar atšķirīgu periodu skaitu klasterizācijai, un pārejas perioda prognozēšanai datu trūkuma apstākļos nosacījuma izpildīšanas nodrošināšanai.

Izstrādātas multiaģentu sistēmas veiktspējas novērtēšanai ir nepieciešams izveidot atbilstošu programmnodrošinājumu, kas implementēs izstrādātus modificētus klasterizācijas algoritmus un realizēs multiaģentu sistēmas struktūru un funkcionalitāti (sk. 4. nodaļu). Sistēmas veiktspējas novērtēšanai ir jāizstrādā eksperimentu plāns, kas nodrošinās drošu un uzticamu pārbaudes rezultātu saņemšanu. Jāizpilda saplānotie eksperimenti un jāveic iegūto rezultātu analīze, balstoties uz kuras var secināt par izstrādātās multiaģentu sistēmas efektivitāti.

### Otrā nodaļa

Mūsdienās aģentu tehnoloģija ir pārstāvēta vairākās nozarēs, ieskaitot paralēlo inženieriju (*concurrent engineering*) [55], sadarbības inženierijas projektēšanu (*collaborative*

*engineering design*), ražošanas uzņēmuma integrāciju un koordinēšanu (*manufacturing enterprise integration*), piegāžu ķēžu vadīšanu [5, 17, 38], ražošanas plānošanu, resursu vadīšanu, sarežģītas (kompleksas) ražošanas sistēmas [8].

N. Jennings [20] secināja, ka aģentu augstā līmeņa mijiedarbības spējas, elastīgums un pielāgojamība atvieglo sarežģītu sistēmu izstrādes procesu. Multiaģentu sistēmas sniedz izstrādātājiem līdzekļus sadalītu sistēmu vadības sarežģītības samazināšanai caur vadības procesa decentralizāciju, tas dod iespēju kontrolēt katru darbību (iterāciju) sistēmā, ļaujot katram aģentam nepārtraukti koordinēt savas darbības ar citiem aģentiem. Nav universālas definīcijas tam, kas ir aģents. Darbā tika izmantotas šādas definīcijas:

- **Aģents** ir datoru programma (vai arī sistēma), kas eksistē kādā vidē un ir spējīga automātiski izpildīt dažādas darbības vidē, cenšoties sasniegt nepieciešamos mērķus.
- **Intelektuālais aģents** ir aģents, kuram piemīt apmācības spējas. Tas ir spējīgs iegūt, apstrādāt, glabāt un izmantot zināšanas par vidi, kurā tas eksistē.
- **Multiaģentu sistēma** satur aģentus un intelektuālus aģentus, kas spēj mijiedarboties, apmainoties ar dažāda veida informāciju.
- **Aģentu saime** ir nedalāma vienība, kas satur aģentus ar līdzīgiem atribūtiem un uzvedību, virzītus uz definēto mērķu sasniegšanu. Atšķirībā no multiaģentu sistēmām, starp aģentiem nav mijiedarbības – tie nav saistīti viens ar otru.

## 2.1. Izstrādātas multiaģentu sistēmas struktūra un koncepcijas

Attēlā 2.1. ir parādīts piedāvātās multiaģentu sistēmas modelis. Izstrādājamai multiaģentu sistēmai tika izvirzītas šādas prasības:

1. Multiaģentu sistēmai jānodrošina pieeja datiem un to sagatavošana turpmākajam analīzes procesam.
2. Sistēmai jānodrošina intelektuālā datu analīze, kuras rezultātā tiks iegūta zināšanu bāze, kas satur saišu modeli starp PDC fāzes pārejas perioda numuru un produkta pieprasījuma līkni.
3. Jābūt nodrošinātam prognozēšanas procesam, kura norises laikā tiks izmantota zināšanu bāze.

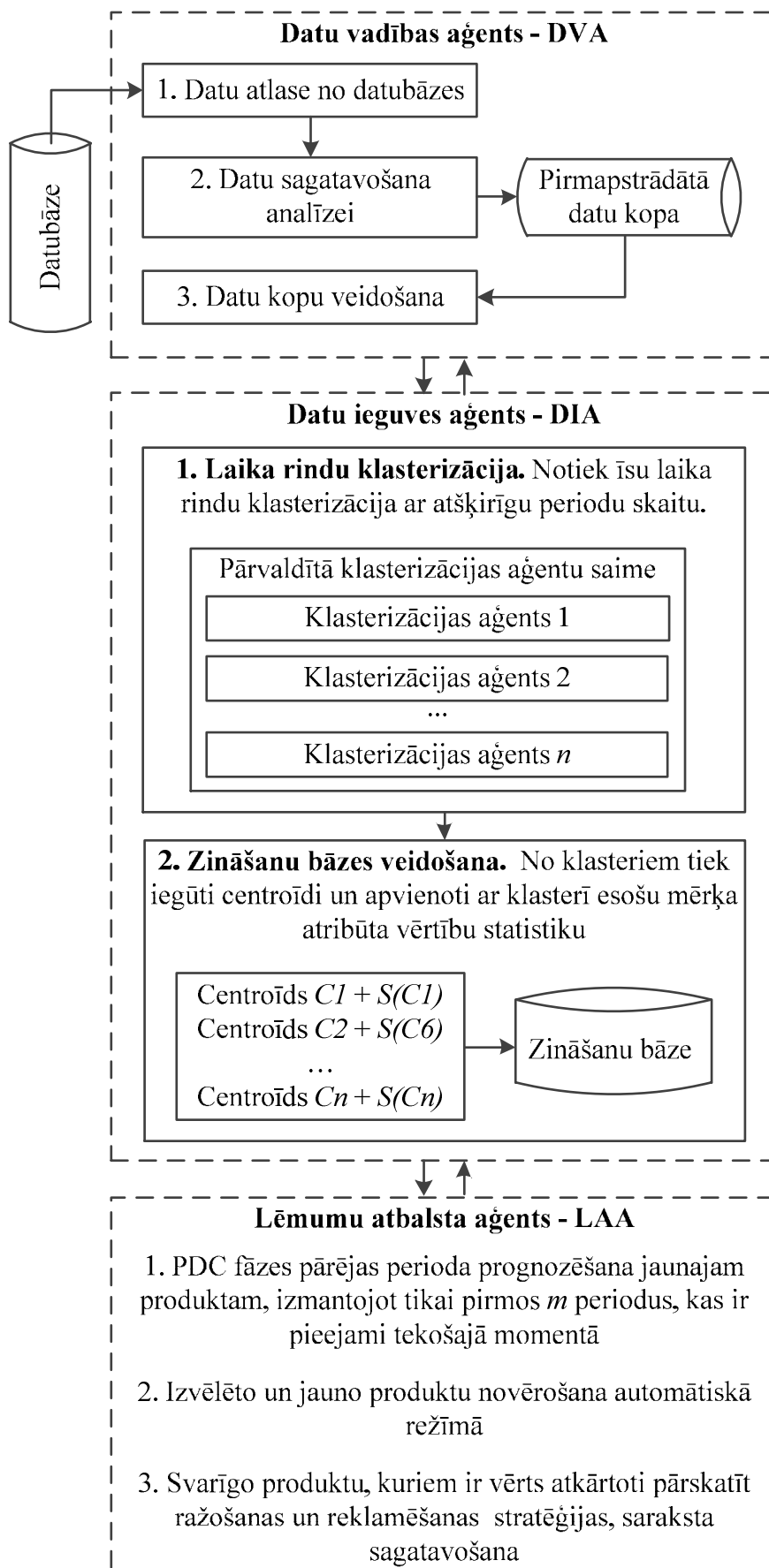
Trīs definētie uzdevumi nepārklājas pēc savas būtības, tomēr, ir savstarpēji saistīti un virzīti uz galamērķa sasniegšanu. Balstoties uz to, iespējams piesaistīt katram uzdevumam savu aģentu un apvienot tos multiaģentu sistēmā. Šādu pieeju var uzskatīt par efektīvu, jo tiek panākta gan ar uzdevumiem saistīto darbību sadalīšana, gan informācijas un datu pieejamība caur aģentu mijiedarbību.

Datu vadības aģents (DVA) nodrošina datu iegūšanu no datubāzes un to sagatavošanu turpmākajam analīzes procesam. Izmantojot grāmatā [39] sniegto aģentu arhitektūru klasifikāciju, DVA ir tuvāk uz mērķa bāzētam aģentam, jo saņemot datus no datubāzes, tas vadās pēc definētas loģikas, lai noteiktu ierakstus, kas tiks sagatavoti un turpmāk lietoti.

Datu ieguves aģents nodrošina zināšanu bāzes izveidošanu, balstoties uz apmācības (vēsturiskiem) datiem un datu ieguves metodēm. Datu ieguves aģentu var klasificēt kā apmācošo aģentu (*learning agent*), jo tas pārbauda izveidotās zināšanu bāzes precizitāti un efektivitāti, tam ir iespējas mainīt apmācības parametrus ar mērķi panākt labāko rezultātu. Datu ieguves aģenta apmācības laikā zināšanu bāze tiek regulāri pielāgota videi.

Lēmumu atbalsta aģenta uzdevums ir prognozēt mērķa atribūta vērtību – produkta dzīves cikla fāzes pārejas periodu, jauniem ierakstiem (jauniem produktiem), pat arī tādos gadījumos, kad pieejami tikai daži pirmie punkti produkta pieprasījuma līknē. Lēmumu atbalsta aģents ir uz piemērotības bāzētais aģents, jo tas, izmantojot prognozēšanas stratēģiju, aprēķina piemērotību prognozējamai vērtībai.

Trīs minētie aģenti ir savstarpēji saistīti un kopā veido piedāvāto multiaģentu sistēmu, kas paredzēta produkta dzīves cikla vadības atbalstam. Izstrādātās sistēmas pielietojamība minētajā virzienā ir analizēta rakstos [27, 29, 30, 31, 33, 35].



2.1. att. Izstrādātas multiaģentu sistēmas modelis

### 2.1.1. Datu vadības aģents

Datu vadības aģenta (DVA) pamata funkcijas ir datu sagatavošana turpmākai analīzei nepieciešamajā formātā un šo datu pieejamības nodrošināšana. DVA ir uz mērķa bāzēts aģents, jo saņemot jaunus datus, tas vadās pēc definētas loģikas, lai noteiktu ierakstus, kas tiks izmantoti analīzei un definētu nepieciešamus datu pirmapstrādes soļus. Vidi, kurā datu vadības aģents ir novietots, veido neapstrādāto datu kopa no ārējas datubāzes. Parādoties jauniem datiem vide ietekmē aģentu, liekot tam izpildīt datu sagatavošanas procedūru. Izpildot minēto datu sagatavošanas procedūru, aģents, savukārt, ietekmē vidi, mainot to un attēlojot kā turpmākajai analīzei sagatavoto datu kopu. Procesi datu vadības aģentā ir shematiski parādīti 2.2. attēlā. To nosacīti var sadalīt divos sektoros:

- Sektors A ietver procesus, kurus aģents var izpildīt automātiskajā režīmā. To izpildes rezultātā tiek iegūta pirmapstrādātā datu kopa.
- Sektors R ietver procesus, kas tiek iniciēti ar ārējo pieprasījumu, t.i. aģents izpilda šos procesus reaktīvi, reaģējot uz DIA pieprasījumiem.

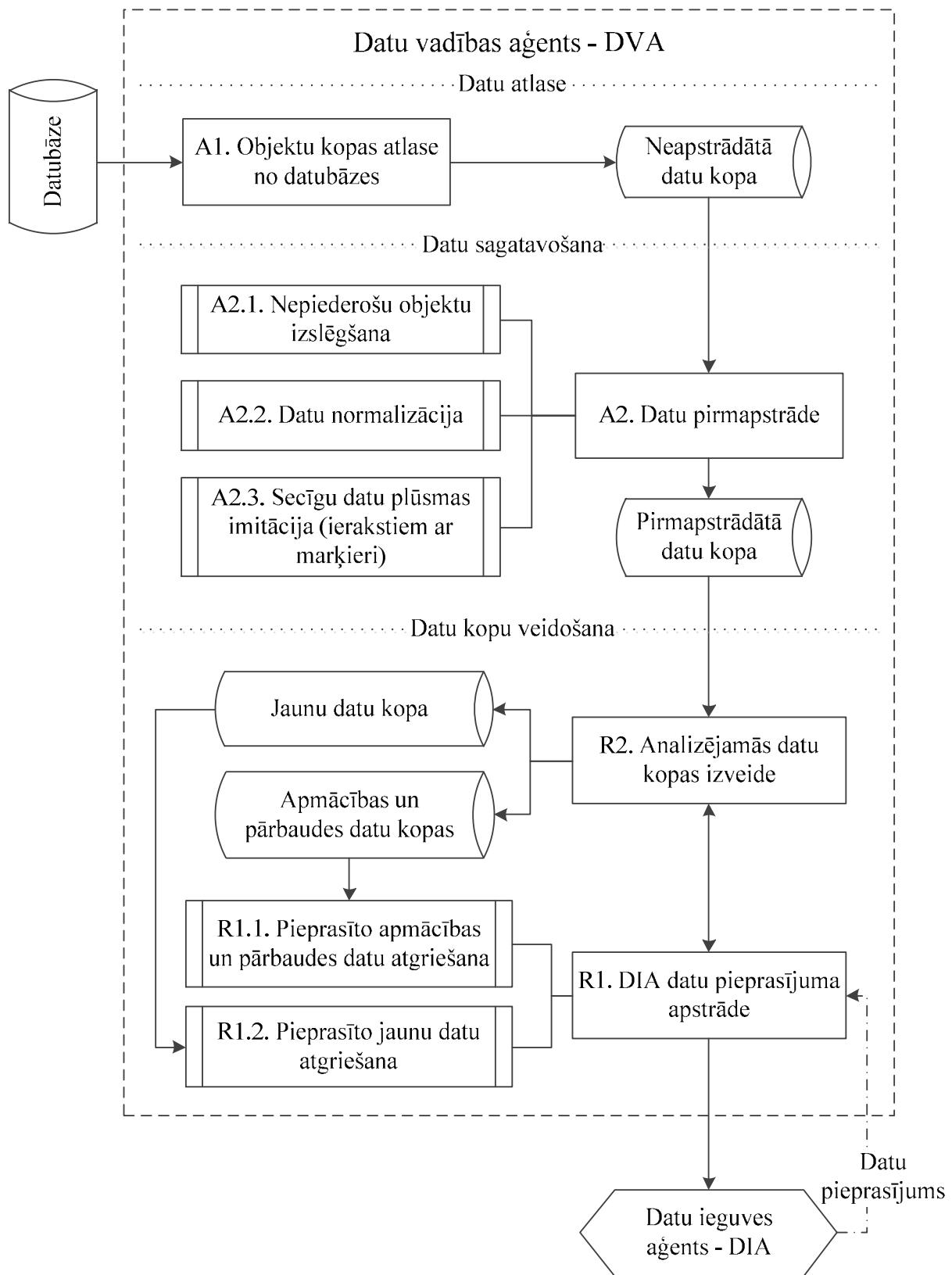
### 2.1.2. Datu ieguves aģents

Datu ieguves aģentam (DIA) ir divi pamatzdevumi – pielietojot intelektuālas datu analīzes metodes, sagatavot saišu modeli starp pieprasījuma laika rindām un mērķa atribūta vērtībām un nodrošināt izveidotā modeļa pieejamību lēmuma atbalsta aģentam. DIA ir intelektuālais aģents. Vide, kurā tas novietots, ir dinamiska un tiek veidota no datu vadības aģenta saņemtajiem sagatavotajiem datiem, kuri var saturēt apmācības datus, sistēmas pārbaudei izdalītus datus un jaunus datus. Mainoties, vide piespiež datu ieguves aģentu izpildīt sistēmas apmācības un zināšanu bāzes izveidošanas procesus, un procesus, kuriem jābūt izpildītiem pirms LAA izpildīs mērķa atribūta vērtības prognozēšanu attiecīgajiem datiem. Izpildot šīs darbības, DIA pats ietekmē vidi. Datu ieguves aģenta procesu blokshēma ir parādīta 2.3. attēlā.

