



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

INFORMACINIŲ SISTEMŲ KATEDRA

Justė Kareivaitė

**DUOMENŲ GAVYBOS METODŲ PANAUDOJIMAS VERSLO PROCESŲ
SIMULIACIJOS REZULTATŲ ANALIZEI**

**APPLICATION OF DATA MINING METHODS FOR ANALYSIS OF
BUSINESS PROCESS SIMULATION RESULTS**

Baigiamasis magistro darbas

Informacinių sistemų programų inžinerijos studijų programa, valstybinis kodas 6211BX017

Informacinių sistemų specializacija

Informatikos inžinerijos studijų kryptis

Vilnius, 2019

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEDRA

TVIRTINU
Katedros vedėjas

(Parašas)
Prof. dr. Dalius Mažeika
(Vardas, pavardė)

(Data)

Justė Kareivaitė

**DUOMENŲ GAVYBOS METODŲ PANAUDOJIMAS VERSLO PROCESŲ
SIMULIACIJOS REZULTATŲ ANALIZEI**
**APPLICATION OF DATA MINING METHODS FOR ANALYSIS OF
BUSINESS PROCESS SIMULATION RESULTS**

Baigiamasis magistro darbas

Informacinių sistemų programų inžinerijos studijų programa, valstybinis kodas 6211BX017
Informatikos inžinerijos studijų kryptis

Vadovas

Prof. dr. Olegas Vasilecas
(Moksl. laipsnis/pedag. vardas, vardas, pavardė) _____
(Parašas) (Data)

Konsultantas

Lekt. Tadas Vysockis
(Moksl. laipsnis/pedag. vardas, vardas, pavardė) _____
(Parašas) (Data)

Lietuvių kalbos konsultantas

Dr. Aušra Žemienė
(Moksl. laipsnis/pedag. vardas, vardas, pavardė) _____
(Parašas) (Data)

Vilnius, 2019

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINŲ SISTEMŲ KATEDRA

Informatikos inžinerijos studijų kryptis
Informacinių sistemų programų inžinerijos studijų programa,
valstybinis kodas 6211BX017

TVIRTINU
Katedros vedėjas

(Parašas)
Prof. dr. Dalius Mažeika
(Vardas, pavardė)

(Data)

BAIGIAMOJO MAGISTRO DARBO UŽDUOTIS

.....Nr.
Vilnius

Studentei Justei Kareivaitei

Baigiamojo darbo tema:

**DUOMENŲ GAVYBOS METODŲ PANAUDOJIMAS VERSLO PROCESŲ
SIMULIACIJOS REZULTATŲ ANALIZEI
APPLICATION OF DATA MINING METHODS FOR ANALYSIS OF BUSINESS PROCESS
SIMULATION RESULTS**

patvirtinta 201...m. d. dekanų potvarkiu Nr.

Baigiamojo darbo užbaigimo terminas 201...m. d.

BAIGIAMOJO DARBO UŽDUOTIS:

1. Atlikti literatūros, susijusios su verslo procesų simuliacija, duomenų gavyba ir duomenų gavybos metodų panaudojimo verslo procesų simuliacijoje, analizę.
2. Atlikti egzistuojančių duomenų gavybos algoritmų, leidžiančių iš duomenų gauti taisykles, analizę.
3. Remiantis literatūra ir atliktos analizės rezultatais, pasiūlyti metodą, kaip patobulinti verslo procesų simuliaciją, atsižvelgiant į ankstesnių simuliacijos rezultatų pagrindu gautą informaciją.
4. Įgyvendinti pasiūlytą metodą ir atlikti eksperimentinį tyrimą, pateikti eksperimento rezultatus, suformuoti galutines išvadas ir pateikti tolimesnes mokslinių tyrimų kryptis.

Baigiamojo darbo rengimo konsultantai:

Lekt. Tadas Vysockis
(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

Dr. Aušra Žemienė
(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

Vadovas
(Parašas)

Prof. dr. Olegas Vasilecas
(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

Užduotį gavau

.....
(Parašas)
Justė Kareivaitė
(Vardas, pavardė)
.....
(Data)

Vilniaus Gedimino technikos universitetas
Fundamentinių mokslų fakultetas
Informacinių sistemų katedra

ISBN ISSN
Egz. sk.
Data~.....~.....

Antrosios pakopos studijų **Informacinių sistemų programų inžinerijos** programos magistro baigiamasis darbas

Pavadinimas **Duomenų gavybos metodų panaudojimas verslo procesų simuliacijos rezultatų analizei**

Autorius **Justė Kareivaitė**

Vadovas **Olegas Vasilecas**

Kalba: lietuvių

Anotacija

Baigiamajame magistro darbe nagrinėjamas duomenų gavybos metodų panaudojimas verslo procesų simuliacijoje. Darbu siekiama patobulinti verslo procesų simuliaciją gaunant žinias iš simuliacijos rezultatų, kai žinioms gauti naudojami duomenų gavybos metodai.

Darbe analizuojamas verslo procesų modeliavimas ir simuliacija, duomenų gavybos metodai bei duomenų gavybos algoritmai. Pasiūlyta kaip panaudoti duomenų gavybos metodą verslo procesų simuliacijoje, kuris leistų nustatyti taisykles verslo procesų simuliacijos rezultatuose bei rasti ryšį tarp simuliacijos pradinio duomenų ir rezultatų.

Taisyklėms nustatyti panaudotas CART sprendimų medžio algoritmas. Ryšio tarp pradinio duomenų ir rezultatų vizualiam vaizdavimui pasitelktas stačiakampių diagramų (angl. boxplot) atvaizdavimas, kuris parodo po duomenų gavybos sudarytų rezultatų grupių pasiskirstymą kiekvieno pradinio parametro atžvilgiu. Pasiūlytas metodas buvo realizuotas ir atlikti eksperimentiniai tyrimai su skirtingais duomenų rinkiniais.

Darbo apimtis – 84 p. teksto be priedų, 32 paveikslų, 7 lentelės, 46 bibliografiniai šaltiniai.

Prasminiai žodžiai: Verslo procesai, modeliavimas, simuliacija, duomenų gavybos metodai, duomenų gavybos algoritmai, žinių gavimas, taisyklių nustatymas, CART sprendimų medžio algoritmas, stačiakampių diagramų atvaizdavimas, duomenų gavybos metodų panaudojimas verslo procesų simuliacijoje.

Vilnius Gediminas Technical University
Faculty of Fundamental Sciences
Department of Information Systems

ISBN ISSN
Copies No.
Date-.....-.....

Master Degree Studies **Information Systems Software Engineering** study programme Master Graduation Thesis
Title **Application of Data Mining Methods for Analysis of Business Process Simulation Results**
Author **Justė Kareivaitė**
Academic supervisor **Olegas Vasilecas**

Thesis language: Lithuanian

Annotation

The final master's thesis examines the use of data mining methods for business process simulation. The aim of the work is to improve the simulation of business processes by finding knowledge from simulation results, when data mining methods are used to generate knowledge.

The paper analyzes business process modeling and simulation, data mining methods and data mining algorithms. A data mining method is proposed for use within the simulation of business processes. It allows identifying rules for simulation results and discovering a relationship between the initial simulation data and the results

The CART decision tree algorithm was used to determine the rules. Initial data and results are visually represented by boxplot diagrams, which illustrate the distribution of the result groups after data mining for each initial parameter. The proposed method was implemented and experiments were performed with different data sets.

Thesis consists of: 84 pages of text without appendices, 32 pictures, 7 tables and 46 references.

Keywords: Business Processes, Modeling, Simulation, Data Mining Methods, Data Mining Algorithms, Knowledge Mining , Rule Finding, CART Decision Tree Algorithm, Boxplot Rendering, Data Mining Methods for Business Processes Simulation.

(Baigiamojo darbo sąžiningumo deklaracijos forma)

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Justė Kareivaitė, 20132980

(Studento vardas ir pavardė, studento pažymėjimo Nr.)

Fundamentinių mokslų fakultetas

(Fakultetas)

Informacinių sistemų programų inžinerija, ISIfm-17

(Studijų programa, akademinė grupė)

BAIGIAMOJO DARBO (PROJEKTO)

SĄŽININGUMO DEKLARACIJA

2019 m. gegužės 28 d.

Patvirtinu, kad mano baigiamasis darbas tema „Duomenų gavybos metodų panaudojimas verslo procesų simuliacijos rezultatų analizei“ patvirtintas 2018 m. spalio 26 d. dekanų potvarkiu Nr. 413fm, yra savarankiškai parašytas. Šiame darbe pateikta medžiaga nėra plagijuota. Tiesiogiai ar netiesiogiai panaudotos kitų šaltinių citatos pažymėtos literatūros nuorodose.

Prenkant ir įvertinant medžiagą bei rengiant baigiamąjį darbą, mane konsultavo mokslininkai ir specialistai: Tadas Vysockis. Mano darbo vadovas profesorius daktaras Olegas Vasilecas.

Kitų asmenų indėlio į parengtą baigiamąjį darbą nėra. Jokių įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs (-usi).

(Parašas)

Justė Kareivaitė
(Vardas ir pavardė)

TURINYS

IVADAS	12
Tyrimo objektas.....	12
Darbo tikslas ir uždaviniai.....	12
Temos naujumas	12
Temos aktualumas	13
Tyrimo metodika	13
Darbo rezultatai	13
Darbo struktūra	13
1. SU VERSLO PROCESŲ SIMULIACIJA IR DUOMENŲ GAVYBA SUSIJUSIOS LITERATŪROS ANALIZĖ.....	15
1.1. Verslo procesai	15
1.1.1.Verslo taisyklės	15
1.1.2.Verslo proceso apibrėžimas	15
1.1.3.Statiniai verslo procesai	16
1.1.4.Dinaminiai verslo procesai.....	16
1.1.5.Tikslo siekiantys verslo procesai	17
1.1.6.Nuo konteksto priklausantys verslo procesai.....	18
1.1.7.Verslo procesų valdymas	19
1.2. Verslo procesų modeliavimas.....	19
1.2.1.Verslo procesų modeliavimo apibrėžimas	19
1.2.2.Verslo procesų modelių klasifikavimas	19
1.2.3.Verslo procesų modeliavimo kalbos	21
1.3. Verslo procesų analizė ir optimizavimas.....	21
1.3.1.Verslo procesų analizė	21
1.3.2.Verslo procesų optimizavimas	22
1.4. Verslo procesų simuliacija	23
1.4.1.Verslo procesų simuliacijos apibrėžimas.....	23

1.4.2. Verslo procesų simuliacijos privalumai ir trūkumai, lyginant su įprasta analize	24
1.4.3. Verslo proceso simuliacijos vykdymo etapai.....	24
1.4.4. Skirtumai tarp verslo proceso modelio ir simuliacijos modelio	25
1.5. Duomenų gavyba verslo procesų simuliacijos rezultatuose	26
1.5.1. Duomenų gavybos apibrėžimas	26
1.5.2. Duomenų gavybos technikų klasifikavimas	27
1.5.3. Grupavimas	27
1.5.4. Klasifikacija	28
1.5.5. Asociacijos	28
1.5.6. Regresija	29
1.5.7. Duomenų gavybos metodų palyginimas	29
1.6. Apibendrinimas ir rezultatai	31
2. DUOMENŲ GAVYBOS ALGORITMŲ TIPŲ ANALIZĖ	33
2.1. Sprendimų medžių algoritmai	33
2.2. Genetiniai algoritmai	37
2.3. Mišrūs sprendimų medžių / genetiniai algoritmai	39
2.4. K-vidurkių algoritmai	41
2.5. Neuroninių tinklų algoritmai	41
2.6. Duomenų gavybos algoritmų tipų palyginimas.....	42
2.7. Apibendrinimas ir rezultatai	43
3. SIŪLOMAS DUOMENŲ GAVYBOS METODAS VERSLO PROCESŲ SIMULIACIJOJE ...	45
3.1. Siūlomas metodas.....	45
3.2. Verslo procesų simuliacijos rezultatų paruošimas naudoti duomenų gavybos algoritmuose	46
3.3. Sprendimų medžio algoritmo panaudojimas simuliacijos rezultatų duomenų gavyboje	48
3.4. Ryšys tarp pradinių simuliacijos parametrų ir gautų simuliacijos rezultatų.....	49
3.5. Apibendrinimas ir rezultatai	50
4. SIŪLOMO DUOMENŲ GAVYBOS METODO REALIZACIJA	52
4.1. Duomenų gavybai atlikti pasirinktos priemonės	52