Datu ieguves aģents var ietekmēt vidi divējādi – tiešā un netiešā veidā. Netieša veida ietekmēšanas procesu var aprakstīt šādi: datu ieguves aģents sūta datu vadības aģentam pieprasījumu atgriezt vajadzīgos datus, kura izpildes rezultātā mainās datu ieguves aģenta vide, ierosinot vienu no tiešās ietekmēšanas variantiem. Pastāv šādi varianti tam, ka datu ieguves aģents var ietekmēt vidi tiešā veidā: sistēmas apmācības un pielietošanas stāvokļos.

Sistēmas apmācības stāvoklī pats datu ieguves aģents atrodas apmācības stāvoklī un notiek zināšanu bāzes veidošanas process. Apmācības stāvokli var iniciēt lietotājs vai arī pats aģents, vadoties pēc tā, cik veca ir zināšanu bāze un kā ir mainījusies vide (apmācības kopa) no laika, kad tika izveidota esošā zināšanu bāze. Skatoties no datu ieguves puses, produkta dzīves cikla fāzes pārejas perioda prognozēšana ir klasifikācijas uzdevums, jo mērķa atribūtam ir galīgs iespējamo vērtību skaits un ir pieejami aprakstošie atribūti – produkta pieprasījuma līkne noteiktajā laika periodā.

Formāli zināšanu bāzes veidošanas uzdevumu var definēt šādi. Pieņemsim, ka datu kopa  $D = \{d_1, d_2, \dots, d_i, \dots, d_n\}$  ir pieprasījuma datu kopa, kur katrs ieraksts  $d = (x_1, x_2, \dots, x_j, \dots, x_l)$  ir diskrēta laika rinda ar garumu  $l$ . Ieraksts  $d_i$  reprezentē  $i$ -tā produkta pieprasījuma līkni noteiktajā produkta dzīves cikla fāzē. Laika rindu garums  $l$  nav konstanta vērtība, un var mainīties katram ierakstam  $d$ , pieņemot vērtības no datu kopas  $L$ ,  $l \in L = \{l_1, l_2, \dots, l_h, \dots, l_s\}$ . Katram ierakstam  $d_i \in D$  ir pievienots marķieris  $p$ , kas norāda uz pēdējo periodu produkta dzīves cikla fāzē, kuru attēlo ieraksts  $d_i$ . Marķieris  $p$  var pieņemt galīgo vērtību skaitu,  $p \in P = \{p_1, p_2, \dots, p_k, \dots, p_m\}$ . Ņemot vērā norādītus pieņēmumus, zināšanu bāze kļūst par implikācijas  $f: D \rightarrow P$  modeli, kuru iespējams pielietot marķiera  $p$  vērtības jaunām ierakstam  $d' \notin D$  prognozēšanai.



**2.2. att. Procesi datu vadības aģenta**

Datu ieguves aģenta apmācības process sastāv no 5 secīgiem soļiem (sk. 2.3. attēlu), izskatīsim katru soli:

- 1.1. *Apmācības parametru iegūšana.* Apmācības pirmais solis ir apmācības parametru iegūšana, nosūtot attiecīgo pieprasījumu. Pats pirmais pieprasījums vienmēr tiek

sūtīts lietotājam, lai iegūtu apmācības parametru sākuma vērtības. Nākamie pieprasījumi var tikt sūtīti lietotājam, kā arī var būt realizēti iekšējo pieprasījumu formā, kuru gadījumā datu ieguves aģents pats nostādīs apmācības parametrus, izejot no iepriekšējo apmācības ciklu rezultātiem.

- 1.2. *Apmācības datu iegūšana.* Šajā posmā tiek sastādīts pieprasījums datu vadības aģentam, sagatavot apmācības datus. Atkarībā no parametriem norādītas sistēmas veiktspējas novērtēšanas stratēģijas, datu vadības aģents sagatavo attiecīgus datus. Rezultātā tiek iegūta apmācības datu kopa.
- 1.3. *Klasterizācijas aģentu saimes inicializācija.* Katrs klasterizācijas aģents realizē noteikto datu klasterizācijas algoritmu (sk. 3.2. apakšnodaļu), kuram šajā solī tiek definēti nepieciešamie parametri. Klasterizācijas aģentu skaits ir atkarīgs no parametra  $Q$  – klasterizācijas aģentu noslodzes. Pieņemot, ka datu kopā  $D$  pastāv laika rindas ar periodu skaitu (garumu)  $l$ ,  $l \in L = \{l_1, l_2, \dots, l_h, \dots, l_s\}$ , klasterizācijas aģentu slodze norādīs laika rindas ar cik dažādām  $l$  vērtībām var apstrādāt viens klasterizācijas aģents  $CA$ . Klasterizācijas aģentu skaits  $n_{CA}$  ir aprēķināms pēc formulas (2.1):

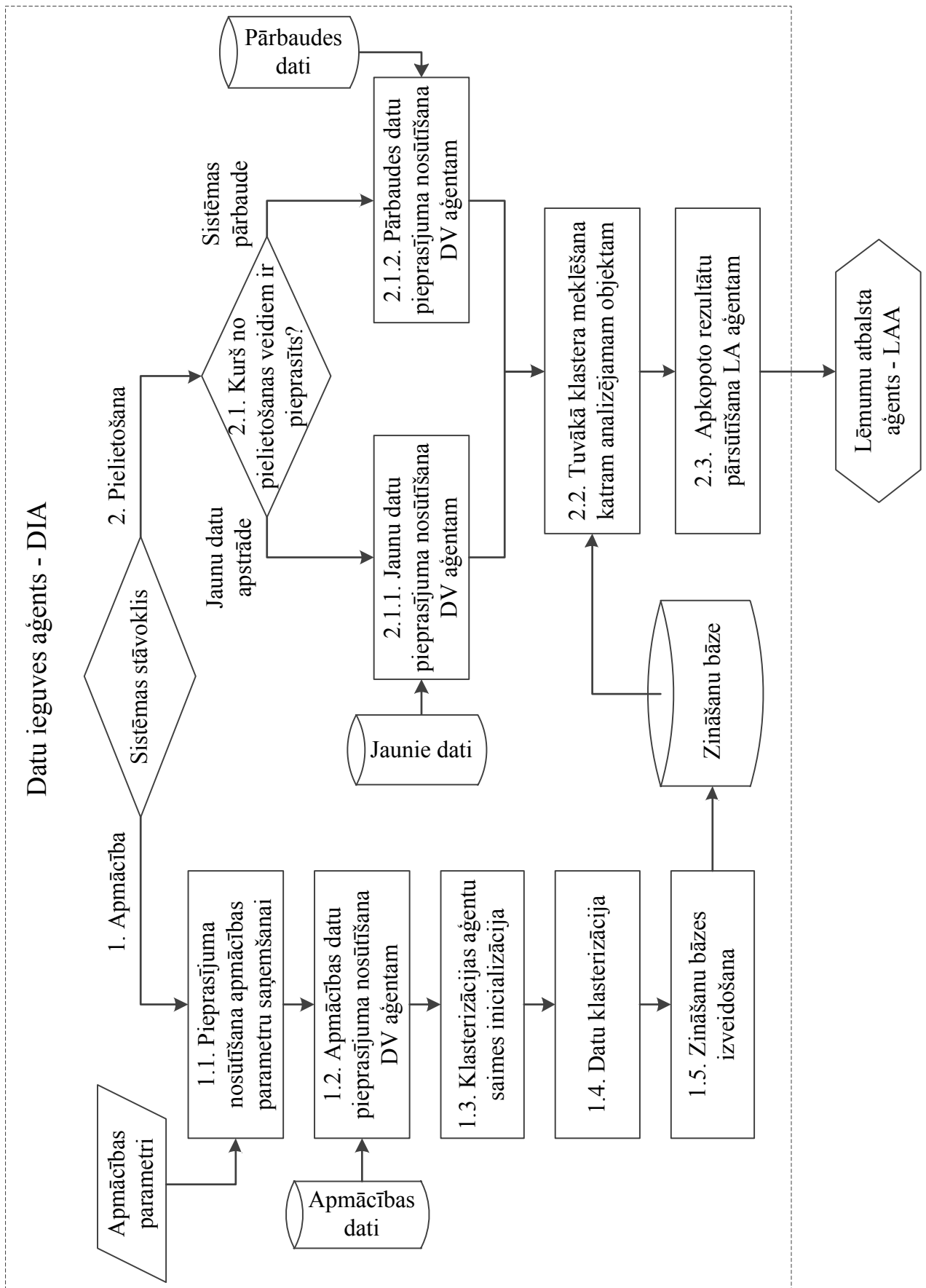
$$n_{CA} = \text{Roundup} \left( \frac{|L|}{Q} \right), \quad (2.1)$$

- 1.4. *Datu klasterizācija.* Piedāvātajā multiaģentu sistēmā datu klasterizācijai tiek izmantotas 3.2. apakšnodaļā piedāvātas pašorganizējošo neironu tīklu algoritma un hierarhiskā gravitācijas klasterizācijas algoritma modifikācijas.
- 1.5. *Zināšanu bāzes izveidošana.* Zināšanu bāze ir sadalīta līmeņos, kuru skaits ir vienāds ar klasterizācijas aģentu skaitu, un katrā līmenī tiek saglabāti klasteri, kurus ieguva attiecīgais klasterizācijas aģents.

Zem sistēmas pielietošanas stāvokļa ir apvienoti jaunu datu analīzes process un sistēmas pārbaudes process. Sistēmas pielietošanas process sastāv no šādiem soliem:

- 2.1. *Analizējamo datu iegūšana.* Atkarībā no tā, vai ir nepieciešams pārbaudīt sistēmas veiktspēju, vai izpildīt prognozi jaunajam produktam, datu vadības aģentam tiek sūtīts pieprasījums pēc pārbaudes kopas vai arī pēc jaunu ierakstu datu kopas. Gadījumā, kad ir nepieciešams izpildīt prognozi jaunam produktam, var tikt pieprasīts konkrētais ieraksts.
- 2.2. *Tuvākā klastera meklēšana.* Kad datu kopa ar analizējamiem datiem ir saņemta, katram ierakstam (objektam) tiek atrasts tuvākais klasteris attiecīgajā zināšanu bāzes līmenī, kas atbilst ieraksta garumam.
- 2.3. *Rezultātu pārsūtīšana lēmumu atbalsta aģentam.* Šajā solī informācija par tuvākiem klasteriem tiek pārsūtīta lēmumu atbalsta aģentam, kas izpilda prognozi.

Apmācības un pārbaudes cikls atkārtojas vairākas reizes, kamēr netiek noteikti apmācības parametri, ar kuriem tiek iegūts labākais rezultāts, kas apmierina lietotāja norādītus labuma kritērijus. Kad parametru vērtības ir noteiktas, zināšanu bāze tiek veidota, izmantojot pilnu datu kopu, kas apvieno sevī datus no apmācības un pārbaudes kopām.



2.3. att. Procesi datu ieguves aģentā

### 2.1.3. Lēmumu atbalsta aģents

Lēmumu atbalsta aģents ir aģents, kas prognozē produkta dzīves cikla fāzes pārejas periodu jauniem produktiem, uzrauga lietotāja izvēlētos un jaunus produktus, un sagatavo sarakstu ar produktiem, kuriem derētu pārskatīt ražošanas plānošanas un reklāmas stratēģijas.

Lēmumu atbalsta aģentu vide ir jaunu datu kopas un sistēmas pārbaudei lietojamas datu kopas. Parādoties jauniem ierakstiem, LAA pieprasa attiecīgo informāciju no zināšanu bāzes un prognozē mērķa atribūta vērtību, ar to mainot vidi. Attēlā 2.4. ir parādīta lēmumu atbalsta aģenta iekšējo procesu blokshēma. Atbilstoši sistēmas stāvokļiem „Pārbaude” un „Jaunu datu analīze”, lēmumu atbalsta aģentam ir divas procesu plūsmas, kuras var tikt izpildītas kā autonomā režīmā, tā arī izpildot lietotāja pieprasījumu. Sistēmas pārbaudes stāvoklī procesu plūsma sastāv no šādiem četriem soļiem:

- 1.1. *Pārbaudes datu analīze.* Datu ieguves aģentam tiek sūtīts pieprasījums noteikt tuvākus klasterus ierakstiem no pārbaudes datu kopas. Šis pieprasījums aktivizē datu ieguves aģentā šādu procesu virkni – solis 2.1.2., solis 2.2. un solis 2.3.
- 1.2. *Prognozēšanas procedūra.* Piedāvātajā multiaģentu sistēmā tiek lietota mērķa atribūta vērtības izvēles stratēģija, kas ir aprakstīta 3.3. apakšnodaļā.
- 1.3. *Prognozes precizitātes novērtēšana.* Piedāvātās sistēmas veikspējas novērtēšanas stratēģija ir aprakstīta 3.3. apakšnodaļā.
- 1.4. *Atskaite par pārbaudes rezultātiem.* Šajā solī sistēmas pārbaudes rezultāti tiek pārsūtīti lietotājam vai arī datu ieguves aģentam, lai tas varētu izpildīt korekcijas un pielāgot apmācības procedūru izmantotajai datu kopai.

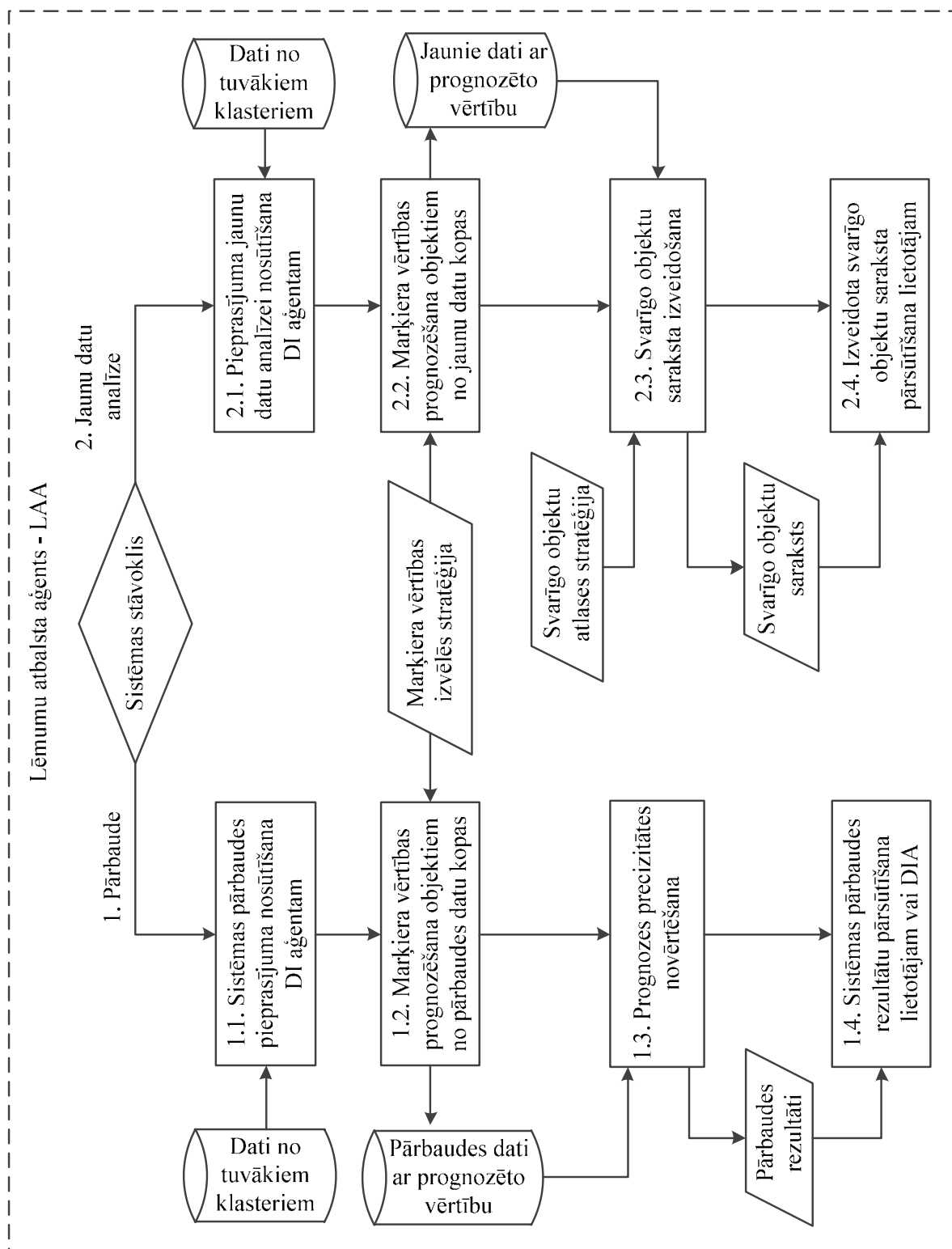
Jaunu ierakstu apstrādes gadījumā procesu plūsma ietver šādus četrus soļus:

- 2.1. *Jaunu datu analīze.* Procesi šajā solī sakrīt ar procesiem 1.1. solī sistēmas pārbaudes stāvoklī.
- 2.2. *Prognozēšanas procedūra.* Procesi šajā solī sakrīt ar procesiem 1.2. solī sistēmas pārbaudes stāvoklī, ņemot vērā ka dati, kuriem tiek izpildīta prognozēšanas procedūra, ir jaunie dati, kuriem nav pieejama mērķa atribūta īstā vērtība.
- 2.3. *Svarīgo objektu saraksta veidošana.* Šajā solī tiek veidots saraksts no produktiem, kuriem ir ieteicams pārskatīt ražošanas plānošanas un reklāmas stratēģijas. Tiek izmantota svarīgo objektu atlases stratēģija, kuru izvēlas lietotājs. Piedāvātajā multiaģentu sistēmā tiek lietota šāda stratēģija:
  - a) Ja  $l < p$  un  $p - l > \theta$ , tad produkts netiek iekļauts svarīgo objektu sarakstā un paliek uzraudzīts;
  - b) Ja  $l < p$  un  $p - l \leq \theta$ , tad produkts tiek iekļauts svarīgo objektu sarakstā;
  - c) Ja  $l \geq p$ , tad produkts tiek iekļauts svarīgo objektu sarakstā,kur  $l$  ir periodu skaits analizējamā laika rindā;  $p$  ir mērķa atribūta prognozētā vērtība;  $\theta$  atspoguļo minimālo svarīguma robežu, lai produkts tiktu iekļauts svarīgo objektu sarakstā.
- 2.4. *Atskaite par paveikto jaunu ierakstu analīzi.* Šajā solī, jau izveidotais svarīgo objektu saraksts, tiek pārsūtīts lietotājam, kas pieņem gala lēmumu par katra produkta statusu.

### Otras nodaļas kopsavilkums

1. Multiaģentu sistēmu trūkumu un priekšrocību analīze sniedza nepieciešamo informāciju intelektuālo aģentu tehnoloģijas pielietošanas pamatojumam, risinot promocijas darbā definētus uzdevumus.

- Balstoties uz promocijas darba uzdevumos izteiktu multiaģentu sistēmai pieprasījumu analīzes, izstrādāts multiaģentu sistēmas modelis, kuras galvenais mērķis ir prognozēt produkta dzīves cikla fāzes pārejas periodu.
- Pielietojot multiaģentu sistēmu funkcionālas slodzes sadalīšanas spējas, tika formulēti konceptuālie uzdevumi katram aģentam izstrādātā multiaģentu sistēmā. Tika izstrādāti un aprakstīti darbības algoritmi un iekšējo procesu shēmas datu vadības aģentam, datu ieguves aģentam un lēmumu atbalsta aģentam.



2.4. att. Procesi lēmumu atbalsta aģentā

Tādējādi, pierādīta pirmā izvirzītā hipotēze - intelektuālu aģentu tehnoloģijas izmantošana var atvieglot sarežģītu vadības un vadības atbalsta sistēmu projektēšanu, piedāvājot plašas iespējas struktūras, procesu un informācijas plūsmu formalizācijai. Intelektuālu aģentu tehnoloģijas ieviešana veicināja sarežģītas produkta dzīves cikla vadības atbalsta sistēmas vienkāršošanas procesu, piedāvājot iespējas sistēmas struktūras un procesu sadalīšanai starp mijiedarbojošiem aģentiem, kuri veido multiaģentu sistēmu.

## **Trešā nodaļa**

Nodaļā ir apkopota informācija par datu ieguves metodēm, kuru izmantošana ir paredzēta sistēmas darbības procesā. Apakšnodaļa 3.1. apkopo pamata informāciju par datu pirmapstrādes procesa nepieciešamību un svarīgumu. Datu ieguves aģenta datu klasterizācijas procesā izmantoto metožu klasisko algoritmu un to piedāvāto modifikāciju apraksti ar pamatojumiem ir apkopoti 3.2. apakšnodaļā. Šo nodaļu noslēdzošā 3.3. apakšnodaļa apkopo informāciju par jaunu objektu mērķa atribūta vērtības prognozēšanai piedāvātām un izmantotām pieejām, kā arī par izstrādātās sistēmas veikspējas novērtēšanas pieejām.

### **3.1. Datu pirmapstrāde**

Datu apjoms, kas tiek izmantots reālajā pasaulē ar datu ieguvi risināmās problēmās, ir ļoti liels un palielinās kopā ar laiku, un ar datu glabāšanas tehnoloģiju attīstību. Bieži vien dati nāk no vairākiem avotiem un satur trokšņainus, trūkstošus vai pretrunīgus datus, kuru rašanās iemesli var būt dažādi. Dati var būt nepareizi savākti vai apkopoti cilvēku vai tehnoloģiju dēļ, var rasties atšķirības pielietotos nosaukumos vai kodos, vai datu formātos, interesējošie atribūti ne vienmēr ir pieejami tāpēc, ka nebija saglabāti kāda pārpratuma vai aparatūras disfunkcijas rezultātā. Neapstrādātie un kļūdainie dati noved pie kļūdainiem rezultātiem, tāpēc pirms datu iekšējo saišu modeļa uzbūves, sistēmas apmācībai vai citu ar zināšanu ieguvi saistītu darbību izpildi ir ieteicams vienmēr izpildīt datu pirmapstrādi. Kopumā pirmapstrādes loma datu ieguves praksē ir tāda, ka šis process var aizņemt līdz pat 80% no visa datu ieguves projekta izstrādāšanas laika. Pastāv dažas datu pirmapstrādes tehnoloģijas [2, 18, 37, 45], kas nav viena otru izslēdzošas un var būt lietojamas vienlaikus:

- *Datu attīrīšanu* pielieto trokšņa nogludināšanai vai izslēgšanai, trūkstošo vērtību aizpildīšanai un pretrunīgu datu izslēgšanai.
- *Datu transformācijai* tiek izmantotas tādas tehnoloģijas, kā nogludināšana, agregācija, vispārināšana, atribūtu konstruēšana un normalizācija, kas uzlabo datu ieguves algoritmu darbību.
- *Datu integrācija* ir dažādu avotu apvienošana vienā saistītā datu bāzē, tādā kā datu noliktavā.
- *Datu reducēšana*.

### **3.2. Datu klasterizācija**

Klasterizācija jeb klasteru analīze ir objektu grupēšana klasteros tā, lai līdzība starp objektiem vienā klasterī būtu lielāka, nekā līdzība starp dažādu klasteru objektiem [2, 15, 18]. Klasterizācijas rezultātā ir iespējams iegūt datu kompakto aprakstu, piemēram, izdalīt bieži sastopamus produkta dzīves cikla attīstības modeļus un tālāk strādāt ar tiem.

Klasteru analīze kalpo gan datu aprakstīšanai, gan tālākai rezultātu izmantošanai. Tās funkcija izriet no tā, ka klasterizācija veido individuālo objektu abstrakciju. Vairāku ierakstu vietā tiek iegūti reprezentējošie prototipi, kas var kļūt par citu datu analīzes pamatu [2, 45], piemēram, prognozējot produkta pieprasījuma attīstību laikā, kas ir labi demonstrēts literatūras avotā [22].

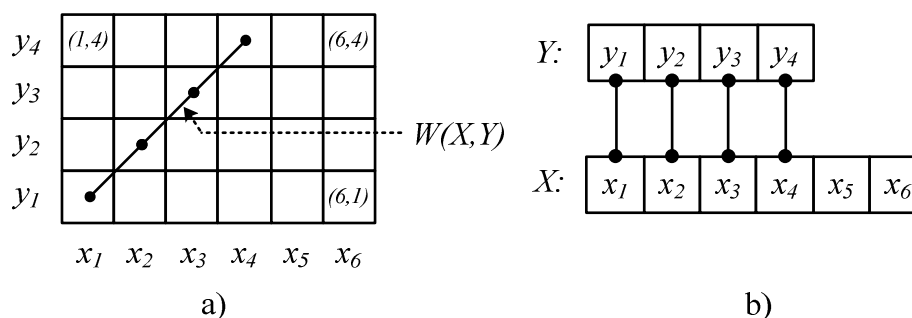
Atlasot darbā definētu uzdevumu risināšanai piemērotus klasterizācijas algoritmus, tika analizēti šādi algoritmi [2]:  $k$ -vidējo un  $k$ -medoīdu sadalošie klasterizācijas algoritmi; aglomeratīvais, divizīvais un gravitācijas hierarhiskie klasterizācijas algoritmi, *DBSCAN* klasterizācijas algoritms un pašorganizējošie neironu tīkli. Analīzes rezultātā turpmākai pētīšanai tika izvēlēti hierarhiskais gravitācijas klasterizācijas algoritms un pašorganizējošie neironu tīkli. Promocijas darba ietvaros izpildītie pētījumi par izvēlēto metožu piemērotību laika rindu klasterizācijai ir apkopoti literatūras avotos [27, 28, 32, 34, 35, 60].

### 3.2.1. Izmantotie attāluma mēri

Darbā izskatītajos klasterizācijas algoritmos laika rindu līdzība tiek mērīta, balstoties uz šo laika rindu attālumu, kas aprēķināms ar tādiem mēriem kā *DTW* – *Dynamic Time Warping* un uz *Eiklīda* attāluma bāzes izstrādātā attāluma noteikšanas heuristika, kas turpmāk tiks dēvētā par *MEuclidean*.

Dinamiskā laika izlīdzināšana (*DTW*) ir pieeja, ar kuras palīdzību tiek meklēta laika izlīdzināšanas (*warping*) pozīcija starp divām laika rindām. Attāluma noteikšanas *DTW* pieeju var aprakstīt ar trim galvenajiem soļiem – attālumu matricas izveidošana, izlīdzināšanas ceļa noteikšana un attāluma aprēķināšana, balstoties uz atrastā izlīdzināšanas ceļa. Izlīdzināšanas ceļš  $W(X, Y)$  ir virkne no attālumu matricas elementiem, tas tiek lietots laika rindu  $X$  un  $Y$  salīdzināšanas kartes izveidošanai.