4.2. Duomenų rinkinio įkėlimas į <i>RStudio</i>	53
4.3. Metodui realizuoti reikalingi paketai.....	54
4.4. Metodo realizacijos aprašymas.....	55
4.5. Metodą aprašantis pseudokodas	57
4.6. Apibendrinimas ir rezultatai	60
5. Eksperimentiniai tyrimai	61
5.1. I eksperimentas	61
5.1.1.Naudojamas duomenų rinkinys.....	61
5.1.2.Eksperimento eiga ir rezultatai	63
5.2. II eksperimentas.....	66
5.2.1.Naudojamas duomenų rinkinys.....	66
5.2.2.Eksperimento eiga ir rezultatai	68
5.3. III eksperimentas	72
5.3.1.Naudojamas duomenų rinkinys.....	72
5.3.2.Eksperimento eiga ir rezultatai	74
5.4. Apibendrinimas ir rezultatai	77
DARBO REZULTATAI	78
IŠVADOS	79
LITERATŪROS ŠALTINIAI	81

PAVEIKSLŲ SĄRAŠAS

1 pav. Procesų analizės tipų tarpusavio ryšis ir priklausomybė verslo procesų modeliavimo grupėms (sudaryta autoriaus remiantis Vergidis et al., 2008).....	22
2 pav. Simuliacijos etapai (sudaryta autoriaus remiantis Van Der Aalst, 2015).....	25
3 pav. Sprendimų medžio formavimo, remiantis atributų nauda (sudaryta autoriaus remiantis Hssina et al., 2014).....	36
4 pav. Genetinio algoritmo populiacijos, chromosomos ir geno pavyzdys su dvejetainėmis vertėmis	37
5 pav. Mažos disjunkcijos pavyzdys sprendimų medyje (sudaryta autoriaus remiantis Carvalho ir Freitas, 2004).....	40
6 pav. Duomenų gavybos panaudojimo verslo procesų simuliacijoje metodas (sudaryta autoriaus remiantis Feldkamp et al., 2015).....	46
7 pav. Stačiakampių diagramų vaizdavimas, susiejant įvykdymo laiko parametą su rezultatų pasiskirstymu grupėse (sudaryta autoriaus remiantis Feldkamp et al., 2015).....	50
8 pav. <i>RStudio</i> programinės įrangos aplinka	52
9 pav. CSV formato failo įkėlimas <i>RStudio</i> aplinkoje	53
10 pav. Duomenų rinkinio įkėlimo lango vaizdas, <i>RStudio</i> programinėje įrangoje	54
11 pav. Pseudokodas: <i>RStudio</i> aplinkos paruošimas.....	57
12 pav. Pseudokodas: sprendimų medžio generavimas ir atvaizdavimas	57
13 pav. Pseudokodas: taisyklių gavimas	57
14 pav. Pseudokodas: duomenų įrašų papildymas grupę nusakančiu atributu I dalis.....	58
15 pav. Pseudokodas: duomenų įrašų papildymas grupę nusakančiu atributu II dalis	Error!
Bookmark not defined.	
16 pav. Pseudokodas: stačiakampių diagramų generavimas.....	Error! Bookmark not defined.
17 pav. Pseudokodas: labiausiai grupių pasiskirstymą įtakančių parametrų reikšmių ir galutinių simuliacijos rezultatų ryšio nustatymas.....	60
18 pav. I eksperimento verslo proceso modelis	63
19 pav. I eksperimento sprendimų medis	64
20 pav. I eksperimento taisyklių gavimo etapai	64
21 pav. I eksperimento stačiakampių diagramų atvaizdavimas	65
22 pav. II eksperimento verslo proceso modelis	67
23 pav. II eksperimento sprendimų medis	68
24 pav. II eksperimento taisyklių gavimo etapai.....	68
25 pav. II eksperimento stačiakampių diagramų atvaizdavimas.....	69

26 pav. Rezultatų priklausomybė nuo parametro <i>University Rating</i>	71
27 pav. Rezultatų priklausomybė nuo parametro <i>CGPA</i>	71
28 pav. III eksperimento verslo proceso simuliacijos modelis	73
29 pav. III eksperimento sprendimų medis	74
30 pav. III eksperimento taisyklių gavimo etapai	74
31 pav. III eksperimento stačiakampių diagramų atvaizdavimas	75
32 pav. Rezultatų priklausomybė nuo parametrų <i>Darbuotojų kiekis</i> ir <i>Vykdytys darbuotojai</i>	76

LENTELIŲ SĄRAŠAS

1 lentelė. Duomenų gavybos metodų palyginimas.....	30
2 lentelė. Sprendimų medžio panaudojimo pavyzdžio pradiniai duomenys (Hssina, Merbouha, Ezzikouri ir Erritali, 2014)	35
3 lentelė. Duomenų gavybos algoritmų tipų palyginimas.....	43
4 lentelė. I eksperimente naudoto duomenų rinkinio antraštė ir 5 pirmieji įrašai.....	62
5 lentelė. II eksperimente naudoto duomenų rinkinio antraštė ir 5 pirmieji įrašai.....	66
6 lentelė. III eksperimente naudoto duomenų rinkinio antraštė ir 5 pirmieji įrašai	72
7 lentelė. Nustatytų, labiausiai rezultatus įtakančiais, parametrų tyrimo rezultatai.....	76

IVADAS

Šiuolaikinis verslas siekia būti lankstus ir vykdyti veiklas kiek galima optimaliau. Tam, kad būtų galima patikrinti norimus įgyvendinti pokyčius ar rasti galutiniam rezultatui naudingą veiklą modifikaciją, neretai pasitelkiama verslo procesų simuliacija. Verslo procesų modeliavimas yra gana plačiai ištirta sritis, nes norint valdyti procesą svarbu mokėti jį pavaizduoti su verslu susijusiems asmenims suprantama išraiška, tačiau simuliacija vis dar turi daug neišanalizuotų aspektų ir yra nuolat tobulinama. Vienas iš trūkumų yra tas, kad simuliacijos metodai neleidžia susieti pradinių parametru ir gautų rezultatų, kitaip sakant, nėra galimybės iš simuliacijos rezultatų gauti taisykles apie simuliuojama verslo procesą. Taigi šiame darbe nagrinėjama problema – egzistuojantys simuliacijos metodai neleidžia nustatyti sąsajų tarp įvesties ir išvesties duomenų.

Tyrimo objektas

Tyrimo objektas – duomenų gavybos metodai, skirti verslo procesų simuliacijos rezultatų analizei.

Darbo tikslas ir uždaviniai

Darbo tikslas – praplėsti verslo procesų simuliacijos rezultatų analizės galimybes, kai žinioms gauti panaudojami duomenų gavybos metodai.

Darbo tikslui pasiekti nustatomi šie uždaviniai:

1. Atlikti literatūros, susijusios su verslo procesų simuliacija, duomenų gavyba ir duomenų gavybos metodų panaudojimo verslo procesų simuliacijoje, analizę.
2. Atlikti egzistuojančių duomenų gavybos algoritmų, leidžiančių iš duomenų gauti taisykles, analizę.
3. Remiantis literatūra ir atliktos analizės rezultatais, pasiūlyti metodą, kaip patobulinti verslo procesų simuliaciją, atsižvelgiant į ankstesnių simuliacijos rezultatų pagrindu gautą informaciją.
4. Įgyvendinti pasiūlytą metodą ir atlikti eksperimentinį tyrimą, pateikti eksperimento rezultatus, suformuoti galutines išvadas ir pateikti tolimesnes mokslinių tyrimų kryptis.

Temos naujumas

Šiuolaikiniai simuliacijos įrankiai dažniausiai pateikia rezultatus kaip statistikų rinkinį, tačiau nesuteikia platesnių analizės galimybių, kurios leistų įvertinti pradinių duomenų įtaką galutiniams simuliacijos rezultatams. Šiame darbe aptariamas siūlomas verslo procesų simuliacijos rezultatų

analizės galimybių praplėtimas, žinias apie simuliacijos pradinių duomenų ir rezultatų sąsajas išgaunant pasitelkus duomenų gavybos metodus, bus visiškai naujas.

Temos aktualumas

Verslo procesų modeliavimo ir simuliacijos tema yra aktuali ir pastaruoju metu plačiai nagrinėjama. Stengiamasi patobulinti modeliavimui ir simuliacijai naudojamus įrankius, juos praplečiant didesnėmis galimybėmis, kad rezultatai būtų kuo panašesni į realiam gyvenime gaunamus rezultatus. Šiuolaikinis verslas nuolat nori tobulėti ir gerinti verslui valdyti reikalingas veiklas, šiam tikslui dažnai pasitelkiama verslo procesų simuliacija. Tačiau simuliacija leidžia tik patikrinti iškeltas hipotezes, kaip būtų galima optimizuoti procesą, bet nepateikia analizės, kuri leistų įvertinti simuliacijoje dalyvaujančių parametrų įtaką galutiniams rezultatams. Šia tema skelbiami moksliniai darbai ir tyrimai, kuriuose įvardinami simuliacijos rezultatų analizės, vizualizuojant simuliacijos rezultatų priklausomybę nuo įvesties parametrų, privalumai – išgauti žinias, kurios simuliacijos rezultatuose nėra pastebimos (Feldkamp, Bergmann and Strassburger, 2015).

Tyrimo metodika

Analitinėje darbo dalyje, atliekant mokslinės literatūros, susijusios su verslo procesų modeliavimu ir simuliacija, analizę, naudojami bibliotekinio tyrimo ir lyginamosios analizės metodai. Analizuojami egzistuojantys metodai ir algoritmų tipai, kurie leidžia iš simuliacijos rezultatų gauti žinias, išskiriant jų galimybes, privalumus ir trūkumus. Remiantis tyrimo rezultatais pasiūlytas duomenų gavybos metodas verslo procesų simuliacijos rezultatams analizuoti. Pasiūlytas metodas realizuotas praktiškai bei su juo atlikti eksperimentiniai tyrimai.

Darbo rezultatai

1. Atlikta literatūros analizė padėjo nustatyti tolimesnę tyrimų kryptį – žinių gavimas iš simuliacijos rezultatų.
2. Atlikus taisyklių iš verslo procesų simuliacijos rezultatų gavimo algoritmų tipų analizę ir nustatius jų privalumus ir trūkumus pasirinktas tolimesniam darbui tinkamiausias algoritmų tipas – sprendimų medžių duomenų gavybos algoritmai. Taip pat nustatyta, kad duomenų apdorojimui ir žinių gavimui sprendimų medžio algoritmu bus pasitelktas grupavimo metodas.
3. Pasiūlytas duomenų gavybos iš verslo procesų simuliacijos rezultatų metodas.
4. Realizuotas siūlomas duomenų gavybos metodas bei atlikti bandymai su skirtingais duomenų rinkiniais.

Darbo struktūra

Darbą sudaro įvadas, penki skyriai ir išvados.