Darbā ir piedāvāta *Eiklīda* attāluma modifikācija (*MEuclidean*), pielietojot kuru var aprēķināt attālumu starp atšķirīga garuma laika rindām. Galvenā ideja ir tāda, ka rēķinot attālumu starp laika rindām ar atšķirīgu periodu skaitu, tiek lietotas tikai abām laika rindām kopējās dimensijas (sk. 3.1. attēlu).



3.1. att. Izlīdzināšanas ceļš un atbilstošā salīdzināšanas karte *MEuclidean* mēram

### 3.2.2. Pašorganizējošie neironu tīkli

Pašorganizējošie neironu tīkli tiek pieskaitīti pie mākslīgo neironu tīklu grupas, kas saucas pašorganizējošas kartes (*Self-Organising Maps, SOM*). Šie tīkli izmanto apmācību bez skolotāja (*unsupervised learning*) un balstās uz konkurences apmācības (*competitive learning*). Neironi no tīkla izeju slāņa konkurē savā starpā par aktivēšanas tiesībām, rezultātā tiek aktivēts tikai viens neirons, kas tiek dēvēts par uzvarētājneironu (*winning neuron*). Tādas neironu konkurences organizēšanas būtībā ir laterālo (atgriezenisko) saišu starp neironiem izmantošana [14, 61], pirmo reizi šo ideju piedāvāja *F.Rozenblatts* [39]. Darbā tika izskatītas un izmantotas, veicot praktiskus eksperimentus, šādas topoloģijas: kvadrāta topoloģija, rombveida (krustveida) topoloģija un lineārā topoloģija. Klasiskais pašorganizējošo neironu tīklu algoritms sastāv no 4 pamata soļiem:

1. solis. *Tīkla inicializācija*. neironu svāriem gadījuma veidā tiek piešķirtas sākuma vērtības. Tām jābūt pietiekoši mazām, lai nodrošinātu, ka apmācības procesa sākumā neironu tīklā nav nekādas organizācijas starp neironiem [61].
2. solis. *Neironu konkurences (competition)* procesā starp visiem neironiem tīklā tiek noteikts uzvarētājneirons, kas atgriež labāko neironu piemērotības funkcijas

vērtību. Darba izskatītajā uzdevumā par neironu piemērotības funkciju tika ņemta minimizējama attāluma funkcija starp laika rindu  $X$  un neirona svaru vektoru  $W$ .

3. solis. *Neironu sadarbība (cooperation)*. Sadarbības solī uzvarētājneirons ierosina tuvākus kaimiņus, kuri kļūst par atkārtoti ierosināmiem neironiem. Uzvarētājneirona topoloģiskā apkārtnē tiek apzīmēta kā  $h_{j,i(x)}$ , kur  $i(x)$  ir uzvarētājneirona indekss ieejas signālam  $x$  un  $j$  ir atkārtoti ierosināmā neirona indekss. Topoloģiskā apkārtnē samazinās laikā, kas ir pašorganizējošo neironu tīklu algoritma unikālā īpašība.
4. solis. *Neironu sinoptiskās adaptācijas (synoptic adaptation)* process sastāv no diviem etapiem – neironu tīkla organizācijas (*organization*) un neironu tīkla konverģences (*convergence*) [23, 61]. Organizācijas etapā notiek neironu svaru vektoru topoloģiskā izkārtošana. Konverģences procesā notiek neironu svaru vektoru precīza pielāgošana ieejas signālu telpai – apmācības kopai. Neironu svaru vektora adaptācijai tiek lietota formula (3.1).

$$W_j(n+1) = W_j(n) + \eta(n) \cdot h_{j,i(x)}(n) \cdot (X - W_j(n)), \quad (3.1)$$

kur  $W_j(n)$  –  $j$ -tā neirona svaru vektors  $n$ -tā iterācijā;

$X$  – ieejas signāls, laika rinda no apmācības kopas;

$\eta(n)$  – apmācības koeficients  $n$ -tā iterācijā;

$h_{j,i(x)}(n)$  – topoloģiskās apkārtnes koeficients  $n$ -tā iterācija.

### 3.2.3. Pašorganizējošo neironu tīklu algoritma modifikācijas

Pašorganizējošie neironu tīkli ar klasisko algoritmu nav spējīgi apstrādāt ieejas vektorus ar atšķirīgu garumu – visiem ieejas vektoriem jābūt vienāda garuma. Pašorganizējošo neironu tīklu algoritma konkurences procesā uzvarētājneirona noteikšanai *Eiklīda* attāluma mērs tika aizvietots ar piedāvāto *MEuclidean* attāluma mēru un *DTW*. Pielietojot katru no minētiem mēriem, tiek izveidota salīdzināšanas karte  $M$ , kas sastāv no elementiem  $(x_i, w_j)$ , kas norāda, kurai laika rindas  $X$  vērtībai  $x_i$  jāpielāgojas neirona konkrētajam svaram  $w_j$ . Izmantojot attāluma mēru *MEuclidean* ir iespējamas šādas situācijas:

1. Neirona svaru vektora  $W$  garums  $l_w$  ir mazāks vai vienāds ar laika rindas  $X$  garumu  $l_x$ . Šajā gadījumā, atbilstoši salīdzināšanas kartei  $M$ , tiek pielāgoti visi neirona svari;
2. Neirona svaru vektora  $W$  garums  $l_w$  ir lielāks par laika rindas  $X$  garumu  $l_x$ . Šajā gadījumā laika rindas vērtībām, atbilstoši salīdzināšanas kartei  $M$ , tiek pielāgoti tikai pirmie  $l_x$  svari, pārējie svari ar indeksiem, kas lielāki par  $l_x$ , netiek mainīti.

Ievērojot aprakstītās situācijas, pašorganizējošo neironu tīklu algoritma neironu adaptācijas procesā tika izdarītas atbilstošas modifikācijas. Tika izmainīts neironu svaru pielāgošanas process, lai būtu iespējams izmantot katru no 3.2.1. sadaļā piedāvātajiem attāluma mēriem. Izmantojot darbā piedāvāto *MEuclidean* attāluma mēru, uzvarētājneirona un atkārtoti ierosināmo neironu svaru pielāgošanai izmantotās formulas (3.1) vietā tiek izmantota jauna formula (3.2):

$$\forall w_k \in W_j, k \leq l_{\min} : w_k(n+1) = w_k(n) + \eta(n) \cdot h_{j,i(x)}(n) \cdot (x_k - w_k(n)), \quad (3.2)$$

kur  $W_j$  –  $j$ -tā neirona svaru vektors;

$$l_{\min} = \min(l_X, l_W);$$

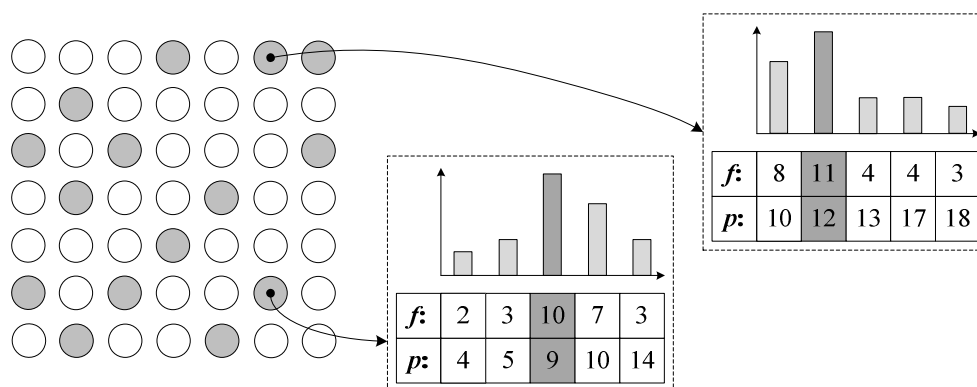
$\eta(n)$  – apmācības koeficients  $n$ -tā iterācijā;

$h_{j,i(X)}(n)$  – topoloģiskās apkārtnes koeficients  $n$ -tā iterācijā.

Zināšanu bāzes izveidošanas procesa nodrošināšanai pašorganizējošo neironu tīklu algoritmam tika pievienots jauns solis, kurā tiek sagatavots apmācības procesā iegūto klasteru saraksts. Ir piedāvāta šāda stratēģija klasteru saraksta sagatavošanai:

1. Katrs no apmācības kopas elementiem tiek sūtīts uz neirona tīkla ieejām.
2. Ar apmācības kopas elementiem ierosinātiem neironiem tiek vākta statistika – cik reizes elements ar noteikto mērķa atribūta vērtību ierosināja konkrēto neironu.
3. Neirona, kas vismaz vienu reizi bija ierosināts klasteru iegūšanas procesa izpildes laikā, svaru vektors tiek kopēts atbilstošā klasterī  $c_j$  kopā ar neironam savākto statistiku 2. punktā.

Klasteris  $c_j \in C$  ir objekts, kas satur  $j$ -tā neirona svaru vektoru  $W_j$  un mērķa atribūta vērtību  $p_i \in P$  iekrišanas biežuma statistiku  $S_j$  (sk. 3.2. attēlu), kas sastāv no elementiem  $(p_i, f)_j$ .



### 3.2. att. Klasteriem apkopotās statistikas $S_j$ piemērs pašorganizējošiem neironu tīkliem

Katrā pāri  $(p_i, f)_j$  elements  $f$  ir mērķa atribūta vērtības  $p_i$  iekrišanas biežums  $j$ -tā neironā. Klasteru iegūšanas un klasteru saraksta formēšanas procedūra var tikt mainīta atbilstoši risināmām uzdevumam.

#### 3.2.4. Gravitācijas klasterizācijas algoritms – *G-Algorithm*

Algoritms *G-Algorithm* [16] ir bez skolotāja apmācības klasterizācijas algoritms, kas ir noturīgs pret kļūdām un trokšņiem un balstās uz *Gravitācijas likumu* un *otro Ņūtona kustības likumu*. Katra datu virkne – laika rinda, analizējamā datu kopā tiek uzskatīta par objektu  $n$ -dimensiju telpā, kas tiek kustināts izmantojot gravitācijas spēku. Līdz šai metodei tika izstrādāts gravitācijas klasterizācijas algoritms, kuru piedāvāja *W.E.Wright* [51], un šī metode ir hierarhiskais aglomeratīvais algoritms, jo gravitācijas spēks tika izmantots kā mehānisms objektu (punktu) apvienošanai līdz brīdim kamēr telpā paliek tikai viens objekts (klasteris). Klasiskais gravitācijas klasterizācijas algoritms sastāv no šādiem soļiem:

1. solis. Tiek definēti klasterizācijai nepieciešamie parametri – gravitācijas konstantes vērtība, vai gravitācijas spēks samazināsies laikā vai nē, attāluma robeža, sasniedzot kuru objekti tiks apvienoti u.c.

2. solis. Tiek veidota klasterizācijas telpa – vide, kurā notiek objektu pārvietošana balstoties uz gravitācijas spēku, un objektu apvienošanas process.
3. solis. Katram objektam tiek aprēķināts uz to ietekmējošs gravitācijas spēks.
4. solis. Atbilstoši aprēķinātam gravitācijas spēkam, katrs objekts tiek pārvietots telpā. Objekta  $x$  pozīcijas izmaiņu aprēķiniem, kuri tika izraisīti ar objekta  $y$  gravitācijas spēku, tiek izmantota formula (3.3):

$$x(t+1) = x(t) + \overline{d(t)} \cdot \frac{G \cdot m_y}{2 \cdot \|\overline{d(t)}\|^3}, \quad (3.3)$$

kur  $\overline{d(t)} = x(t) - y(t)$ ;  $G$  – gravitācijas konstante;  $m_y$  – objekta  $y$  masa.

5. solis. Tiek pārbaudīts vai kādi objekti sasniedza apvienošanas robežu. Ja tādi pastāv, tad tie tiek apvienoti. Algoritms atgriežas uz otro soli un izpildās, kamēr neiestāsies apstāšanās kritērijs.
6. solis. Izpildoties apstāšanās kritērijam, tiek veidots klasteru saraksts. Ja kumulatīvajā objektā ir apvienots pietiekoši daudz elementu (skaits tiek norādīts ar speciālu parametru), tad kumulatīvais objekts kļūst par klasteri un tiek iekļauts sarakstā. Klasterim tiek aprēķināts pārstāvis kā iekļauto objektu centroīds.

### 3.2.5. Gravitācijas klasterizācijas algoritma modifikācijas

Klasiskais gravitācijas klasterizācijas algoritms *G-Algorithm* izmanto *Eiklīda* attāluma mēru objektu tuvuma noteikšanai. Tas ievieš ierobežojumu klasterizējamiem datiem – visām laika rindām jābūt ar vienādu periodu skaitu.

Iespējas nodrošināšanai vienlaicīgi klasterizēt īsas laika rindas ar atšķirīgu garumu, gravitācijas klasterizācijas algoritmam tika modificēti otrais, trešais un sestais soļi. Ir izdarītas šādas modifikācijas:

2. solis. Tika mainīts viens no klasterizācijas telpas ierobežojumiem, kas skan šādi: „*N*-dimensiju telpā var eksistēt tikai objekti ar *n* dimensijām”. Piedāvātajā modificētajā redakcijā šis ierobežojums skan šādi: „*N*-dimensiju telpā var eksistēt objekti ar *m* dimensijām, kur  $m \leq n$ ”. Tas nozīmē, ka 10-dimensiju telpā var eksistēt laika rinda ar 4 periodiem, bet tā būs pārstāvēta tikai pirmajās 4 dimensijās. Dimensijās, kurām kārtas numurs ir lielāks par 4, šī laika rinda nebūs pārstāvēta. Šis pieņēmums dod iespēju ieviest jaunus attāluma mērus un modificēt gravitācijas spēka aprēķināšanas stratēģiju, mainot gravitācijas klasterizācijas algoritma 3. soli.
3. solis. *Eiklīda* attālums tika aizvietots ar vienu no attāluma mēriem, aprakstītiem 3.2.1. sadaļā. Lietojot *MEuclidean* mēru, attālumu aprēķina, izmantojot tikai salīdzināmo objektu  $X$  un  $Y$  kopējās dimensijas. Rezultātā ir iespējami šādi gadījumi:
  1. Dimensiju skaits objektam  $X$ , kuram tiek aprēķināts gravitācijas spēks, ir mazāks vai vienāds ar ietekmējošā objekta  $Y$  dimensiju skaitu. Šajā situācijā gravitācijas spēks tiek aprēķināts visām objekta  $X$  dimensijām un tas tiek pārvietots telpā atbilstoši katrai dimensijai aprēķinātajam gravitācijas spēkam.
  2. Dimensiju skaits objektam  $X$  ir lielāks par ietekmējošā objekta  $Y$  dimensiju skaitu. Gravitācijas spēks tiek aprēķināts tikai abu objektu

kopējām dimensijām, līdz ar to objekts  $X$  tiek pārvietots tikai dimensijās, kurām tika aprēķināts gravitācijas spēks.