Įvade aprašoma darbe nagrinėjama problema, tyrimo objektas, darbo tikslas ir uždaviniai, aptariamas temos naujumas, aktualumas, tyrimo metodika, darbo mokslinė vertė ir rezultatai.

Pirmame skyriuje yra nagrinėjama susijusių mokslinių darbų literatūra ir analizuojamos su dalykine sritimi susijusios sąvokos.

Antrame skyriuje analizuojami ir lyginami egzistuojantys duomenų gavybos algoritmų tipai, kurie iš verslo procesų simuliacijos rezultatų gali gauti taisykles. Taip pat, tolimesniam darbui, pasirenkamas vienas iš algoritmų tipų – sprendimų medžių.

Trečiame skyriuje pateikiamas metodas, skirtas duomenų gavybai iš verslo procesų simuliacijos rezultatų.

Ketvirtame skyriuje aprašoma metodo realizacija, pasirinktoje duomenų gavybos programinėje įrangoje.

Penktame skyriuje pateikiama eksperimentinių tyrimų eiga ir rezultatai.

Išvadose ir pasiūlymuose pateikiamos darbo išvados bei siūlomos tolimesnių tyrimų kryptys, kurias būtų galima vystyti remiantis šiuo darbu.

1. SU VERSLO PROCESŲ SIMULIACIJA IR DUOMENŲ GAVYBA SUSIJUSIOS LITERATŪROS ANALIZĖ

1.1. Verslo procesai

1.1.1. Verslo taisyklės

Verslo taisyklės galima apibrėžti kaip taisyklės, pagal kurias vadovaujama verslui vykdamas kasdienės operacijas (Ross, 2000). Jei nebūtų naudojamos verslo taisyklės, nebūtų formalizuota organizacijos veikla, apibrėžiama kaip turi elgtis verslas tam tikroje situacijoje. Be to, vadovaujantis verslo taisyklėmis valdoma strategija, terminai, kainos, apibrėžimai, kontroliuojami ir reguliuojami verslo procesai, organizacinės struktūros ir verslo elgsena (Raibova, 2009). Taigi galima teigti, kad verslo taisyklės yra verslo pagrindas.

Analizuotame Ivana'os Raibova'os straipsnyje išskiriamos trys verslo taisyklių rūšys: valdymo, veiklos ir automatizavimo (2009). Šių rūšių charakteristikos:

- Valdymo taisyklė – teisinis teiginys, oficialus reglamentas, privalomas susitarimas, verslo strategija. Šia taisykle siekiama apriboti ar reguliuoti verslą.
- Veiklos taisyklė – konstatuojamasis sakiny. Ši taisyklė privalo būti nedviprasmiška ir tiesiogiai susijusi su įmonės vidine veikla.
- Automatizuota taisyklė – konkreti verslo taisyklė, nurodanti elgseną tam tikroje situacijoje.

1.1.2. Verslo proceso apibrėžimas

Dar 1993 metais Michael'as Hammer'is ir James'as Champy apibrėžė verslo procesą, kaip veiklų seką, kuri kartu sukuria klientui naudingą rezultatą (1993). Laikui bėgant buvo pasiūlyta daug šio apibrėžimo variantų, tačiau visi jie esminiais įvardija du punktus: veiklų seką ir siekiamą rezultatą. Atsižvelgiant į tai, kas vyksta proceso metu, kol pasiekiamas galutinis rezultatas, galima išskirti kelis papildomus apibrėžimus (Mili et al., 2010):

- Veiklas vykdo aktoriai, kurie atlieka tam tikrą vaidmenį bei vykdydami numatytąsias veiklas veikia organizacinių ribų kontekste.
- Atliekant veiklą yra sunaudojami vieni ištekliai ir gali būti sukuriami kiti.
- Veikla gali būti suaktyvinta įvykių metu bei sukurti kitus įvykius.
- Proceso veiksmas gali būti priklausomi nuo išteklių ar kontrolės.

Pagal savo struktūrą verslo procesai skirstomi į du pagrindinius tipus: statinius ir dinامينius. Vis dėlto dažnai dinaminiai verslo procesai būna dar papildomai skaidomi į kelis tipus pagal savo specializaciją ar modifikavimo galimybes, vieni dažniausiai mokslinėje literatūroje aprašomų

dinaminių tipų atmainų yra tikslo siekiantys verslo procesai ir nuo konteksto priklausantys verslo procesai.

1.1.3. Statiniai verslo procesai

Statiniai verslo procesai dar gali būti vadinami struktūruotais, tai tokie procesai, kurių struktūra nesikeičia ir yra žinoma iš anksto (Szelagowski, 2014). Šie procesai gali būti nesunkiai perduoti automatiniam vykdymui robotams ar kompiuteriams, nes jie visada yra tokie pat ir nereikalauja žmogaus įsikišimo. Statiniai procesai gali būti naudojami procesų tobulinimui, identifikavimui, komunikacijai tarp suinteresuotųjų asmenų, darbo eigos automatizavimui ar duomenų rinkimo ir analizavimo automatizavimui.

Nekintančių verslo procesų privalumai: lengvas perdavimas automatiniam vykdymui, supratimo apie organizacijoje vykstančius procesus suvienodinimas tarp darbuotojų, procesų skaidrumas, proceso tobulinimas, realaus laiko proceso vykdymo kontrolė, pavyzdžiui klaidų aptikimas procesui vykstant. Žinoma šio tipo verslo procesai turi ir trūkumų: neatitikimas dinaminiam verslui, galimybės kurti individualizuotus procesus trūkumas. Dėl išvardintų trūkumų dabartiniame versle labai retai statiniai verslo procesai padeda efektyviai valdyti verslą, nebent labai griežtai apibrėžtose srityse, kuriose proceso metu jokie pokyčiai niekada negali įvykti. Tačiau statiškas modeliavimas yra labai naudingas siekiant suprasti ir pristatyti procesą versle dalyvaujantiems susijusiems asmenims, net jeigu jie neturi verslo procesų analitinių žinių (Patel ir Hlupic, 2001).

1.1.4. Dinaminiai verslo procesai

Dinaminiais yra vadinami tokie verslo procesai, kurie neturi griežtai nustatytos veiklų sekos ir gali greitai prisitaikyti prie besikeičiančių verslo poreikių (Vasilecas et al., 2016; Rouse, 2012). Dinaminiuose verslo procesuose veiklos gali būti parenkamos atsižvelgiant į taisykles ir aplinkos pokyčius, kitaip dar vadinamus konteksto pokyčiais. Svarbu, kad pokyčius būtų galima atlikti bet kuriuo metu, greitai ir efektyviai.

Kintantys procesai, kaip ir statiniai, gali būti naudojami tose pat srityse, tačiau jie gali būti pritaikomi veikti kintančioje aplinkoje (Szelagowski, 2014). Dinaminių verslo procesų privalumai taip pat kartojasi su statinių, išskyrus tai, kad dinaminį verslo procesą gana sudėtinga perduoti automatiniam vykdymui, tačiau jį lengva pritaikyti, modifikuoti pagal pasikeitusius poreikius. Šio tipo procesų trūkumai: eksperimento žlugimas dėl nepakankamo vykdymų skaičiaus, nes rezultatų atitikimas realybei priklauso nuo bandymų kiekio, ir rezultatų chaotiškumas dėl nekontroliuojamų eksperimentų.

Dinaminiai verslo procesai gali būti skirstomi pagal dinamiškumo lygius. Margaret Rouse siūlo tris verslo procesų valdymo metodų dinamiškumo lygius (2012):

- Žemo lygio: leidžia spręsti paprastus verslo sprendimus, kai yra nustatytos galimos veiklos tam tikru atveju. Pavyzdžiui šio lygio metoduose gali būti nustatomi slenksčiai, kada numatytus veiksmus automatiškai vykdo sistema ir kada turi įsikišti žmogus.
- Aukštesniojo lygio: leidžia naudoti dinaminę konfigūraciją. Šiame lygyje galima keisti veiklų atlikimo tvarką arba negalint vykdyti pirminės sekos pasirinkti numatytą alternatyvų kelią.
- Aukščiausio lygio: leidžia vartotojams kurti į tikslą orientuotus procesus, kurie gali kisti laike, kitaip sakant, jų vykdymo metu, priklausomai nuo besikeičiančio konteksto ar klientų poreikių, svarbu, kad būtų pasiektas numatytas tikslas.

Vienoje iš analizuotų konferencijos medžiagų aprašoma ir lankstumo klasifikaciją, autoriai procesų lankstumą skirsto pagal šiuos aspektus (Bhat ir Deshmukh, 2005):

- Lankstumas proceso sekoje: paprastumas proceso veiklų sekoje prisitaikyti prie pokyčių.
- Lankstumas taikomoms verslo taisyklėms: verslo taisyklių keitimas ir strategijų nustatymas bei pritaikymas.
- Praktikos lankstumas: gebėjimas įtraukti naują praktiką.
- Lankstumas tvarkantis su proceso išimtimis: proceso reagavimas į išimtinis atvejus.

1.1.5. Tikslų siekiantys verslo procesai

Tikslas įmonėje gali būti apibūdinamas kaip verslo pagrindinis siekis, o sistemoje – jos siekiamas įeigos apdorojimo rezultatas. Peter'is Kueng'as ir Peter'is Kawalek'as tikslus apibrėžia kaip teiginius, kuriuos verslo procesai turi pasiekti arba kaip tik jų išvengti (1997).

Tradicionis verslo proceso modelis remiasi nuoseklia veiksmų seka, o į tikslą orientuotas verslo procesas žiūri šiek tiek plačiau, jo požiūriu svarbiau pasiekti verslo tikslą nei galvoti apie individualius veiksmus (Greenwood, 2016). Kitaip sakant į tikslą orientuotame verslo procese veiksmas, kurie bus naudojami tikslui pasiekti, gali būti kintantys, svarbu, kad prie visų pokyčių būtų prisitaikyta ir galiausiai gaunamas reikiamas rezultatas.

Tikslų siekiantys verslo procesai, kurie orientuojasi į galutinį rezultatą, o ne jo įgyvendinimą, puikiai pritaikomi dinamiškai besikeičiančiose verslo srityse, kuriose sprendimai atliekami tiriant nuolat atnaujinamą informaciją apie kontekstą (Lapouchnian ir Yu, 2016). Savybė realiu laiku reaguoti į kintančią aplinką suteikia galimybę tokius procesus panaudoti analizei ir juos optimizuoti, tačiau kol kas modeliavimo ir simuliacijos įrankiai nėra tam pakankamai pritaikyti.

1.1.6. Nuo konteksto priklausantys verslo procesai

Šiandieniniame versle klientų lūkesčiai nuolat auga, o jų poreikiai sparčiai keičiasi, todėl ir verslas privalo būti lankstus. Kad patenkintų nuolat besikeičiančio verslo poreikius verslo procesai privalo atsižvelgti į verslo kontekstą. Su kontekstu susijusios žinios gali padėti verslo proceso elgseną pritaikyti prie situacijos (Saidani ir Nurcan, 2007). Tradiciškai yra naudojamas vienas verslo proceso modelis, kitaip sakant statinis, kuris tenkina klientų poreikius konkrečiame kontekste, kuriam yra pritaikytas, tačiau susidūręs su kitokiu kontekstu tampa neįgyvendinamas.

Galvojant apie tai, kad verslo procesas turi gebėti veikti esant skirtingoms aplinkybėms, svarbu suprasti kokios jos gali būti ir kaip pakeisti proceso eigą, kad kuo efektyviau prisitaikyti prie esamo konteksto. Taip pat būtina įvertinti konteksto reikšmingumą, įvertinti kas yra aktualiausia tam tikroje situacijoje. Pavyzdžiui reikiant procesą įgyvendinti labai skubiai, svarbu atsižvelgti į darbuotojų kompetenciją, tada netinka tipiškas darbų paskyrimas, kai darbas siunčiamas tiems darbuotojams, kurie nėra užimti, svarbu išrinkti daug darbo patirties turintį darbuotoją. Konteksto supratimas leidžia pritaikyti verslo procesus prie besikeičiančių konteksto aplinkybių, kad jie savarankiškai galėtų atsižvelgti į pokyčius. Be to, labai svarbu įgyvendinti verslo procesų lankstumą, kad kontekstą būtų galima keisti ar papildyti bet kuriuo metu.