Lietojot  $DTW$  mēru, tiek veidota objektu salīdzināšanas karte, kas nosaka to kāda ietekmējošā objekta dimensija ietekmēs kādu ietekmējamā objekta dimensiju. Neatkarīgi no objektu dimensiju skaita, aprēķinot attālumu starp objektiem, vienmēr tiek lietotas visas gan ietekmējamā, gan ietekmējošā objekta dimensijas.

6. solis. Par klastera centru tiek ņemta kumulatīvā objekta pozīcija klasteru ieguves posmā. No klasterī apvienotiem objektiem tiek vākta mērķa atribūta vērtību  $p_i \in P$  parādīšanas biežuma statistika, no elementiem  $(p_i, f)_j$  sastāvošas kopas  $S_j$  veidā, kur  $f$  ir vērtības  $p_i$  parādīšanas biežums  $j$ -tā klasterī. Savāktā statistika tiek lietota mērķa atribūta prognozēšanai jauniem objektiem.

### 3.3. Prognozēšanas procedūra un veiktspējas mērīšana

Mērķa atribūta vērtības prognozēšanas procedūra tiek palaista ar attiecīgu lietotāja pieprasījumu, izpildot prognozēšanu jaunajam objektam, vai arī ar sistēmas iekšējo pieprasījumu no LAA – Lēmumu Atbalstā Aģenta, pārbaudot sistēmu pēc apmācības posma pabeigšanas vai arī sistēmai funkcionējot autonomā režīmā. Abos gadījumos mērķa atribūta vērtību prognozēšana notiek pēc vienāda algoritma, kas sastāv no trim soliem:

1. Noteikt līmeni zināšanu bāzē, kur tiks meklēts objektam tuvākais klasteris. Katrs līmenis zināšanu bāzē atbilst attiecīgajam klasterizācijas aģentam, līdz ar ko arī saglabājas laika rindu garuma robežas;
2. Izvēlētajā zināšanu bāzes līmenī atrod objektam tuvāko klasteri  $c_j$ . Tuvums līdz objektam tiek noteikts, balstoties uz viena no 3.2.1. sadaļā aprakstītiem attāluma mēriem – jo mazāks attālums, jo klasteris ir tuvāk objektam;
3. Balstoties uz klasterī  $c_j$  esošās statistikas  $S_j$ , prognozējamā vērtība  $p_i \in P$  tiek izvēlēta kā viena ar lielāko parādīšanās biežumu  $f$ .

Prognozēšanas procedūras trešajā solī ir iespējams gadījums, kad vairākām  $p_i$  vērtībām būs vienāds lielākais parādīšanās biežums  $f$ . Šajā gadījumā prognozējamā vērtība tiek ņemta mazākā pēc moduļa. Šāds lēmums dod iespēju agrāk paziņot lietotājam, ka kādam objektam ir vērts pievērst uzmanību. Potenciāli, mērķa atribūta vērtības prognozēšanai ir arī iespējams apvienot statistiku no vairākiem objektam tuvākiem klasteriem.

Prognozējamās mērķa atribūta vērtības precizitāte ir sistēmas veiktspējas galvenais rādītājs. Prognozes kļūdas mērīšanai tiek izmantota vidējā absolūta kļūda (*Mean Absolute Error*). Kā papildus rādītājus sistēmas veiktspējas noteikšanai, ir iespējams izmantot sistēmas *Loģisko kļūdu* un sistēmas *Klasterizācijas kļūdu*. Sistēmas loģiskā kļūda norāda cik gadījumos sistēma pieņēma loģiski nepareizus lēmumus. Lai to aprēķinātu, ir nepieciešams definēt loģiski pareizu un loģiski nepareizu lēmumus. Piemēroti risinājamam uzdevumam, loģiski pareizus un nepareizus lēmumus ir iespējams definēt šādi:

1. Pieņemsim, ka laika rindas garums ir  $g_l$ , bet mērķa atribūta vērtība ir  $p = p_l$ , kur  $p_l > g_l$ . Šis pieņēmums norāda uz to, ka produkts nav pārgājis uz nākamo PDC fāzi. Līdz ar to loģiski pareizi būs prognozēt  $r_l > g_l$ , un loģiski nepareizi  $r_l \leq g_l$ ;
2. Pieņemsim, ka laika rindas garums ir  $g_l$ , bet mērķa atribūta vērtība ir  $p = p_l$ , kur  $p_l \leq g_l$ . Šis pieņēmums norāda uz to, ka produkts atrodas jau citā PDC fāzē. Līdz ar to loģiski pareizi būs prognozēt  $r_l \leq g_l$ , un loģiski nepareizi prognozēt  $r_l > g_l$ .

Apgalvojumu, ka gadījumā, kad  $r_l = g_l$ , produkts atrodas jau citā PDC fāzē, ir iespējams uzskatīt par pareizu, tāpēc kā datu pieejamība laika rindas  $l$  periodā norāda uz to, ka periods ir loģiski pabeigts un līdz ar to visas pārejas, ja kādas bija, ir notikušas.

Loģisko kļūdu nav iespējams izmantot par pamatmēru sistēmas precizitātes mērīšanai, jo tā nesniedz pietiekoši daudz informācijas par sistēmas precizitāti, bet to ir iespējams pielietot papildus informācijas iegūšanai par sistēmas funkcionalitāti.

Klasterizācijas kļūda norāda uz to, cik precīzi apmācības kopas dati tika sadalīti klasteros. Klasterizācijas kļūda vispirms tiek aprēķināta katram klasterim un tad tiek summēta visai sistēmai, kļūdas aprēķināšanas algoritms ir šāds:

1. Katram objektam  $o_{i,j}$  klasterī  $c_j$  aprēķināts attālums  $d(o_{i,j}, c_j)$  līdz klastera centram. Pielietojot mērus, aprakstītus 3.2.1. sadaļā, pilnais attālums būs atkarīgs no izmantoto dimensiju skaita (*MEuclidean*) vai no soļu skaita objektu salīdzināšanas ceļā (*DTW*). Tāpēc aprēķinot klasterizācijas kļūdu, kā attālums starp objektu  $o_{i,j}$  un klastera  $c_j$  centru, tiek ņemts kopējais attālums;
2. Katram klasterim  $c_j$  zināšanu bāzē summējot attālumus  $d(o_{i,j}, c_j)$ , tiek iegūta summētā klasterizācijas kļūda;
3. Summējot katra klastera  $c_j$  klasterizācijas kļūdu, tiek iegūta sistēmas summētā klasterizācijas kļūda.

Minētajā algoritmā ir iespējams mainīt otro soli, klastera summētas klasterizācijas kļūdas vietā rēķinot vidējo klasterizācijas kļūdu katram objektam  $o_{i,j}$  klasterī  $c_j$ . Šajā gadījumā sistēmā tiek iegūta summētā vidēja klasterizācijas kļūda, proporcionāli uz vienu dimensiju, lietojot *MEuclidean* mēru, vai proporcionāli uz vienu soli objektu salīdzināšanas ceļā, lietojot *DTW* mēru.

### Trešās nodaļas kopsavilkums

1. Balstoties uz datu klasterizācijas algoritmiem izvirzīto prasību analīzi, tika atlasīti klasterizācijas algoritmi turpmākai pielietošanai darbā.
2. Pētot un analizējot atlasītos klasterizācijas algoritmus, tika konstatētas algoritmu nepilnības, kuru dēļ nav iespējams klasterizēt īsas laika rindas ar atšķirīgu periodu skaitu.
3. Nodaļā ir piedāvāti attāluma mēri, kas ļauj aprēķināt attālumu starp īsām laika rindām ar atšķirīgu periodu skaitu.
4. Izstrādāta pašorganizējošo neironu tīklu algoritma modifikācija, kas spēj apstrādāt un klasterizēt īsas laika rindas ar atšķirīgu periodu skaitu.
5. Izstrādāta hierarhiska gravitācijas klasterizācijas algoritma modifikācija, kas spēj apstrādāt un klasterizēt īsas laika rindas ar atšķirīgu periodu skaitu.
6. Balstoties uz lēmumu atbalsta aģentam izvirzītajām prasībām un izstrādātās LAA iekšējo procesu shēmas (sk. 2.1.3. sadaļu), tika izstrādāta produkta dzīves cikla fāzes pārejas perioda prognozēšanas procedūra.

### Ceturtnā nodaļa

Izstrādātas multiaģentu sistēmas veiktspējas novērtēšanai tika izstrādāts specializēts programmnodrošinājums, kas realizē multiaģentu sistēmas struktūru un datu plūsmas starp aģentiem, klasterizācijas algoritmus un to parametru izmaiņas iespējas, sistēmas apmācību un pārbaudi ar dažādiem datiem, apkopojot pārbaudes rezultātus viegli uztveramā formātā. Programmatūras realizācijai tika izmantota programmēšanas vide *Microsoft Visual Studio*

2008, pēdējie programmatūru uzlabojumi tika izpildīti jau *Microsoft Visual Studio 2010* vidē. Tika izmantota *Visual Basic* programmēšanas valoda. Nodaļa satur izstrādātā programmnodrošinājuma funkcionalitātes un lietotāja saskarnes aprakstus.

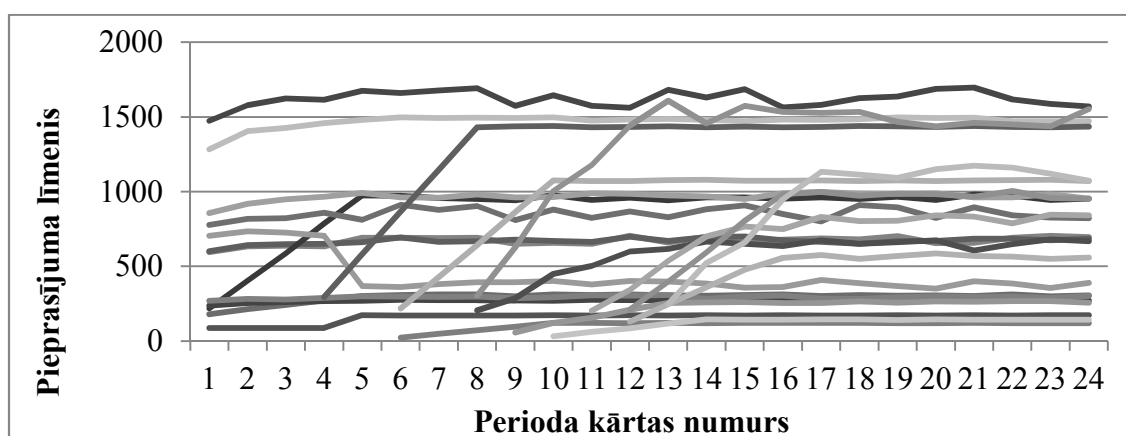
## Piektā nodaļa

### 5.1. Izmantotas datu kopas

Praktiskajos eksperimentos izmantotie dati tika iegūti starptautiskā projekta *ECLIPS* [56] ietvaros, un saturēja 1000 ieviešanas fāzes pieprasījuma līknes starptautiskas ķīmiskas ražošanas kompānijas *HUNTSMAN* [58] produkcijai. Līknes bija ar garumu līdz 23 periodiem, nenormalizētas, saturēja trūkstošus un trokšņainus datus, un nebija izlīdzinātas laikā. Attēlā 5.1. ir parādīts neapstrādāto datu piemērs.

Vairākumā no laika rindām līknes profili ir līdzīgi sākuma periodos, kas sistēmai apmācības posmā var palielināt prognozes kļūdu. Mērķa atribūta prognozes entropijas samazināšanai tika ieviests laika rindas minimālā garuma nosacījums – laika rindā jābūt vismaz 4 periodiem. Laika rindas tika izlīdzinātas pēc sākuma perioda – laika rindas sākas vienā laika momentā. No datu kopas tika izslēgti ieraksti ar trūkstošiem datiem, atlikušās laika rindas tika normalizētas, pielietojot *Z-novērtējuma normalizācijas metodi ar standartnovirzi*.

Pirmapstrādes procesa beigās tika iegūta analīzei sagatavotā datu kopa, kas saturēja 312 normalizētas, īsas laika rindas ar garumu no 4 līdz 23 periodiem, kas attēloja produkta pieprasījumu ieviešanas fāzē. Katras laika rindas pēdējā perioda kārtas numurs tika izmantots kā marķieris, kas norāda pēc kura perioda produkts mainīja ieviešanas fāzi uz brieduma fāzi.



5.1. att. Pamata datu piemērs

Tika izmantota arī sintētisko datu kopa, kurai tika mākslīgi samazināts trokšņa līmenis. Tā tika izveidota, izmantojot no pamata datu kopas atlasītas 60 parauga līknes ar garumu no 4 līdz 23 periodiem. Izmantojot formulu (5.1), tika ģenerētas 400 jaunas līknes:

$$x'_i = x_i (1 - \eta + 2\eta \cdot RND()), \quad (5.1)$$

kur  $\eta$  - robežas noteicošais koeficients, kas sintētisku datu kopas ģenerēšanas procesā bija vienāds ar 0.2, t.i. 20%.

### 5.2. Eksperimentu realizācija

Apakšnodaļā ir aprakstīts izstrādātu multiaģentu sistēmu veiktspējas novērtēšanai nepieciešamo eksperimentu kopas formēšanas process. Sistēmai ar katru no izmantojamām klasterizācijas metodēm (sk. 3.2. apakšnodaļu) ir aprakstīts veiktspējas novērtēšanas process, apkopoti un grafiski attēloti iegūtie eksperimentu rezultāti.

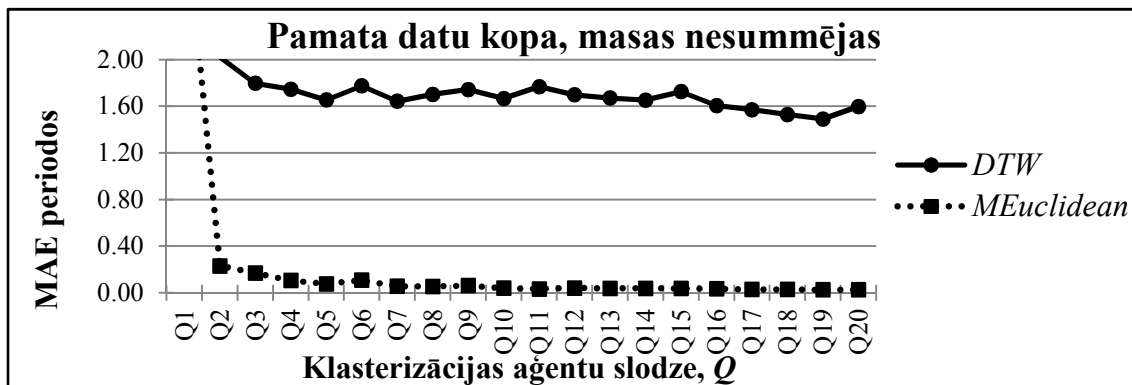
Tas, ka vērtējamo multiāģentu sistēmu galvenajiem parametriem ir galīgs iespējamo vērtību skaits, dod iespēju iegūt sistēmu novērtēšanai nepieciešamo eksperimentu skaitu. Abus algoritmus ir paredzēts pārbaudīt ar divām datu kopām, pielietojot 10-kārtīgu šķērsvalidāciju un divus attāluma mērus – *DTW* un modificēto *Eiklīda* attālumu (*MEuclidean*). Pārbaudījumu laikā klasterizācijas aģentu slodze mainīsies robežās no 1 līdz 20, t.i.  $Q \in \{1, 2, \dots, 20\}$ . Kopumā sanāk 800 eksperimenti.

Multiāģentu sistēmai ar *SOM* bāzes papildus tika pārbaudītas trīs neironu tīklu topoloģijas, kas palielina eksperimentu skaitu līdz 2400. Sistēmai, kas izmanto hierarhisko gravitācijas klasterizācijas algoritmu (*GC*), ir maināms tikai viens parametrs, kas norāda uz to, ka objektu masas pēc apvienošanas tiks summētas vai arī nē, kas palielina eksperimentu skaitu līdz 1600. Kopumā ir nepieciešams izpildīt 4000 eksperimentu, lai apskatītu visas galveno parametru vērtību kombinācijas. Kopējo eksperimentu skaitu ir iespējams samazināt, balstoties uz novērtētas apmācības kļūdas.

### 5.2.1. Sistēmas apmācības kļūdas novērtēšana un eksperimentu kopas definēšana

Balstoties uz informācijas literatūras avotos [16, 23, 61] un empīriski iegūtas pieredzes, tika fiksētas dažu parametru vērtības katrai no izstrādātajām klasterizācijas algoritmu modifikācijām. Tās vērtības tika izmantotas sistēmas apmācības kļūdas aprēķināšanas procesā un turpmāk darbā netika mainītas, novērtējot sistēmu pārbaudes kļūdu.

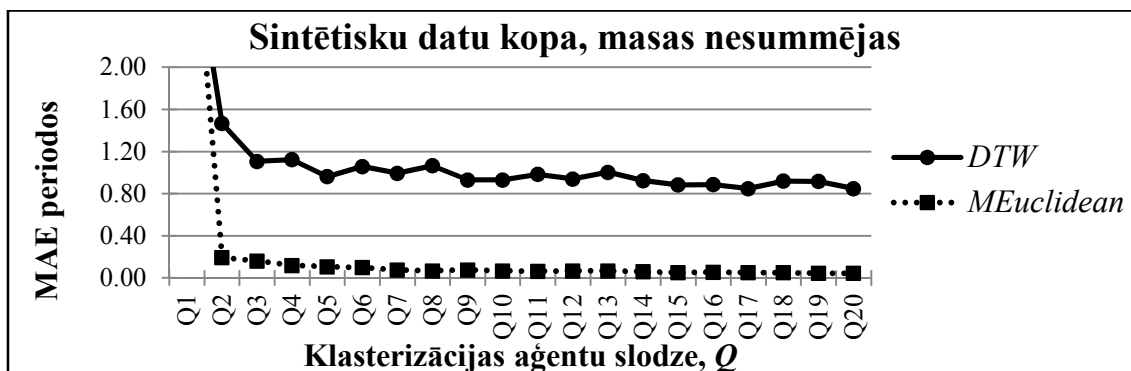
Sistēmas ar gravitācijas klasterizācijas algoritmu apmācības kļūdas novērtēšanai tika sagatavots eksperimentu plāns no 160 sistēmas palaišanas gadījumiem. Attēlos 5.2. un 5.3. ir parādīts apmācības kļūdas novērtējums pamata un sintētisko datu kopām gadījumam, kad objektu masas netika summētas pēc to apvienošanas. Līdzīga situācija ir gadījumam, kad masas tika summētas – mazākā apmācības kļūda tiek iegūta, pielietojot *MEuclidean* attāluma mēru, kas ļauj atteikties no *DTW* attāluma mēra izmantošanas, pārbaudot sistēmu ar *GC*.



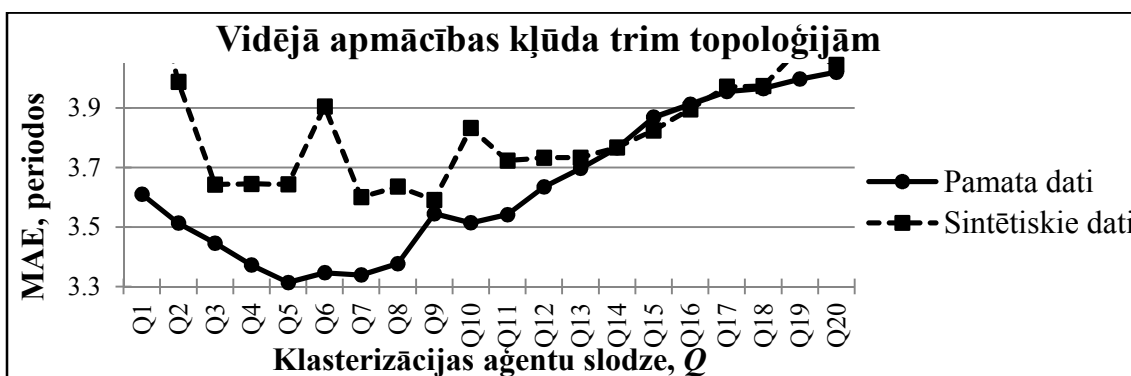
5.2. att. Sistēmas ar *GC* apmācības kļūda (bez objektu masu summēšanas)

Variāntam ar masu summēšanu ir trūkums. Apvienojoties objektiem, palielinās kumulatīvā objekta masa, kas var novēst pie tā, ka objekta masai būs lielākā ietekme nekā subjektu tuvībai. Tas atļauj atteikties no varianta ar masu summēšanu turpmākas izmantošanas. Apmācības kļūdas samazināšanās tendence, palielinoties klasterizācijas aģentu parametra vērtībai, ļauj atlasīt samazināto parametra  $Q$  iespējamo vērtību apakškopu: pamata datu kopai -  $Q \in \{16, 17, 18, 19, 20\}$ ; sintētisku datu kopai -  $Q \in \{15, 17, 18, 19, 20\}$ . Sistēma, kas izmanto *SOM* klasterizācijas algoritmu, tika pārbaudīta, pielietojot abus *DTW* un *MEuclidean* attāluma mērus, un izmantojot trīs topoloģijas – kvadrāta, rombveida un lineāro topoloģiju. Attēlā 5.4. ir parādīts sistēmas apmācības kļūdas novērtējums vidēji trim topoloģijām, pielietojot *MEuclidean* attāluma mēru. No tā var redzēt, ka apmācības kļūdas līmenis ir krietni lielāks, nekā sistēmai ar *GC*, un var secināt, ka arī pārbaudes kļūdas līmenis būs lielāks, nekā sistēmai ar *GC*. Tas dod iespēju samazināt klasterizācijas aģentu slodzes

vērtību skaitu, pie kurām tiks veikta pārbaude. Tiks izmantotas trīs vērtības ar minimālo kļūdu: pamata datu kopai -  $Q \in \{5, 6, 7\}$ ; sintētisko datu kopai -  $Q \in \{7, 8, 9\}$ .

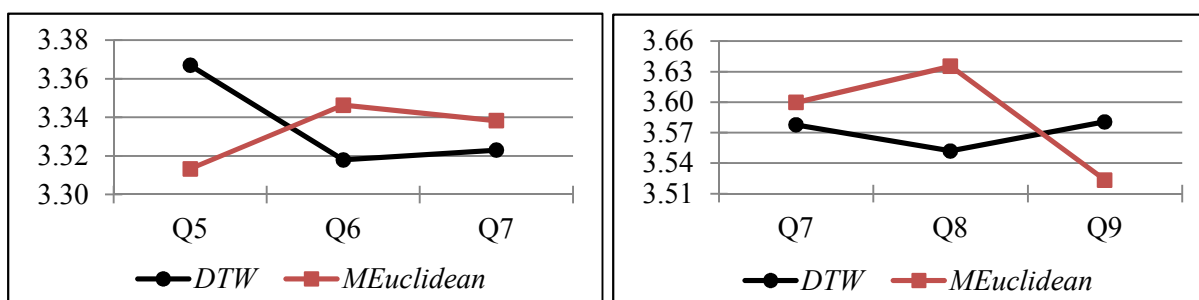


5.3. att. Sistēmas ar GC apmācības kļūda (bez objektu masu summēšanas)



5.4. att. Sistēmas ar SOM apmācības kļūda pie vidējas trim topoloģijām vērtības, lietojot MEuclidean attāluma mēru

Attēlā 5.5. ir parādīts sistēmas ar SOM apmācības kļūdas novērtējums. Rezultāti parāda, ka abiem attāluma mēriem apmācības kļūda sistēmai ar SOM atrodas vienā līmenī, līdz ar to turpmāk tiks pielietoti abi attāluma mēri.



a) Pamata datu kopai

b) Sintētisku datu kopai

5.5. att. Sistēmas ar SOM apmācības kļūda periodos vidēji trim topoloģijām

### 5.2.2. Veiktspējas novērtēšana multiāģentu sistēmai ar SOM

Sistēma tika pārbaudīta ar divām datu kopām, izmantojot 10-kārtīgu šķērsvalidāciju. Tabula 5.1. apkopo rezultātus, pārbaudot sistēmu ar pamata datu kopu. Ar treknrakstu ir izdalīti labākie rezultāti katrai no izmantotajām neironu tīklu topoloģijām.

Sistēmas ar *SOM* vidējās absolūtās kļūdas periodos pārbaude ar pamata datu kopu

Klasterizācijas aģentu slodze	Topoloģija ar 2 kaimiņiem		Topoloģija ar 4 kaimiņiem		Topoloģija ar 8 kaimiņiem	
	<i>DTW</i>	<i>MEuclidean</i>	<i>DTW</i>	<i>MEuclidean</i>	<i>DTW</i>	<i>MEuclidean</i>
$Q = 5$	3.475	3.447	3.362	3.437	3.434	3.578
$Q = 6$	<b>3.267</b>	<b>3.267</b>	<b>3.332</b>	<b>3.404</b>	<b>3.353</b>	<b>3.353</b>
$Q = 7$	3.344	3.411	3.372	3.513	3.371	3.509

Apkopjot iegūtus rezultātus var secināt, ka sistēmas labākais prognozes rezultāts – 3.267 periodi, bija sasniegts, pielietojot lineāru neironu tīkla topoloģiju ar 2 kaimiņiem, izmantojot *DTW* un *MEuclidean* attāluma mērus ar klasterizācijas aģentu slodzi  $Q = 6$ . Tabula 5.2. apkopo rezultātus, pārbaudot sistēmu ar sintētisku datu kopu. Ar treknrakstu ir izdalīti labākie rezultāti katrai no izmantotajām neironu tīklu topoloģijām.

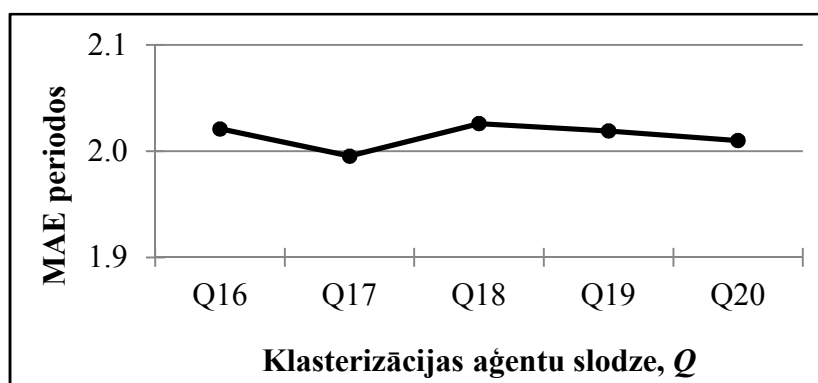
Sistēmas ar *SOM* sintētisku datu kopas vidējās absolūtās kļūdas periodos pārbaude

Klasterizācijas aģentu slodze	Topoloģija ar 2 kaimiņiem		Topoloģija ar 4 kaimiņiem		Topoloģija ar 8 kaimiņiem	
	<i>DTW</i>	<i>MEuclidean</i>	<i>DTW</i>	<i>MEuclidean</i>	<i>DTW</i>	<i>MEuclidean</i>
$Q = 7$	3.521	<b>3.889</b>	3.553	<b>3.889</b>	3.657	<b>3.910</b>
$Q = 8$	<b>3.512</b>	3.925	<b>3.494</b>	3.925	<b>3.565</b>	4.000
$Q = 9$	3.575	4.017	3.553	4.090	3.653	4.135

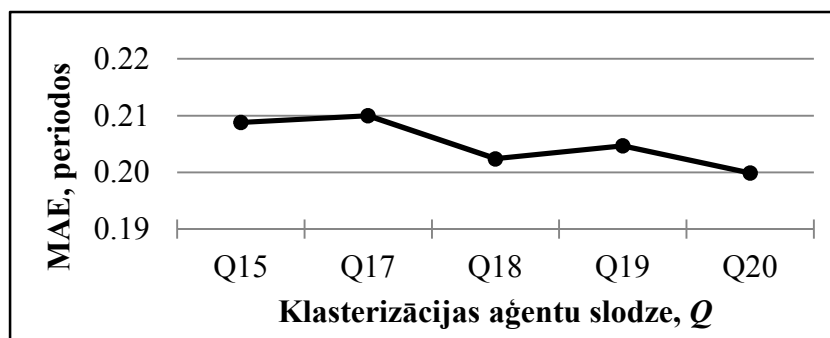
Labākie sistēmas ar *SOM* pārbaudes rezultāti ar sintētisku datu kopu tika iegūti, lietojot *DTW* attāluma mēru topoloģijai ar 4 kaimiņiem un klasterizācijas aģentu slodzi  $Q = 8$ , kas sastādīja 3.494 periodus.