Jan'as Vom Brocke'as, Sarah Zelt ir Theresa Schmiedel verslo procesų valdymo kontekstą apibūdina perspektyvomis ir skaido į keturis aspektus (2016): verslo proceso valdymo tikslus, verslo proceso ypatumus, organizacijos ypatybes ir verslo aplinkos ypatybes. Iš verslo proceso valdymo tikslų galima išskirti šiuos pavyzdžius: noras padidinti efektyvumą, pagerinti veiklos kokybę, patobulinti procesą ar pritaikyti inovacijas. Organizacinėmis ypatybėmis galima įvardinti tokias savybes kaip proceso įgyvendinimo aplinka – organizacinė ar vykstanti ir už jos ribų, organizacijos dydis, pramonės rūšis, kultūrinės vertybės. Verslo aplinkos ypatybės yra tokios, kurios vyksta organizacijos išorėje, kaip rinkos, socialiniai, kultūriniai ar politiniai veiksniai.

Oumaima Saidani ir Selmin Nurcan išskiria keturias svarbiausias konteksto rūšis (2007):

- Su vieta susijęs kontekstas: nurodo vietos ypatybes. Pavyzdžiui užduoties skyrimas darbuotojui priklauso nuo fizinės vietos, kurioje jis dirba, nes tik tam tikroje vietoje esantis žmogus gali užduotį įgyvendinti.
- Su laiku susijęs kontekstas: jis gali apimti laiką, kiek ilgai užduotis turi būti atliekama, užduoties skubumą arba kitaip sakant svarbumą, darbo trukmę, dažnumą ir laiko taupymą.
- Su ištekliais susijęs kontekstas: apima fizinius ir žmogiškuosius resursus. Žmogiškieji resursai gali būti vertinami pagal tokias savybes kaip amžius ir lytis ar tiesiogiai susijusiomis su darbu, pavyzdžiui motyvacija, įvykdymas, bendradarbiavimas, mobilumas, taip pat savybės dar gali

atspindėti santykius tarp subjektų. Fizinių išteklių savybės gali būti susijusios su verslo objektų ypatumais, kaip išteklių stoka, prieinamumas, taip pat finansinėmis savybėmis arba laiko ištekliais.

- Su organizacija susijęs kontekstas: apima darbo vietos charakteristikas, organizacinės struktūros tipą, kultūrinius ir socialinius aspektus.

1.1.7. Verslo procesų valdymas

Verslo procesų valdymas jungia verslą ir informacines technologijas, siekiant tobulinti organizacijos verslo veiklą (Vom Brocke, Mathiassen ir Rosemann, 2014). Pasak Monika'os Malinova'os ir Jan'o Mendling'o, organizacijos verslo procesų valdymą naudoja kaip metodą didinti turimą supratimą ir žinias apie verslo procesus (2013). Taip pat dar išskiriama, kad verslo procesų valdymas gali būti naudojamas siekiant optimizuoti procesą (Rima ir Vasilecas, 2015). Didelė dalis įmonių vis daugiau taiko verslo procesų valdymą, jis pasitelkiamas skirtingiems tikslams, tokiems kaip procesų automatizavimas, projektavimas ar greitaveikos padidinimas. Vis dėlto nedaug kompanijų geba suprasti ir efektyviai naudoti verslo procesų valdymą bei gauti iš jo realios naudos.

1.2. Verslo procesų modeliavimas

1.2.1. Verslo procesų modeliavimo apibrėžimas

Verslo procesų modeliavimas yra pradinė verslo procesų valdymo dalis ir organizacijų, kurios vykdo procesais pagrįstą veiklą, pagrindinis elementas (Gabryelczyk ir Jurczuk, 2017). Analizuotų mokslinių darbų autoriai teigia, kad verslo procesų modeliai kuriami tam, kad būtų galima grafiškai atvaizduoti realų procesą, kuris iš tiesų yra vykdomas versle, tačiau norint jį geriau suprasti ar pasidalinti su susijusiais asmenimis, naudinga procesą pateikti naudojant grafinius elementus.

Anot David'o J. Teece'o, norint gerai sumodeliuoti verslo procesą būtina įsigilinti į klientų poreikius, technologinius bei organizacinius išteklius, atsižvelgti į kitų naudojamus efektyvius verslo modelius bei remtis strategine verslo analize (2017). Taigi modeliuojant verslo procesą pagal realiai vykdomas veiklas galima patikrinti ar jis iš tiesų vykdomas pakankamai efektyviai, o gal net būtų naudinga jį patobulinti.

1.2.2. Verslo procesų modelių klasifikavimas

Mokslinėje literatūroje galima rasti įvairių verslo procesų modelių klasifikacijos būdų, pradedant skaidymu pagal skirtingus požiūrius į verslo procesų pobūdį, naudojimo tikslus ir modelių aktyvumą bei pagal struktūrines ypatybes, galimybes analizuoti ir optimizuoti.

Nuno Melao ir Michael'as Pidd'as išskyrė keturias perspektyvas, kurioms priskiriami modeliavimo metodai pagal požiūrį į verslo procesų pobūdį (2000). Pirma perspektyva – verslo

procesai, kaip deterministinės mašinos, kuri teigia, kad naudojama fiksuota, aiškiai apibrėžtų veiksmų seka, kuri įvestis paverčia išvestimis, pasiekiant aiškiai apibrėžtus tikslus. Šiai perspektyvai pakanka statinio proceso modeliavimo, tokio kaip: funkcinio modeliavimo metodas (IDEF0), procesų aprašymo suvokimo metodas (IDEF3), srautų diagrama, vaidmenų veiklos diagrama (RAD), klientų-tiekėjų protokolai. Antroje perspektyvoje verslo procesai yra suprantami kaip sudėtingos, dinaminės ir interaktyvios sistemos. Šis požiūris pabrėžia sąveiką ir dinaminę elgesį bei gali būti apibrėžiamas kaip posistemių rinkinys: žmonės, užduotys, struktūra, technologijos ir kita, kai posistemiai sąveikauja tarpusavyje ir su aplinka, siekiant įgyvendinti konkrečius tikslus. Trečioji perspektyva – sąveikaujanti grįžtamojo ryšio linija, kurioje dinamišką procesą bandoma suvokti ne atskirų komponentų požiūriu, o vidinės struktūros ir strategijos sąveika. Paskutinė perspektyva yra socialiniai konstruktai, kuri verslo procesus mato žmonių, kurie juos kuria ir įgyvendina, požiūriu. Ši perspektyva vadinama minkšta ir gali būti modeliuojama nestruktūriniais iliustraciniais modeliais.

Ruth Sara Aguilar-Saven verslo procesų modelius klasifikuoja pagal du aspektus: tikslus ir aktyvumą (2004). Išskiriamos keturios modelių tikslų kategorijos:

1. Aprašomieji mokymosi.
2. Sprendimų palaikymo, skirti procesui kurti ir projektuoti.
3. Sprendimų palaikymo, proceso vykdymo ir valdymo metu.
4. Informacinių technologijų palaikymo patvirtinimo.

Antruoju aspektu modeliai gali būti aktyvūs arba pasyvūs. Aktyviais vadinami tokie modeliai su kuriais vartotojas gali bendrauti, pasyvūs – su kuriais bendrauti nėra galimybės.

Kostas Vergidis, Ashutosh'as Tiwari ir Basim'as Majeed'as pasiūlė klasifikavimą pagal struktūrines verslo procesų modelių ypatybes, galimybes juos analizuoti ir optimizuoti (2008). Autoriai siūlo tris klasifikavimo grupes:

1. Grafiniai modeliai – juose verslo procesų modeliai atvaizduojami vizualinėmis diagramomis. Šios standartizuotos diagramos yra lengvai suprantamos ir paprastai naudojamos, tačiau modeliuose trūksta formalios semantikos ir kiekybinės informacijos. Grafinių modelių grupei priklausančios technikos: srautų diagramos, IDEF (angl. *Integration Definition Function Modeling*) standartų šeima, RAD (angl. *Role Activity Diagrams*), vieninga modeliavimo kalba UML (angl. *Unified Modeling Language*), Petri tinklai.
2. Matematiniai arba formalūs modeliai – juose visi elementai turi matematinį ar formalų pagrindą, o proceso koncepcijos yra griežtai ir tiksliai apibrėžtos. Formalių modelių pranašumas, kad juos galima patikrinti matematiškai, tačiau dauguma verslo procesų yra

kokybinio pobūdžio, tad jų negalima modeliuoti matematiniu pagrindu. Šių modelių grupei priklauso matematiniai ar algoritmų modeliai ir Petri tinklai.

3. Verslo procesų modeliavimo kalbos – tai programinės įrangos kalbos, kurios palaiko verslo procesų modeliavimą. Verslo procesų kalbos stengiasi išlaikyti formalių modelių nuoseklumą ir plačias analizės galimybes, tačiau sumažinti modelių sudėtingumą. Verslo procesų kalboms priskiriamos technikos: UML, verslo procesų vykdymo kalba BPEL (angl. *Business Process Execution Language*), modeliavimo kalba BPML (angl. *Business Process Modeling Language*) ir verslo procesų modeliavimo notacija – BPMN (angl. *Business Process Model and Notation*).

1.2.3. Verslo procesų modeliavimo kalbos

Verslo procesų modeliais remiasi ne vienos srities organizacijos darbuotojai, todėl svarbu juos atvaizduoti skirtingas kompetencijas turintiems darbuotojams suprantama grafinių modelių forma. Dėl poreikio naudoti plačiai suprantamas modeliavimo formas buvo pasitelktos verslo procesų modeliavimo kalbos, kitaip dar vadinamos žymėjimais. Labiausiai paplitusios yra dvi: vieninga modeliavimo kalba, sutrumpintai vadinama UML ir verslo procesų modeliavimo notacija – BPMN, kuri net yra tapusi modeliavimo kalbų *de facto* standartu.

UML yra plačiai naudojama modeliavimo kalba, tad didžiausias jos privalumas, kad ši kalba yra gerai pažįstama daugelio sričių specialistams, be to, ji lengvai skaitoma, net ir neturintiems teorinių žinių apie UML. Verslo procesų modeliavimui gali būti naudojamos UML veiklos diagramos (Razavian ir Khosravi, 2008). Vieninga modeliavimo kalba turi labai mažai grafinių elementų, todėl ją naudojant nėra lengva sukurti sudėtingo proceso modelį.

BPMN atsirado kaip procesus aprašanti kalba, kurią naudojo procesų inžinieriai veiklą sekai (angl. *workflow*) aprašyti arba verslo procesų valdymo sistemose, vėliau informacinių technologijų specialistai pastebėjo, kad ši notacija puikiai leidžia kurti ir techninius modelius, taigi dabar BPMN yra naudojamas tiek verslo procesų analizei, tiek modeliavimui verslo procesų valdymo sistemose (Allweyer, 2016). Ši kalba yra gerokai sudėtingesnė už UML, todėl skaitant modelius būtina turėti ir teorinių žinių. Tačiau dėl savo sudėtingumo ir plataus elementų sąrašo BPMN leidžia verslo proceso modelyje parodyti daug jo vykdymui reikšmingų detalių.