5.2.3. Veiktspējas novērtēšana multiāģentu sistēmai ar *GC*

Sadaļā ir apkopota informācija par sistēmas ar *GC* veiktspējas pārbaudes rezultātiem, izmantojot 10-kārtīgu šķērsvalidāciju. Attēlā 5.6. ir parādīti pārbaudes ar pamata datu kopu rezultāti sistēmai ar gravitācijas klasterizāciju. Sistēmas kļūda svārstās 2 periodu līmenī, precīzākais rezultāts 1.995 periodi tika iegūts pie klasterizācijas aģentu slodzes  $Q$  vērtības, kas vienāda ar 17.

5.6. att. Sistēmas ar *GC* pārbaudes kļūda pamata datu kopai

Attēlā 5.7. ir parādīti pārbaudes rezultāti sistēmai ar gravitācijas klasterizāciju sintētisku datu kopai. Sistēmas vidējā absolūtā kļūda svārstās līmenī 0.2 periodi un sasniedz minimālo vērtību 0.200 periodi pie klasterizācijas aģentu slodzes parametra  $Q$  vērtības 20.



5.7. att. Sistēmas ar GC pārbaudes kļūda periodos sintētisku datu kopai

### 5.3. Rezultātu apkopojums un analīze

Pārbaudot sistēmu ar *SOM* ar pamata datu kopu, labākais rezultāts 3.267 periodi tika iegūts, izmantojot lineāro topoloģiju un klasterizācijas aģentu slodzi  $Q = 6$ . Attāluma mēra izvēle neietekmēja rezultātus – abiem attāluma mēriem rezultāts sastādīja 3.267 periodi. Skaitot pie trim topoloģijām vidējās vērtības, labākais rezultāts sastādīja 3.317 periodus un tika iegūts lietojot *DTW* attāluma mēru un klasterizācijas aģentu slodzi  $Q = 6$ . Labākais rezultāts pie trim topoloģijām vidējās vērtības, lietojot attāluma mēru *MEuclidean* un to pašu klasterizācijas aģentu slodzi, sastādīja 3.341 periodus. Gan labāko, gan trim topoloģijām vidējo rezultātu var uzskatīt par samērā lielu kļūdu, ievērojot to, ka par periodu ir ņemts viens mēnesis. Pārbaudot ar pamata datu kopu sistēmu ar hierarhisko gravitācijas klasterizācijas algoritmu, mazākā prognozēšanas kļūda, kas tika sasniegta, ir 1.995 periodi. Šis rezultāts tika iegūts izmantojot *MEuclidean* attāluma mēru un klasterizācijas aģentu slodzes parametra  $Q$  vērtību vienādu ar 17. Prognozēšanas kļūdas līmenis ir zemāks, nekā sistēmai ar pašorganizējošiem neironu tīkliem, bet tomēr paliek 2 periodu robežās. Kā loģisks skaidrojums šādai situācijai varētu būt tas, ka dati no pamata datu kopas tika saņemti no veiksmīgi funkcionējoša un attīstīta uzņēmuma. Savukārt, produktu dzīves cikla fāzes pārejas marķieru vērtības definēja uzņēmuma ekspertu grupa, balstoties uz personisko pieredzi un tirgū esošās situācijas, kas ievieš papildus haotiskumu, ar to palielinot entropiju datus. Ņemot vērā aprakstīto situāciju, iegūto prognozes kļūdu 1.995 periodi ir iespējams uzskatīt par pietiekoši zemu un secināt, ka sistēmai ar hierarhisko gravitācijas klasterizācijas algoritmu ir salīdzinoši laba veikspēja.

Pārbaudot sistēmas ar datiem no sintētisku datu kopas, sistēmai ar pašorganizējošiem neironu tīkliem labākais rezultāts tika iegūts izmantojot rombveida topoloģiju, *DTW* attāluma mēru un klasterizācijas aģentu slodzi  $Q = 8$  un tas sastādīja 3.494 periodus. Savukārt, sistēmai ar hierarhisko gravitācijas klasterizācijas algoritmu kļūda sastādīja tikai 0.2 periodus, lietojot *MEuclidean* attāluma mēru un klasterizācijas aģentu slodzi  $Q = 20$ .

Analizējot sistēmu pārbaudes rezultātus pamata datu kopai un sintētisku datu kopai, var izdalīt šādu likumsakarību, ka sistēmai ar hierarhisko gravitācijas klasterizācijas algoritmu labākie rezultāti tika iegūti pie maksimālas klasterizācijas aģentu slodzes ( $Q$ ) vērtībai tuvām vērtībām:  $Q = 17$  pamata datu kopai un  $Q = 20$  sintētisku datu kopai, kā arī tika ievērota tendence prognozēšanas kļūdas samazināšanai palielinoties klasterizācijas aģentu slodzei. Šādu likumsakarību izraisīja tas, ka pie minētām parametra  $Q$  vērtībām klasterizācijas aģentu saimē tiek izveidoti divi vai viens klasterizācijas aģents. Rezultātā klasterizācijas aģentam kļūst pieejama lielāka datu kopas daļa, kas kopā ar secīgu datu plūsmas imitāciju ietekmē to, ka klasteru saraksta veidošanas procesā savāktā iespējamo mērķa atribūta vērtību statistika

klūst precīzāka. Praktiski, iespējamo mērķa atribūta vērtību skaits vienā klasterī samazinās līdz 1-2, kas labvēlīgi ietekmē sistēmas veiktspēju.

Savukārt, sistēmai ar pašorganizējošiem neironu tīkliem precīzākie rezultāti tika iegūti pie šādām klasterizācijas aģentu slodzes parametra vērtībām:  $Q = 6$  pamata datu kopai un  $Q = 8$  sintētisku datu kopai, kā arī netika konstatēta tendence prognozēšanas kļūdai samazināties, palielinoties klasterizācijas aģentu slodzei. Tādas pašorganizējošo neironu tīklu rezultātu atšķirības no rezultātiem, iegūtiem ar hierarhisko gravitācijas klasterizācijas algoritmu, var būt skaidrojamas ar konceptuālām atšķirībām minētajos klasterizācijas algoritmos. Pašorganizējošie neironu tīkli, klasterizējot datus, cenšas samazināt klasterizācijas kļūdu. Tas ir definēts pašorganizējošo neironu tīklu algoritma neironu svaru adaptācijas procesā. Tāpēc rezultātā veidojas klasteri ar salīdzinoši zemu klasterizācijas kļūdu, kuri ne vienmēr satur precīzu mērķa atribūta vērtību statistiku, kas tika eksperimentāli pierādīts 5.2.2. sadaļā. Savukārt, hierarhiskais gravitācijas klasterizācijas algoritms izmanto citu stratēģiju. Algoritms secīgi pārvieto un apvieno objektus, vadoties pēc attāluma starp objektiem. Objektiem apvienoties veidojas klasteri kā ar mazu, tā arī ar salīdzinoši lielu klasterizācijas kļūdu. Savukārt, datu secīga apvienošana kopā ar darbā piedāvāto *MEuclidean* attāluma mēru veicina to, ka veidojas klasteri ar precīzāku mērķa atribūta vērtību statistiku, kas tika eksperimentāli pierādīts 5.2.3. sadaļā.

Salīdzinot sistēmas pēc apmācības kļūdas, izteikti dominē sistēma ar hierarhisko gravitācijas klasterizācijas algoritmu. Šīs sistēmas apmācības kļūda tiecas uz nulli un samazinās palielinoties klasterizācijas aģentu slodzes parametra  $Q$  vērtībai. Sistēmai ar hierarhisko gravitācijas klasterizācijas algoritmu izteikti dominē darbā piedāvātais *MEuclidean* attāluma mērs, pretēji *DTW* mēram. Šo likumsakarību iespējams izskaidrot ar to, ka datus tiek imitēta secīgu datu plūsma, kas kopā ar darbā piedāvāto *MEuclidean* attāluma mēru veicina klasteru izveidi ar precīzāku mērķa atribūta vērtību statistiku.

Iegūtie sistēmu pārbaudes rezultāti ar sintētisku datu kopu ļauj secināt par sistēmu robustuma līmeni – noturīgumu pret trokšņainiem datiem. Kā redzams no rezultātiem sistēmai ar pašorganizējošiem neironu tīkliem (sk. 5.2.2. sadaļu), mērķa atribūta vērtības prognozēšanas kļūda ir salīdzinoši liela, tomēr prognozēšanas kļūdas līmenis gan pamata, gan sintētisku datu kopai ir praktiski vienāds. Tas ļauj apgalvot, ka pašorganizējošie neironu tīkli ir pietiekoši noturīgi pret trokšņainiem datiem, jo sintētisku datu kopā trokšņu līmenis bija mākslīgi samazināts, salīdzinot ar pamata datu kopu. Tajā pat laikā sistēma ar hierarhisko gravitācijas klasterizācijas algoritmu asāk reaģēja uz trokšņa līmeņa izmaiņām datus – pamata datu kopai prognozes kļūda palielinājās gandrīz 10 reizēs (sk. 5.2.3. sadaļu), salīdzinoši ar rezultātu, kas iegūts sintētisku datu kopai, kurai bija mazāks trokšņa līmenis.

Balstoties uz izpildītas analīzes var secināt, ka hierarhiskās klasterizācijas metodes ir mazāk noturīgas pret trokšņainiem datiem, nekā pašorganizējošie neironu tīkli, bet precīzāk sadala datus grupās, ļaujot prognozēt mērķa atribūta vērtību ar pieļaujamo kļūdas līmeni. Noslēdzot šo apakšnodaļu jāsecina, ka darbā iegūtie rezultāti un izteiktie apgalvojumi par sistēmām savā vairākumā ir virzīti uz darbā definētiem uzdevumiem un izmantojamiem datiem.

## SECINĀJUMI

Promocijas darbā, izstrādātas produkta dzīves cikla vadības atbalsta multiaģentu sistēmas piemērā, izpētīti un izanalizēti intelektuālo aģentu tehnoloģijas īpašības un iespējas, kas piemērotas sarežģītu, sadalītu sistēmu projektēšanai un modelēšanai. Darbā piedāvātā multiaģentu sistēma tika izstrādāta specifiskas problēmas risināšanai, precīzāk – PDC fāzes pārejas perioda prognozēšanai secīgu datu plūsmas apstākļos. Tomēr, balstoties uz intelektuālo aģentu tehnoloģijas, sistēmai tika izstrādāta pietiekoši vienkārša struktūra, kuru var viegli pielāgot citu līdzīgu uzdevumu risināšanai. Sistēmā iekļauto aģentu struktūras un procesu sadalīšanas stratēģijas ļauj viegli ieviest izmaiņas aģentu darbības procesā, nemainot

sistēmu konceptuāli. Multiaģentu sistēmas izstrādes procesā iegūtie rezultāti apstiprina pirmo no ievadā izvirzītām hipotēzēm par sarežģītas vadības un vadības atbalsta sistēmu projektēšanas procesa vienkāršošanu izmantojot intelektuālo aģentu tehnoloģiju.

Datu ieguves metožu kombinēšana sniedza iespējas izstrādāt zināšanu ieguves un mērķa atribūta prognozēšanas sadalīto procesu, piemēroti promocijas darbā definētiem uzdevumiem. Datu ieguves metožu pielāgojamība ļāva izstrādāt klasterizācijas metožu iespēju paplašinošas modifikācijas. Izstrādātās sistēmas veikspējas pārbaudes procesā savākie rezultāti apstiprina otro ievadā izvirzīto hipotēzi par iespējamību, pielietojot datu ieguves metodes, izveidot produkta dzīves cikla paraugu modeli, paredzētu jaunu produktu analīzei. Promocijas darba zinātniskie un praktiskie rezultāti ir:

1. Izstrādāta produktu dzīves cikla vadības atbalsta multiaģentu sistēma, kas nodrošina sistēmas lietotāju ar papildus informāciju par produkcijas stāvokli tirgū un par situācijas iespējamiem attīstības virzieniem, kas veicina argumentēto lēmumu pieņemšanu par ražošanas plānošanas, tirdzniecības attīstīšanas un reklāmas stratēģijām. Izejot no iegūtiem rezultātiem var secināt, ka apvienojot intelektuālo aģentu tehnoloģiju un datu ieguves metodes ir iespējams izstrādāt sarežģītu vadības sistēmu.
2. Izpildīta intelektuālo aģentu tehnoloģijas pamatprincipu un koncepciju analīze, kuras rezultāti norāda uz to, ka intelektuālo aģentu tehnoloģijas pielietošana var atvieglot sarežģītu, sadalītu sistēmu modeļa projektēšanas procesu.
3. Balstoties uz promocijas darbā definētiem uzdevumiem un izteiktām prasībām multiaģentu sistēmai, izstrādāti un aprakstīti funkcionēšanas algoritmi piedāvātās sistēmas aģentiem – datu vadības aģents, datu ieguves aģents un lēmumu atbalsta aģents. Rezultāti ļauj secināt, ka intelektuālo aģentu tehnoloģija veicina sistēmas iekšējo procesu precīzāku formalizāciju un strukturēšanu.
4. Izpildīta populāro klasterizācijas metožu analīze pielietojamībai pieprasījuma datu klasterizācijai. Izpētīti un aprakstīti uzdevumam piemēroto pašorganizējošo neironu tīklu un hierarhiskā gravitācijas klasterizācijas algoritma procedūru pamatprincipi un koncepcijas. Izpildīta minēto algoritmu pielietojamības analīze īsu laika rindu ar atšķirīgu periodu skaitu klasterizācijai. No iegūtiem rezultātiem var secināt, ka pašorganizējošo neironu tīklu un hierarhiskā gravitācijas klasterizācijas algoritma klasiskie varianti nenodrošina iespēju klasterizēt laika rindas ar atšķirīgu garumu.
5. Izstrādātas pašorganizējošo neironu tīklu algoritma un hierarhiskā gravitācijas klasterizācijas algoritma modifikācijas, kas nodrošina īsu laika rindu ar atšķirīgo periodu skaitu klasterizācijas iespēju. Algoritmos ir izdarītas kā strukturālās, tā arī funkcionālās izmaiņas.
6. Piedāvāts *MEuclidean* attāluma mērs, kas paredz attālumu noteikšanu laika rindām ar atšķirīgu periodu skaitu.
7. Izstrādāts programmnodrošinājums, kas realizē promocijas darbā piedāvātas multiaģentu sistēmas konceptuālo un procesuālo struktūru un implementē izstrādātas pašorganizējošo neironu tīklu algoritma un hierarhiskā gravitācijas klasterizācijas algoritma modifikācijas.
8. Ar izstrādātā programmnodrošinājuma palīdzību izpildīta piedāvātas multiaģentu sistēmas aprobācija un salīdzinošā analīze, salīdzinot izstrādātas klasterizācijas algoritmu modifikācijas un attālumu mērus. Sistēmas veikspējas novērtēšanas rezultāti parādīja, ka piedāvātā multiaģentu sistēma ir spējīga automātiskā režīmā izveidot datus pastāvošo saišu modeli un pielietot to mērķa atribūtu vērtības prognozēšanai jauniem datiem ar salīdzinoši augstu precizitāti.