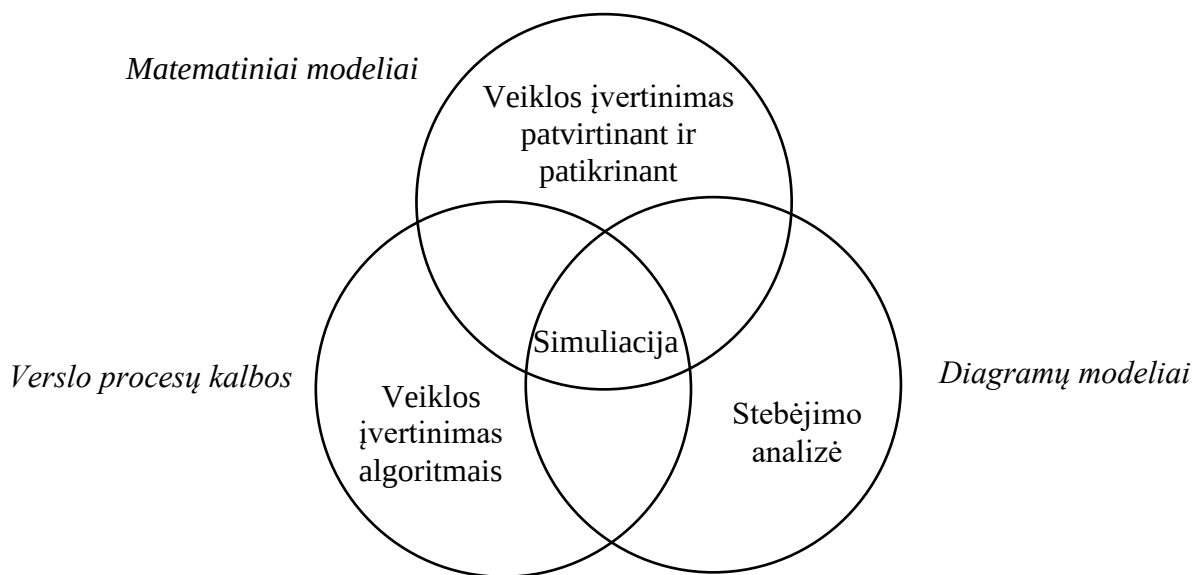
1.3. Verslo procesų analizė ir optimizavimas

1.3.1. Verslo procesų analizė

Verslo procesai analizuojami siekiant nustatyti proceso charakteristikas, suprasti kaip jį būtų galima pagerinti. Galima išskirti tris skirtingus analizės tipus: stebėjimo analizė, veiklos įvertinimas

patvirtinant ir patikrinant bei veiklos įvertinimas algoritmais. Kiekvienas iš šių analizės tipų atitinka po vieną iš K. Vergidžio, A. Tiwari ir B. Majeed'o apibrėžtų verslo procesų modelių klasifikavimo grupių, metodų (2008).

Stebėjimo analizė dažniausiai naudojama grafiniams modeliams analizuoti. Šios analizės metu diagramos yra tikrinamos keičiant proceso struktūrą. Stebėjimas gali užtrukti labai daug laiko ir turi didelę priklausomybę nuo jį modeliuojančio žmogaus patirties. Veiklos įvertinimas patvirtinant ir patikrinant priskiriamas matematinių modelių analizės metodui. Patvirtinimas gaunamas bandymų būdu tikrinant ar tam tikrame kontekste verslo procesas elgsis taip kaip tikimasi, o patikrinimas vykdomas nustatant verslo proceso teisingumą. Verslo procesų modeliavimo kalbų modeliuose naudojama veiklos įvertinimo algoritmais analizė. Šios analizės metu atliekamas gebėjimo laikytis reikalavimų įvertinimas, atsižvelgiant į kiekybinius veiksnius. 1 paveiksle pavaizduota visų analizės tipų priklausomybė verslo procesų modelių klasifikuotoms grupėms ir tarpusavio ryšis.



1 pav. Procesų analizės tipų tarpusavio ryšis ir priklausomybė verslo procesų modeliavimo grupėms (sudaryta autoriaus remiantis Vergidis et al., 2008)

Iš 1 paveikslo matyti, kad simuliacija apima visus analizės tipus, tad galima teigti, kad dėl to simuliacija ir pasitelkiama kaip didžiausias analizės galimybes turintis įrankis.

1.3.2. Verslo procesų optimizavimas

Optimizuojant yra tobulinama tam tikra pasirinkta sritis ar jos savybės, siekiant rasti geriausią sprendimą, nekeičiant esamo funkcionalumo. Verslo procesų optimizavimą K. Vergidis, A. Tiwari ir B. Majeed'as apibrėžia kaip automatizuotą tobulinimą, naudojant iš anksto apibrėžtus kokybinius veiklos rodiklius (2008). Verslo procesų optimizavimas gali būti naudojamas įvairiems tikslams,

pavyzdžiui siekiant sumažinti proceso vykdymo išlaidas, įgyvendinimo laiką, pagerinti teikiamų paslaugų kokybę, padidinti klientų pasitenkinimą.

Išskiriami trys verslo procesų optimizavimo metodai: grafiko mažinimo technikos, algoritmų ir verslo proceso veiklų ar užduočių konsolidavimo (sustiprinimo) (Vergidis et al., 2008). Grafiko mažinimo technikos keičia proceso diagramos modelį, tam, kad būtų nustatyti struktūriniai konfliktai arba pašalintos aklavietės. Algoritmų metodu siekiama sumažinti sprendimų įgyvendinimo išlaidas, tai siekiama padaryti sukuriant kiekybinį modelį. Konsolidavimo tikslas yra sutrumpinti proceso ciklo trukmę, tai padaryti galima sustiprinant veiklas, perduodant daugiau veiklų vienam veikėjui.

1.4. Verslo procesų simuliacija

1.4.1. Verslo procesų simuliacijos apibrėžimas

Verslo procesų simuliacija, tai verslo proceso modelio imitavimas kompiuteryje, siekiant atspindėti proceso vykdymą realybėje (Van Der Aalst, 2015). Simuliacijos modelis kuriamas turint konkretų tikslą, tikslas gali būti rasti atsakymą į tam tikrą, su verslu susijusį, klausimą, kaip, pavyzdžiui – kokia yra vidutinė klientų laukimo trukmė, kol gaunamas patvirtinimas dėl sėkmingai atlikto prekės užsakymo? Marc'o I. Kellner'io, Raymond'o J. Madachy ir David'o M. Raffo straipsnyje išskiriamos šešios simuliacijos tikslų grupės: strateginio valdymo, planavimo, kontrolės ir operacinio valdymo, proceso tobulinimo ir technologijos pritaikymo, supratimo bei mokymo (1999).

Wil'as M. P. Van Der Aalst'as išskiria tris situacijas, kuriose yra tikslinga naudoti simuliaciją (2015):

1. Norint suprasti esamą arba būsimą situaciją – simuliuojant galima surasti kritines arba neefektyvias verslo proceso vietas.
2. Norint imituoti eksperimentą, kurį atlikti realiai kainuotų pernelyg brangiai – simuliuojant galima patikrinti ar numatomi pokyčiai verslo procese bus tokie efektyvūs kaip tikimasi ir ar padės išspręsti norimas pašalinti problemas.
3. Norint imituoti eksperimentą, kurį atlikti realiai būtų pernelyg pavojinga – modeliavimas padeda įvertinti planuojamus pokyčius, tačiau realiam saugumui užtikrinti reikia imtis ir kitų priemonių, tokių kaip detali analizė.

Simuliuojant procesą gali būti nurodyta į ką reikia atsižvelgti simuliacijos metu, pavyzdžiui išteklius ar konteksto pokyčius. Ištekliai gali būti riboti ar prieinami ne visą simuliacijos laiką, jais gali reikėti dalintis su kitais procesais. Jei išteklius yra žmogiškasis, reikia atsižvelgti į jo darbo laiką, pertraukas. Kontekste gali pasikeisti kliento norai, tam tikrų veiklų prioritetai.

1.4.2. Verslo procesų simuliacijos privalumai ir trūkumai, lyginant su įprasta analize

Viename iš analizuotų W. M. P. van der Aalst'o mokslinių darbų, autorius išskiria tris verslo procesų simuliacijos privalumus prieš įprastą analizę (2015):

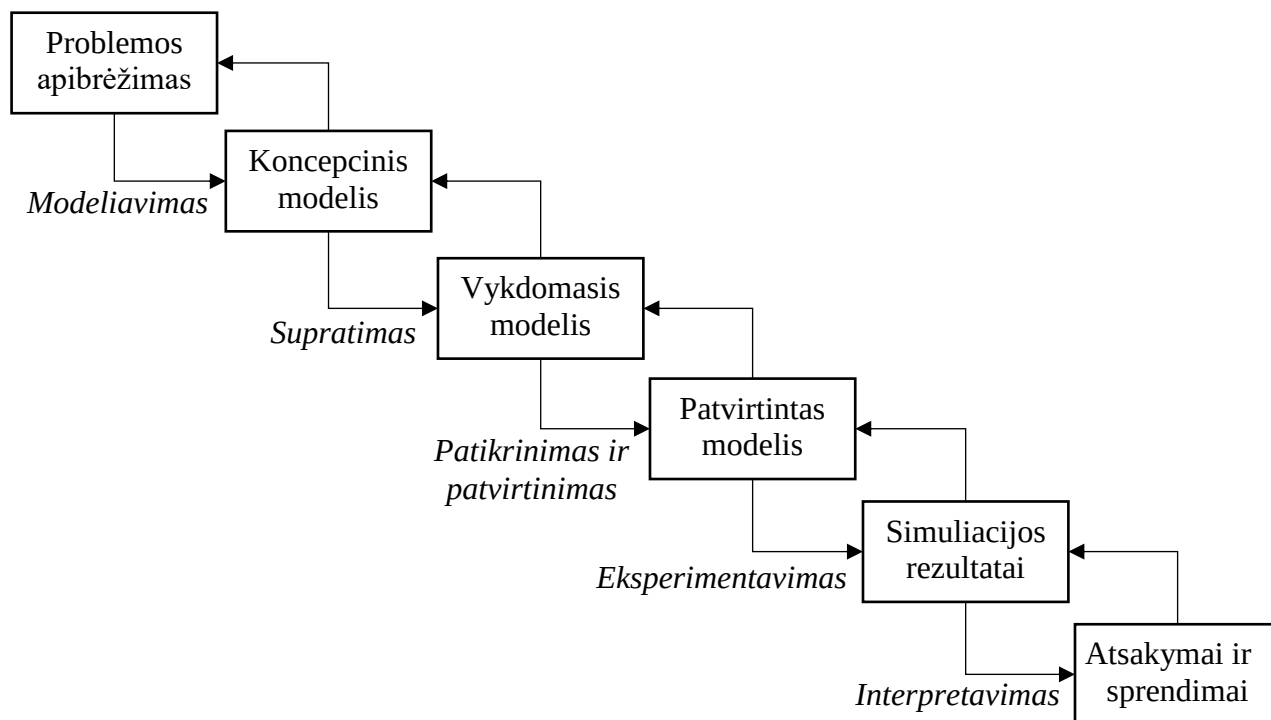
- Labai lanksti, ją galima pasitelkti imituojant net labai sudėtingas situacijas.
- Gali būti panaudota ieškant atsakymų į įvairius klausimus, pavyzdžiui galima įvertinti laiką, užimtumą, klaidų procentą.
- Lengvai suprantama, nebūtina būti tos srities specialistu ar turėti specifinių žinių.

Taip pat aprašomi ir trys verslo procesų simuliacijos metodo trūkumai (Van Der Aalst, 2015):

- Simuliacija gali labai ilgai užtrukti, jei norima gauti patikimus rezultatus negalima trumpinti simuliacijos proceso, būtina ją vykdyti tiek kartų, kol rezultatų suvestinė taps panaši į realų proceso vykdymą.
- Rezultatai gali būti nepatikimi, tad labai svarbu atsargiai daryti interpretacijas.
- Simuliacija nesuteikia įrodymų, nes gauti imitavimo rezultatai gali neatitikti visų galimų situacijų realybėje, nes neįmanoma numatyti visko.

1.4.3. Verslo proceso simuliacijos vykdymo etapai

Simuliacijos etapams atvaizduoti dažnai naudojamas „krioklio“ metodas, kuriame etapas gali prasidėti neužsibaigus ankstesniems etapams ir gali būti paveiktas sekančių po jo etapų (Van Der Aalst, 2015). Simuliaciją sudaro šeši etapai: problemos apibrėžimas, koncepcinis modelis, vykdomasis modelis, patvirtintas modelis, simuliacijos rezultatai, atsakymai ir sprendimai. Simuliacijos etapų eiliškumas pavaizduotas 2 paveiksle.



2 pav. Simuliacijos etapai (sudaryta autoriaus remiantis Van Der Aalst, 2015)

Iš 2 paveikslo matyti, kad simuliacija prasideda problemos apibrėžimu, šiame etape apibūdinami tikslai ir nustatoma simuliacijos tyrimo apimtis, taip pat nurodomi klausimai, į kuriuos simuliacijos rezultatai turės atsakyti, pateikiant kiekybinius vertinimus (Van Der Aalst, 2015). Nustačius problemą prasideda modeliavimas ir kuriamas koncepcinis modelis, kuris apibrėžia objektų klases ir santykius tarp tų objektų, taip pat pateikiamos objektų kokybinės ir kiekybinės charakteristikos. Koncepcinis modelis turi būti suplanuotas taip, kad būtų numatyta kaip išmatuoti laukimo laiką ir išteklių užimtumo lygį. Vykdomojo modelio etape pradedama realizacija, koncepcinis modelis verčiamas į tokį, kurį jau bus galima simuliuoti kompiuteryje. Patvirtinto modelio etapo vykdymo metu pirmiausia vykdomasis modelis yra tikrinamas, ieškant kokybinių ir kiekybinių klaidų, po to jis tvirtinamas, lyginant modelį su faktine sistema. Simuliacijos rezultatų etape vykdomas eksperimentas iš kurio gaunami kiekybiniai rezultatai. Paskutiniame, atsakymų ir sprendimų, etape eksperimento metu gauti rezultatai yra interpretuojami ir suformuojama galutinė ataskaita, kurioje pateikiami atsakymai į problemos formulavimo etapo klausimus ir pasiūlymai.

1.4.4. Skirtumai tarp verslo proceso modelio ir simuliacijos modelio

Simuliacijos modelis yra kuriamas iš verslo proceso modelio ir savo struktūra smarkiai skiriasi nuo pirminio modelio, nes kuriant simuliacijai pritaikytą modelį reikia pridėti papildomą informaciją, detaliau aprašyti savybes, reakciją į tam tikrus sprendimus ar procesą veikiančios aplinkos pokyčius. Remiantis Averill'io M. Law'o konferencijos medžiaga (2009) bei M. I. Kellner'io, R. J. Madachy ir

D. M. Raffo straipsniu (1999) galima išskirti, kad simuliacijos modelyje gali atsirasti tokia papildoma informacija:

- Keliami modelio problema ir klausimai, kurie padės suprasti kaip ši problema gali būti sprendžiama bei rodikliai, kurie bus naudojami efektyvumui vertinti.
- Rezultatų kintamieji, kurie būtini norint atsakyti į keliamus klausimus. Dažniausiai naudojami kintamieji: pastangos, kaina, proceso ciklo trukmė, defektas, personalo reikalavimai ir panaudojimas, ekonominė nauda, produktyvumas ir sekos ilgis.
- Tikimybių pasiskirstymas, pavyzdžiui laikas nuo sistemos/produkto sugedimo iki pataisymo.
- Prielaidų dokumentas, kuriame surašytos ne tik modelio prielaidos, bet ir algoritmai, sąvokos bei duomenų santraukos. Tokio tipo dokumente taip pat gali būti proceso srauto ar sistemos išdėstymo schemos, modelio apribojimai, svarbios informacijos šaltiniai.
- Animacija, gali būti naudojama norint patikrinti simuliacijos modelio teisingumą.

Atsiradus naujiems duomenims, atsiranda ir nauji su duomenimis galintys iškilti keblumai: duomenys neparodo to, ką buvo norėta modeliuoti, jie nėra tinkamo tipo ar formato, apima matavimo, įrašymo ar apvalinimo klaidas, yra šališki, buvo surinkti nežinomomis sąlygomis (Law, 2009).

1.5. Duomenų gavyba verslo procesų simuliacijos rezultatuose

1.5.1. Duomenų gavybos apibrėžimas

Anot Safiye Ghasemi, Mania'os Ghasemi ir Mehrta'os Ghasemi, duomenų gavyba (angl. *data mining*) yra didelio duomenų kiekio analizė, kai šie duomenys saugomi kompiuteryje (2011). Iš įvairių įmonės duomenų saugojimo šaltinių surinkti duomenys yra tiriami, analizuojami, sumažinami iki reikiamo kiekio ir gali būti panaudojami pakartotinai. Duomenų gavyboje dažniausiai naudojami klasikiniai statistiniai metodai, bet gali būti naudojami ir automatiniai dirbtinio intelekto metodai.

Duomenų gavybos esmė yra surasti reikiamas žinias, kiekvienu atveju ieškoma žinių pagal konkrečią situaciją, tačiau išskiriamos trys bendros savybės, kurias visos ieškomos žinios turi turėti: tikslumas, suprantamumas ir įdomumas (Freitas, 2003). Taip pat vienas dažniausiai keliamų uždavinių atrastoms žinioms yra tam tikras gebėjimas prognozuoti reikiamus duomenis, kai remiantis ankstesniais duomenimis būtų galima numatyti sekančius.

Duomenų gavyba susideda iš trijų etapų (Gaur, 2012):

1. Duomenų paruošimas – apdoroti duomenis, kad jie tiktų reikiamam gavybos metodui. Paruošimas apima duomenų valymą, kai šalinamas triukšmas, taisomi neatitikimai ir

užpildoma trūkstama informacija, taip pat duomenų parinkimą, išankstinį apdorojimą ir išreiškimą, kai duomenys paverčiami gavybai reikiama forma.

2. Taisyklių gavimas – naudojami taisyklių iš duomenų gavimo metodai.
3. Taisyklių vertinimas – priklauso nuo taikymo, tačiau bet kuriuo atveju naudojami bendri tikslai: surasti optimaliausią taisyklių seką, išbandyti taisyklių tikslumą, nustatyti kiek žinių buvo gauta, nustatyti nuoseklumą tarp gautų taisyklių ir neuroninio tinklo apmokymo.

1.5.2. Duomenų gavybos technikų klasifikavimas

Gavus žinias iš simuliacijos rezultatų svarbu yra jas suprantamai apdoroti ir pateikti, tam pasitelkiamos įvairios duomenų gavybos technikos. Duomenų gavimo metodai leidžia nustatyti paslėptus, vien atlikus simuliaciją nematomus, rezultatus. Mokslinėje literatūroje galima rasti duomenų gavybos skaidymą į keturis dažniausiai taikomus metodus, kartais dar vadinamus technikomis: duomenų grupavimą (angl. *clustering*), klasifikavimą (angl. *classification*), asociacijų taisyklių atradimą (angl. *association*) ir regresiją (angl. *regression*) (Radeschütz, Schwarz ir Niedermann, 2015).

Minėtos analizės technikos leidžia pastebėti tam tikrą dėsningumą tarp duomenų rinkinių. Taikant duomenų grupavimą objektai vienas po kito atidedami pasirinktoje koordinačių sistemoje, kol galiausiai visi yra suskirstomi į tam tikras klases, kuriose objektai yra panašūs tarpusavyje, o klasės skiriasi vienos nuo kitų. Klasifikavimas labai panašus į grupavimą, tik jame elementai priskiriami iš anksto apibrėžtomis klasėmis. Asociacijos technika naudojama atskleisti ryšius tarp nesusijusių duomenų, ją vykdant duomenys nuskaitomi daug kartų kol surenkami asociacijas turinčių elementų rinkiniai. Regresijos metodas yra panašus į klasifikavimo, pagrindinis skirtumas, kad klasifikuojant gaunami diskretūs rezultatai, o taikant regresiją gaunamos nepertraukiamo vertinimo funkcijos.

1.5.3. Grupavimas

Grupavimo technika nesugrupuotus elementus priskiria grupėms, kai imami elementai ir atidedami pagal savo savybes, kol galiausiai taip atidėjus visus elementus grupės susidaro savaime (Feldkamp, Bergmann ir Strassburger, 2015). Elementai grupuojami taip, kad vienoje grupėje būtų labai panašūs tarpusavyje, tačiau smarkiai besiskiriantys nuo už grupės ribų esančių elementų. Kiekvienas objektas privalo priklausyti grupei į kurios narius yra labiausiai panašus ir kiekvienoje grupėje turi būti bent vienas elementas. Panašumas tarp elemento ir grupės yra vertinamas atstumu, šis matavimo vienetas išreiškia kaip du objektai yra panašūs lyginant tarpusavyje.

Grupavimas vykdomas su kiekvienu elementu atskirai, pažymint jį pagal turimas savybes atitinkamoje vietoje dvimatėje koordinačių sistemoje, kur kiekvienas taškas reiškia atskirą elementą.

Koordinatėse atidėjus visus elementus gaunamos grupės, o labai arti pagal savo savybes susisiję elementų grupė yra vadinama klasteriu.

1.5.4. Klasifikacija

Klasifikacijos metu kiekvienas duomenų elementas įvedamas į vieną iš anksto apibrėžtų klasių, kur kiekvienai klasei nurodytas specialus tikslo atributas (Caramiaux ir Tanaka, 2013). Kitaip sakant, vykdant klasifikaciją sprendžiama, kuriai kategorijai priskirti pasirinktą duomenų elementą, kai kategorijos jau yra žinomos prieš pradedant klasifikaciją. Klasifikacijoje naudojami du etapai: mokymosi ir testavimo. Naudojant klasifikaciją reikia rasti taisyklės, pasiekiant tik mokymosi rinkinį. Kai mokymosi procesas yra baigiamas ir surandamos visos taisyklės, šių taisyklių nuspėjamas veikimas vertinamas testo rinkinyje. Vienas svarbiausių klasifikavimo metodo tikslų yra maksimaliai padidinti klasifikacijos modelio nuspėjamą tikslumą.

Klasifikacijoje naudojami etapai:

1. Mokymosi (angl. *training*) – remiantis turimais pavyzdžiais sukuriamos visos klasės.
2. Testavimo (angl. *testing*) – imami nepažymėti duomenys ir naudojant klasifikavimo funkciją priskiriami anksčiau nustatytoms klasėms.

1.5.5. Asociacijos

Asociacijos taisyklės yra teiginiai, kurie padeda atskleisti ryšius tarp nesusijusių duomenų (Kumbhare ir Chobe, 2014). Asociacijos taisyklių metode yra sugeneruojamas tik vienas elementas, kitaip sakant gali būti naudojama tokia taisyklė kaip $X \cap Y \Rightarrow Z$, tačiau negali būti kelių elementų: $X \Rightarrow Y \cap Z$.

Asociacijos vykdymo metu duomenys nuskaitymi daug kartų, kad būtų surinkti asociacijas turinčių elementų rinkiniai. Siekiant, kad metodas būtų efektyvesnis, naudojamas elementų įvertinimo metodas, kurio tikslas yra atmesti elementus, kurie negali sudaryti didelių rinkinių ir nebūtų vykdomas labai nuo kitų duomenų nutolusių elementų skaičiavimas.