9. Balstoties uz promocijas darbā piedāvātas multiāģentu sistēmas aprobācijas un veikspējas novērtēšanas rezultātu analīzes secināts, ka pielietojot datu ieguves metodes ir iespējams izveidot produkta dzīves cikla paraugu modeli, paredzētu jaunu produktu dzīves cikla analīzei.

## LITERATŪRAS SARAKSTS

1. *S.Paršutins, A.Sukovs, G.Kuļešova, A.Borisovs, L.Aleksejeva Datu ieguve: Programmētāji. – Rīga: Rīgas Tehniskā universitāte, SIA „Latgales Druka”, 2008. – 111 lpp. ISBN: 978-9984-3798-68-4.*
2. *A.Sukovs, L.Aleksejeva, K.Makejeva, A.Borisovs. Datu ieguve: Pamati. – Rīga: Rīgas Tehniskā universitāte, SIA „Drukātava”, 2006. – 130 lpp. ISBN: 978-9984-798-19-6.*
3. Adeyemi S.L. Just-in-Time Production Systems (JITPS) in Developing Countries: The Nigerian Experience // *Journal of Social Sciences.* – Vol.22, No.2 (2010), pp. 145-152.
4. Aitken J., Childerhouse P., Towill D. The impact of product life cycle on supply chain strategy // *International Journal of Production Economics.* – Vol.85, No.2 (2003), pp. 127-140.
5. Al-zu`bi H. Applying Electronic Supply Chain Management Using Multi-Agent System: A Managerial Perspective // *International Arab Journal of e-Technology.* – Vol.1, No.3 (2010), pp. 106-113.
6. Athanasiadis I.N., Mitkas P.A. Software Agents for Assessing Environmental Quality: Advantages and Limitations // *Proceedings of 18th International Conference on Informatics for Environmental Protection: EnviroInfo 2004, Switzerland, Geneva, October 2004.* – pp. 303-305.
7. Boukhelfa K., Boufaida M. A Generic Multi-Agent Architecture for the Virtual Enterprise // *Lecture Notes in Informatics.* – Vol.P-56, (2004), pp. 199-210. ISBN 3-88579-385-7.
8. *Chaib-Draa B., Muller J.P., (Eds.). Multiagent based Supply Chain Management. – Springer-Verlag Berlin Heidelberg, 2006. – 450 p. ISBN: 978-3-540-33875-8.*
9. Chang-Hong J., Xi-Hai D. Forecasting Demand of Short Life Cycle Products Based on Modified Bass Model // *Science Technology and Engineering.* – Vol.10, No.10 (2010), pp. 2577-2580. ISSN: 1671-1815.
10. Chien C.-F., Chen Y.-J., Peng J.-T. Manufacturing intelligence for semiconductor demand forecast based on technology diffusion and product life cycle // *International Journal of Production Economics.* – Vol.128, No.2 (2010), pp. 496-509.
11. CIMdata Inc. Product Lifecycle Management. CIMdata report. CIMdata Inc, 2002. – 9 p.
12. Cohen B. A New Approach to Strategic Forecasting // *The Journal of Business Strategy.* – Vol.9, No.5 (1988), pp. 38-43.
13. Cooper M.C., Maris F., Régnier P. Solving Temporally-Cyclic Planning Problems // *Proceedings of 17th International Symposium on Temporal Representation and Reasoning, Paris, France, 6 – 8 Sept. 2010.* – pp. 113-120. ISBN: 978-0-7695-4187-7.
14. *Dreyfus G. Neural Networks. Methodology and Applications. Second Edition. – Springer-Verlag Berlin Heidelberg, 2005. – 386 p. ISBN: 978-3-540-22980-3.*
15. *Dunham M.H. Data Mining. Introductory and Advanced Topics. – Prentice Hall, Pearson Education, 2003. – 315 p. ISBN: 0-13-088892-3.*

16. Gomez J., Dasgupta D., Nasraoui O. A New Gravitational Clustering Algorithm // Proceedings of the Third SIAM International Conference on Data Mining – May 1-3, San Francisco, USA, 2003. – pp. 83-94.
17. Grovez W., Collins J., Ketter W., Gini M. Analyzing Market Interactions in a Multi-Agent Supply Chain Environment // Lecture Notes in Business Information Processing. – Vol.52, (2010), pp. 44-58.
18. Han J., Kamber M. *Data Mining: Concepts and Techniques. Second Edition.* – Morgan Kaufmann, Elsevier Inc., 2006. – 800 p. ISBN: 978-1-55860-901-3.
19. Japan Management Association. *Kanban Just-in Time at Toyota: Management Begins at the Workplace.* - Productivity Press, 1989. – 224 p. ISBN: 9780915299485.
20. Jennings N. On agent-based software engineering // Artificial Intelligence. – Vol.117, (2000), pp. 277-296.
21. Keogh E.J., Pazzani M.J. Derivative Dynamic Time Warping // Proceedings of the First SIAM International Conference on Data Mining – April 5-7, Chicago, USA, 2001. – pp. 1-11.
22. Kirshners A., Parshutin S., Borisov A. Combining clustering and a decision tree classifier in a forecasting task // Automatic Control and Computer Science. - Vol.44, No.3 (2010), pp. 124-132. Citēts: SpringerLink.
23. Kohonen T. *Self-Organizing Maps. Third Edition.* – Springer-Verlag Berlin Heidelberg, 2001. – 501 p. ISBN: 3-540-67921-9.
24. Lapide L. New Developments in Business Forecasting // The Journal of Business Forecasting Methods and Systems. – Vol.21, No.2 (2002), pp. 11-14.
25. Ming H.X.G., Lu W.F. Zhu C.F. Technology Challenges for Product Lifecycle Management / Technical report STR/04/058/SP. Singapore Institute of Manufacturing Technology, 2004. – pp. 348-353.
26. Nicholas R., Mangili G. Short Product Life Cycle Planning // IOM Control Magazine. – Vol.30, No.1 (2004), pp. 24-26.
27. Parshutin S., Aleksejeva L., Borisov A. Forecasting Product Life Cycle Phase Transition Points with Modular Neural Networks Based System // Lecture Notes in Artificial Intelligence 5633, Advances in Data Mining, Germany, Leipzig, July 20-22, 2009. – pp. 88-102. Citēts: SpringerLink.
28. Parshutin S., Aleksejeva L., Borisov A. Time Series Analysis with Modular Neural Networks // Scientific Journal of Riga Technical University. Series 5, Computer Science. - Vol. 36, (2008), pp. 162-169. Citēts: EBSCO.
29. Parshutin S., Borisov A. Agents Based Data Mining and Decision Support System // Lecture Notes in Artificial Intelligence 5680, Agents and Data Mining Interaction, Hungary, Budapest, 11-11. May, 2009. – pp. 36-49. Citēts: SpringerLink.
30. Parshutin S., Borisov A. Agent-Based Production Planning Support System // Baltic Journal on Sustainability. Technological and Economic Development of Economy. – Vol.16, No.3 (2010), pp. 455-470. Citēts: InformaWorld (ERA).
31. Parshutin S., Borisov A. Data Mining Driven Decision Support // Polish Journal of Environmental Studies. - Vol.18, No.4A (2009), pp. 8-11.
32. Parshutin S. Clustering Time Series of Different Length Using Self-Organising Maps // Scientific Journal of Riga Technical University. Series 5, Computer Science. – Vol. 31, (2007), pp. 104-110.

33. Parshutin S., Kirshners A. Intelligent Agent Technology in Modern Production and Trade Management // Efficient Decision Support Systems: Practice and Challenges – From Current to Future / Book Chapter. INTECH, 2011, In Press, 22 p.
34. Parshutin S., Kuleshova G. Time Warping Techniques in Clustering Time Series // Proceedings of 14th International Conference on Soft Computing, MENDEL 2008, Czech Republic, Brno, June 18-20, 2008. – pp. 175-180.
35. Parshutin S. Managing Product Life Cycle with MultiAgent Data Mining System. // Lecture Notes in Artificial Intelligence 6171, Advances in Data Mining, Germany, Berlin, 2010. – pp. 308-322. Citēts: SpringerLink.
36. Pechoucek M., Vokrinek J., Becavar P. ExPlanTech: Multiagent Support for Manufacturing Decision Making // Intelligent Systems, IEEE. – Vol.20, No.1 (2005), pp. 67-74. ISSN: 1541-1672.
37. Pyle D. *Data Preparation for Data Mining*. - Morgan Kaufmann, Academic Press, 1999. – 540 p. ISBN: 978-1558-605-29-9.
38. Roy D., Anciaux D., Monteiro T., Ouzizi L. Multi-agent architecture for supply chain management // Journal of Manufacturing Technology Management. - Vol.15, No.8 (2004), pp. 745 – 755.
39. Rozenblatt F. The Perceptron: A probabilistic model for information storage and organization in the brain // Psychological Review. – Vol.65. (1958), pp. 386-408.
40. Russell S., Norvig P. *Artificial Intelligence – A modern approach, 2nd edition*. – Prentice Hall, 2003. – 1132 p. ISBN: 0137903952.
41. Saaksvuori A., Immonen A. *Product Lifecycle Management, 3rd edition*. – Springer, 2008. – 253 p. ISBN: 978-3-540-78173-8.
42. Salvador S., Chan Ph. FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space // Intelligent Data Analysis. – Vol.11, No.5 (2007), pp. 561-580.
43. Schmidt E., Dada M., Ward J., Adams D. Using Cyclic Planning to Manage Capacity at ALCOA. // Journal of Interfaces. – Vol.31, No.3 (2001), pp. 16-27.
44. Stark J. *Product Lifecycle Management: 21st Century Paradigm for Product Realisation*. – Springer-Verlag, 2005. – 441 p. ISBN: 978-1-85233-810-7.
45. Tan P.-N., Steinbach M., Kumar V. *Introduction to Data Mining*. – Addison Wesley, Pearson Education, 2006. – 769 p. ISBN: 0-321-32136-7.
46. Van de Capelle J.-P. An examination of new product diffusion models. – RIT Printing Industry Center, 2004. – 52 p.
47. Venkatesan R., Kumar V. A genetic algorithms approach to growth phase forecasting of wireless subscribers // International Journal of Forecasting. – Vol.18, No.4 (2002), pp. 625-646.
48. Weiss G. (Ed.). *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*. – The MIT Press, 1999. – 619 p. ISBN: 978-0262-73131-7.
49. Wen Z.-H., Chen J.-L., Chang Q., Hu J.-L. State-action pairs reduction for strong cyclic planning via state reachability. Proceedings of International Conference on Fuzzy Systems and Knowledge Discovery, FSKD'10, Yantai, Shandong, 10-12 Aug. 2010. – pp. 1917-1921. ISBN: 978-1-4244-5931-5.
50. Wooldridge M. *An Introduction to Multiagent Systems*. – John Wiley & Sons, LTD, 2005, 3rd edition. – 348 p. ISBN: 978-0471-49691-5.

51. Wright W.E. Gravitational clustering // Pattern Recognition Vol.9, (1977), pp. 151-166 Pergamon Press.
52. Xianhao X., Qizhi S. Forecasting for Products with Short Life Cycle Based on Improved Bass Model // Industrial Engineering and Management. – Vol.5, (2007), pp. 27-31. ISSN: 1007-5429.
53. Yelland P.M. A Model of the Product Lifecycle for Sales Forecasting. – Sun Microsystems, 2004. – 60 p.
54. Yelland P.M., Lee E. Forecasting Product Sales with Dynamic Linear Mixture Models. – Sun Microsystems, 2003. – 22 p.
55. Yongtong H., Ping L., Yuhong Y., Danian Z., Changchao M., Bode J., Shouju R. A Multiagent System for the Support of Concurrent Engineering. Proceedings of the IEEE International Conference on System, Man and Cybernetics, Beijing, China, 14-17 Oct., 1996. – pp. 959-964. ISBN: 0-7803-3280-6.
56. Vadības uzdevumiem un biznesa administrēšanai veltīta Interneta centra NetMBA WEB-lapa. / Internets. - <http://www.netmba.com/marketing/product/lifecycle/>. Resurss apskatīts 18.03.2011.
57. Official WEB-page of an international ECLIPS project. / Internets. – <http://www.eclipsproject.com>. Resurss apskatīts 17.01.2011.
58. Official WEB-page of an international company HUNTSMAN / Internets – <http://www.huntsman.com>. Resurss apskatīts 17.01.2011.
59. Гринглаз Л., Копытов Е. Математическая статистика с примерами решения на компьютере: Учеб. Пособие. – 2-е изд. – Рига: ВШЭК, 2002. - 326 с.
60. Паршутин С.В. Кластеризация временных рядов с применением карт самоорганизации. // Сборник научных трудов международной конференции «Интегрированные модели и мягкие вычисления в искусственном интеллекте» (IMSCAI'2007) – 28-30 мая, Коломна, Россия, 2007. – 465-472 с.
61. Хайкин С. Нейронные сети. Полный курс. Второе издание. Перевод с английского. – Вильямс, 2006. – 1104 с.