Asociacijos metodo žingsniai:

1. Skenuojant gautus duomenis sugeneruojami dažnų elementų rinkiniai.
2. Kiekvienam elementui sugeneruojamas palaikymo skaičius ir jie surašomi į pirmą rinkinį. Remiantis palaikymo skaičiumi atmetami tie elementai, kurių palaikymo skaičius mažesnis už nustatytą minimalų.

3. Sukuriamas antras rinkinys, kuriame pirmojo rinkinio elementai praplečiami su kitais elementais. Remiantis palaikymo skaičiumi vėl atmetami tie elementai, kurie netenkina palaikymo slenksčio.

Procesas kartojamas kol nebeliks elementų arba dažnų elementų rinkinių.

1.5.6. Regresija

Regresijos metodas parodo ryšį tarp elementų, kai vienas iš jų yra numatytas, o kitas, ar keli, prognozuojami pagal numatytąjį (Gupta, 2015). Numatyta reikšmė vadinama priklausomu kintamuoju, o prognozuojama – nepriklausomu. Paprasčiausia regresijos rūšis yra linijinė regresija, ji naudojama prognozuoti dviejų kintamųjų santykį, kurie įvardijami kaip X ir Y. Linijinėje regresijoje naudojama matematinė lygtis, kuri apibūdina kaip Y kintamasis susijęs su X, kai žinoma X reikšmė, Y atidėjimas regresijos linijoje ir linijos nuolydis.

Tiesinė regresija gali parodyti, kad tarp kintamųjų nėra jokio ryšio, jeigu linija tarp jų neturi jokio nuolydžio. Taip pat ryšys gali būti teigiamas, jeigu kreivė nukrypsta viršun Y ašies atžvilgiu, o jei apačion – neigiamas.

Swati Gupta išskiria 6 linijinės regresijos metodo žingsnius (2015):

1. Pasirenkamos kintamųjų X ir Y reikšmės.
2. Apskaičiuojamas X kintamojo vidurkis.
3. Apskaičiuojamas Y kintamojo vidurkis.
4. Apskaičiuojamas regresijos koeficientas β (nuolydis).
5. Apskaičiuojamas regresijos koeficientas α (Y atidėjimas regresijos linijoje).
6. Įstačius visas reikšmės į lygtį $Y = \alpha + \beta X$ apskaičiuojama regresija.

1.5.7. Duomenų gavybos metodų palyginimas

Norint vaizdžiai palyginti duomenų gavybos metodus sukurta palyginimo lentelė. 1 lentelėje lyginami anksčiau apžvelgti keturi metodai: grupavimas, klasifikacija, asociacijos ir regresija. Kiekvienam iš metodų nurodomas veikimo principas, kiek elementų yra imama vienu metu vykdant metodą, sudėtingumas, privalumai bei trūkumai.

1 lentelė. Duomenų gavybos metodų palyginimas

Metodas Lyginimo kriterijus	Grupavimas	Klasifikacija	Asociacijos	Regresija
Veikimo principas	Imant po vieną elementą jie atidedami pagal turimas savybes, kol galiausiai susidaro klasės.	Elementai skirstomi į iš anksto numatytas klases.	Ieškoma ryšių tarp elementų, pagal kuriuos jie surenkami į atitinkamus rinkinius.	Ieškoma ryšių tarp elementų, kai vienas pasirenkamas kaip numatytasis, o kitas jam prognozuojamas.
Keli elementai imami vienu metu	1	1	2	≥ 2
Sudėtingumas	Nesudėtinga	Nesudėtinga	Sudėtinga	Sudėtinga
Privalumai	Labai lengvai įgyvendinama, nereikalauja jokios išankstinės informacijos.	Lengvai suprantama ir įgyvendinama, tinka sudėtingų problemų analizei, naudojama prognozavimui.	Gera filtruoja smarkiai išsiskiriančius duomenis, įvertina duomenų tikslumą.	Lengva nustatyti statistines savybes, linijinė regresija nelabai sudėtinga, nelinejinė tinka sudėtingų problemų analizei, naudojama prognozavimui.
Trūkumai	Negalima pakeisti priimto sprendimo.	Jautru „triukšmui“, reikia iš anksto numatyti klases, tai gali būti netikslu.	Lėtas veikimas, nes duomenys nuskaitomi daug kartų, kol sukuriama visi rinkiniai.	Rezultatai, palyginti su kitais metodais, abstraktūs, taikomi sudėtingi skaičiavimai.

Atsižvelgiant į visus kriterijus galima teigti, kad grupavimas ir klasifikacija yra labai panašūs metodai, pagrindinis skirtumas, kad klasifikacijos metodui būtina prieš klasifikuojant duomenis iš anksto numatyti visas galimas klases, be to klasifikacija gali būti naudojama prognozavimui. Panašiai galima teigti ir apie asociacijų ir regresijos metodus, abiejuose šiuose metoduose ieškoma ryšių tarp duomenų elementų, tačiau regresijoje naudojami skaičiavimai yra sudėtingesni, o asociacijos pagrindinis privalumas – gerai filtruojamas triukšmas, kurį galima apibūdinti kaip ryškiai išsiskiriančius duomenis.

1.6. Apibendrinimas ir rezultatai

Magistro baigiamojo darbo literatūros analizės skyriuje buvo apžvelgtos penkios svarbiausios su darbu susijusios temos: verslo procesai, verslo procesų modeliavimas, verslo procesų analizė ir optimizavimas, verslo procesų simuliacija bei duomenų gavybos metodai.

Apžvelgus verslo procesus paaiškėjo, kad juos galima suskaidyti į statinius, dinامينius, siekiančius tikslo ir priklausančius nuo konteksto. Patys paprasčiausi yra statiniai verslo procesai, nes jų struktūra yra žinoma iš anksto ir nesikeičia. Statinių procesų didžiausias privalumas yra tas, kad juos nesudėtinga perduoti automatiniam vykdymui, nes nereikalingas žmogaus įsikišimas, be to, jie puikiai pasitarnauja norint suprasti ar pristatyti procesą. Tačiau siekiant atsižvelgti į galimai kintančius verslo poreikius patogiau naudoti dinامينius verslo procesus, nes jie neturi griežtai nustatytos veiklų sekos ir gali greitai prisitaikyti prie pokyčių. Tikslo siekiančių ir nuo konteksto priklausančių verslo procesų esmė yra tokia pat kaip dinامينių, tačiau juos galima suprasti kaip dar labiau praplėstas, atsižvelgiant į tam tikrus poreikius, dinامينių verslo procesų šakas. Tikslo siekiantys procesai orientuojasi tik į galutinį rezultatą, o ne individualius veiksmus, tad sprendimai dėl veiksmų atliekami nuolat tiriant atnaujinamą informaciją apie verslo aplinką. Nuo konteksto priklausančios verslo procesai suprojektuojami labai lanksčiai, kad kontekstą, kuriuo jie remiasi būtų galima keisti ir papildyti bet kuriuo metu, atsižvelgiant tiek į organizacijos, tiek į išorinį kontekstą. Trumpai apžvelgus verslo procesų valdymą nagrinėtų darbų autoriai pabrėžia, kad norint valdyti verslo procesus jie jungiami su informacinėmis technologijomis ir automatizuojami.

Analizuojant mokslinę literatūrą apie verslo procesų modeliavimą paaiškėjo, kad jie klasifikuojami įvairiais pjūviais, šiame darbe išskirti trys klasifikavimo būdai. Anksčiausiai, dar 2000 metais, aprašytas klasifikavimas į keturias grupes pagal požiūrį į verslo proceso pobūdį: verslo procesai suprantami kaip deterministinės mašinos su fiksuota veiksmų seka, sudėtingos dinaminės ir interaktyvios sistemos su dinamiškai kintančia elgsena, sąveikaujančios sistemos su sudėtinga vidine struktūra, kuri suvokiama kaip visuma, o ne atskiri komponentai bei socialiniai konstrukai, kai atsižvelgiama į tai kaip procesus suvokia žmonės. Kiek vėliau pradėta skaidyti verslo procesus pagal jų siekiamus tikslus ir aktyvumą, kai aktyvumas nusako komunikaciją su vartotojais. Galiausiai pasiūlytas klasifikavimas pagal struktūrines verslo procesų modelių ypatybes: grafiniai, matematiniai arba formalūs ir verslo procesų modeliavimo kalbomis paremti modeliai. Tuo pačiu glaustai aprašomos dažniausiai naudojamos verslo procesų modeliavimo kalbos: UML ir BPMN.

Atsižvelgiant į analizuotą literatūrą aprašyti trys verslo procesų analizės tipai: stebėjimo analizė, veiklos įvertinimas patvirtinant ir patikrinant bei veiklos įvertinimas algoritmais. Stebėjimo analizė priskiriama grafinių modelių verslo procesų klasei, veiklos įvertinimo patvirtinant ir patikrinant

analizė – matematinių modelių, o veiklos įvertinimo algoritmais analizė – verslo procesų kalbomis paremtų modelių. Be to, šie analizės tipai gali būti naudojami kartu, jungiant juos vienus su kitais, o simuliacija apima visus tris tipus, būtent dėl to simuliaciją galima apibūdinti kaip turinčią didžiausias analizės galimybes. Atsižvelgiant į analizės rezultatus galima tobulinti pasirinktą verslo proceso sritį, ją optimizuoti.

Buvo apibrėžti trys pagrindiniai verslo procesų simuliacijos privalumai: lankstumas, panaudojimas ieškant atsakymų į rūpimus klausimus bei lengvai suprantami rezultatai. Taip pat išskirti ir trys trūkumai: ilga vykdymo trukmė patikimiems rezultatams gauti, galimybė, kad rezultatai nepaisant simuliacijos kartojimų kiekio bus nepatikimi ir tikslumo įrodymų trūkumas, nes neįmanoma numatyti visų galinčių įvykti atvejų. Išskirta, kad simuliacijoje naudojami šeši vykdymo etapai: problemos apibrėžimas, koncepcinis modelis, vykdomasis modelis, patvirtintas modelis, simuliacijos rezultatai, atsakymai ir sprendimai. Visi šie etapai eina iš eilės, tačiau sekantis etapas gali prasidėti ir neužsibaigus ankstesniam bei visada galima grįžti prie ankstesniojo etapo. Galiausiai, remiantis nagrinėta literatūra išskirta, kad verslo proceso modelis nuo simuliacinio modelio skiriasi turima papildoma informacija, tokia kaip: keliama problema ir klausimai, rezultatuose numatomi kintamieji, tikimybės, dokumentas, kuriame surašytos modelio prielaidos ir kita reikiama informacija bei animacija, kuri gali būti naudojama siekiant aiškiau pavaizduoti simuliacijos rezultatus.

Apžvelgus mokslinę literatūrą apie duomenų gavybą išskirtos keturios dažniausiai naudojamų metodų klasės: duomenų grupavimas, klasifikavimas, asociacijų taisyklių atradimas ir regresija. Grupuojant imama po vieną elementą ir jie atidedami pagal turimas savybes, kol visi elementai išdėstomi į savaime susidarančias klases. Klasifikacijos metu elementai taip pat suskaidomi į klases, tačiau jos būna iš anksto apibrėžtos. Naudojant asociaciją ieškoma ryšių tarp elementų ir pagal rastus ryšius jie surenkami į atitinkamus rinkinius. Regresija taip pat ieško ryšių tarp elementų, tačiau vienas elementas pasirenkamas kaip numatytasis, o kitas jam prognozuojamas ir vienu metu ieškoma ryšių tik tarp pasirinktų elementų.

Atlikus literatūros analizę buvo apžvelgta tyrimo sritis, nuo pačios abstrakčiausios temos – verslo procesų, iki labai konkrečios – duomenų gavybos metodų. Apžvelgus sritį, kurioje toliau bus vykdomas darbas paaiškėjo, kad ji iš tiesų yra labai aktuali šiuolaikiniame versle, tad tikrai svarbu tobulinti ir dar labiau efektyvinti verslo procesų simuliacijos rezultatų analizės egzistuojančias galimybes, siekiant gauti kuo naudingesnę informaciją. Taigi atsižvelgiant į šį poreikį pasirinkta tolimesnio tyrimo sritis – taisyklių gavimas iš simuliacijos rezultatų.

2. DUOMENŲ GAVYBOS ALGORITMŲ TIPŲ ANALIZĖ

Šiame skyriuje yra analizuojami duomenų, konkrečiau taisyklių, gavimo iš verslo procesų simuliacijos rezultatų algoritmų tipai. Analizė leis ne tik susipažinti su dažniausiai naudojamais duomenų gavybos algoritmų tipais, bet ir juos palyginus nustatyti tinkamiausią algoritmą tolimesnei darbo eigai.

Taisyklių iš verslo procesų simuliacijos rezultatų gavimo algoritmai leidžia apdoroti simuliacijos metu gautus duomenis, siekiant juos susisteminti ir gauti iš pirmo žvilgsnio nepastebimas žinias. Pačią duomenų gavybą galima apibūdinti kaip sistemingą didelių duomenų rinkinių analizę. Tokios analizės metu iš duomenų rinkinio sužinoma naudinga informacija, pavyzdžiui, taisyklės, pagal kurias kartais galima prognozuoti naujus duomenis ar tobulinti patį verslo procesą. Gauti duomenis iš atliktos simuliacijos rezultatų galima įvairiais metodais, iš kurių dažniausiai naudojami šie penki algoritmų tipai: sprendimų medžių, genetiniai, mišrūs algoritmai, sudaryti iš sprendimų medžių ir genetinių algoritmų, K-vidurkių algoritmai ir neuroniniai tinklai.

2.1. Sprendimų medžių algoritmai

Sprendimų medis – tai grafikas, kuriame naudojami grafiniai sprendimai ir įmanomas poveikis po jų, įskaitant įvykių rezultatus, išteklių sąnaudas ir naudingumą (Bhargava, Sharma, Bhargava ir Mathuria, 2013). Sprendimų medis gali būti vadinamas klasifikacijos medžiu, kuris naudojamas norint nustatyti klasifikacijos funkciją, kurią sudaro iš nepriklausomų įvesties atributų gauti priklausomi atributai. Kitaip sakant, kiekviena medžio šaka yra tarsi padedantis klasifikuoti duomenis klausimas, kai klausimų kiekis priklauso nuo to, kaip smulkiai norima išskirstyti duomenis. Kiekvienas sprendimų medžio šakos mazgas yra pasirinkimas tarp galimų alternatyvų, o kiekvienas lapas atitinka sprendimą. Medis prasideda nuo šaknies mazgo, o toliau, pagal galimus pasirinkti variantus sprendimai skaidomi į šakas. Galutinai suformuotame medyje kiekviena šaka atspindi galimą sprendimų scenarijų ir jo rezultatą. Egzistuoja daug įvairių sprendimų medžių algoritmų, iš jų populiariausi yra ID3 (*Iterative Dichotomiser 3*), C4.5 (*successor of ID3*), CART (*Classification And Regression Tree*), CHAID (*CHi-squared Automatic Interaction Detector*) ir J48 (C4.5 versija parašyta Java kalba).

Sprendimų medžiai yra laikomi vienu iš galingiausių metodų, taikomų duomenų gavybos procese, nes juos galima panaudoti daugelyje sričių, pavyzdžiui, duomenų ir teksto gavybos, informacijos gavimo, mašinų mokymosi ir modelio atpažinimo. Sprendimo medžio metodas turi penkis pagrindinius privalumus: medžio struktūrą nesudėtinga suprasti vartotojui, jis gali naudoti įvairius įvesties duomenis, gali apdoroti klaidingus duomenis arba trūkstamas vertes, kokybiškai dirbti su nedideliu bandymu skaičiumi, taip pat gali būti vykdomas įvairiose platformose.

Medžio algoritmo sudėtingumas matuojamas pagal bendrą mazgų ir lapų skaičių, medžio gylį ir atributų kiekį. Lengviausia dirbti su nedideliais – nesudėtingais medžiais, kuriuos galima valdyti genėjimo (angl. *pruning*) metodu, ši technika naudojama teisingai klasifikuoti neapibrėžtus atvejus duomenų rinkiniuose. Genėjimo tikslas yra sumažinti medį taip, kad būtų pasiekiamas kiek galima efektyvesnis rezultatas. Be to, genėjimas gali būti atliekamas jau sukūrus medį arba jį kuriant.

Sprendimo medžiuose naudojama indukcija, kai siekiama įrodyti, kad jei koks nors teiginys yra teisingas, tai kiti sekos teiginiai taip pat bus teisingi. Indukcija sprendimo medžiuose glaudžiai susijusi su taisyklėmis, nes jos ir yra svarbiausieji teiginiai, kurių teisingumas yra vertinamas. Sprendimų medyje taisykle vadinamas kiekvienas kelias, kuris prasideda nuo sprendimo medžio šaknies ir baigiasi vienu iš lapų.

Sprendimų medžiai skirstomi į du metodus:

1. Vienarūšis sprendimų medis – skaidymas atliekamas naudojant vieną atributą vidiniuose mazguose (pavyzdžiui $X < 2$). Tokio tipo medžiams yra daug algoritmų, pavyzdžiui ID3, C4.5, J48.
2. Nevienarūšis sprendimų medis – naudojamas daugiau nei vienas atributas (pavyzdžiui $X + Y > 10$). Kuriant nevienarūšius sprendimų medžius atliekami trys žingsniai: pirmiausia patikrinama ar visi atvejai priklauso tai pačiai klasei, tada kiekvienam atributui apskaičiuojama informacija ir informacijos gavimas ir galiausiai randamas geriausias skilimo atributas.

Nevienarūšis sprendimų medis sugeba apibendrinti atributų koreliaciją – statistinį ryšį tarp atributų, kai du atributai yra nepriklausomi tuo atveju, jei koreliacijos koeficientas lygus nuliui. Vienarūšiai medžiai bando vieną atributą daugiau nei vieną kartą, todėl gali būti neefektyvūs, tuo tarpu nevienarūšiai medžiai atlieka skirtingus bandymus su duomenimis naudojant daugiau nei vieną atributą, tad yra gerokai efektyvesni, tačiau sudėtingesni.

Sprendimų medžių trūkumai: algoritmo vykdymo laiko sudėtingumas atitinka medžio gylį, kuris negali būti didesnis už atributų skaičių. Medžio gylis yra susijęs su pavyzdžių skaičiumi, tad algoritmas smarkiai sulėtėja turint didelius duomenų kiekius.

Sprendimų medžio kūrimą sudaro trys pagrindiniai žingsniai (Bhargava, Sharma, Bhargava ir Mathuria, 2013):

1. Patikrinti ar visi atvejai priklauso tai pačiai klasei.
2. Kiekvienam atributui apskaičiuoti informaciją ir informacijos naudą (angl. *information gain*).
3. Rasti geriausią skaidymo atributą, priklausomai nuo pasirinkimo kriterijaus.

Informacijos nauda pasitelkiama apskaičiuoti kiekvieno iš galimų išsiskojimui pasirinkti atributų naudingumui, o jai apskaičiuoti naudojama entropija. Entropija – duomenų netvarkos arba neapibrėžtumo matas, kuris matuojamas bitais, netais arba banais. Informacijos nauda svarbi duomenų skaidymui į medžio šakas, nes efektyvus medis gaunamas tuomet, kai jis pagrįstas didžiausia nauda, kitaip sakant anksčiau medžio mazgams pasirenkami atributai su didesne naudos verte. Nauda apskaičiuojama žemiau pateikta formule (1):

$$Nauda(p, j) = Entropija(p - Entropija(j|p)) = - \sum_{j=1}^n \frac{|p_j|}{|p|} \log \frac{|p_j|}{|p|} - \frac{|p_j|}{|p|} \log \frac{|p_j|}{|p|}. \quad (1)$$

Naudos lygtyje p kintamasis nusako visų reikšmių skaičių, p_j – konkrečios vienos reikšmės pasikartojimų skaičių. Žemiau pateikiamas sprendimų medžio kūrimo pavyzdys, naudojant atributų naudos apskaičiavimą, kuris aprašytas Badr'o Hssina, Abdelkarim'o Merbouha, Hanane Ezzikouri ir Mohammed'o Erritali straipsnyje (2014). Aprašomas sprendimų medžio kūrimas, kai prie tam tikrų oro sąlygų žaidimas įvyksta arba ne. Pavyzdyje naudojami keturi oro sąlygas nusakantys atributai: vaizdas (saulėta, debesuota, lietinga), temperatūra (karšta, maloni, šalta), drėgmė (aukšta, normali), vėjas (silpnas, stiprus). Iš viso sąlygos ir įvykę žaidimai buvo stebimi 14 dienų, surinkti pradiniai duomenys nurodyti 2 lentelėje.

2 lentelė. Sprendimų medžio panaudojimo pavyzdžio pradiniai duomenys (Hssina, Merbouha, Ezzikouri ir Erritali, 2014)

Diena	Vaizdas	Temperatūra	Drėgmė	Vėjas	Žaidžiama
1.	Saulėta	Karšta	Aukšta	Silpnas	Ne
2.	Saulėta	Karšta	Aukšta	Stiprus	Ne
3.	Debesuota	Karšta	Aukšta	Silpnas	Taip
4.	Lietinga	Maloni	Aukšta	Silpnas	Taip
5.	Lietinga	Šalta	Normali	Silpnas	Taip
6.	Lietinga	Šalta	Normali	Stiprus	Ne
7.	Debesuota	Šalta	Normali	Stiprus	Taip
8.	Saulėta	Maloni	Aukšta	Silpnas	Ne
9.	Saulėta	Šalta	Normali	Silpnas	Taip
10.	Lietinga	Maloni	Normali	Silpnas	Taip
11.	Saulėta	Maloni	Normali	Stiprus	Taip
12.	Debesuota	Maloni	Aukšta	Stiprus	Taip
13.	Debesuota	Karšta	Normali	Silpnas	Taip
14.	Lietinga	Maloni	Aukšta	Stiprus	Ne

Pirmiausia ieškoma šaknies mazge labiausiai tinkamo atributo, tam pasirenkamas atributas, kurio naudingumas yra didžiausias. Kiekvieno atributo nauda apskaičiuojama pagal naudos formulę, iš galutinio rezultato entropijos atėmus atributo reikšmių susumuotas entropijas: (2) formulėje apskaičiuojama žaidimo įvykimo entropija, (3) – vaizdo atributo entropija, kuriai dar reikia

apskaičiuoti kiekvienos galimos reikšmės entropijas, jos apskaičiuojamos (4), (5) ir (6), galiausiai gautos entropijų reikšmės panaudojamos pradinėje vaizdo atributo formulėje (7).

$$Entropija(\check{Zaid\check{z}iama}) = -\frac{9}{14} \cdot \log_2 \frac{9}{14} - \frac{5}{14} \cdot \log_2 \frac{5}{14} = 0,94; \quad (2)$$

$$Nauda(\check{Zaid\check{z}iama}, Vaizdas) = Entropija(\check{Zaid\check{z}iama}) - \frac{5}{14} \cdot (Entropija \check{Zaid\check{z}iama}_{saul\check{e}ta}) - \frac{4}{14} \cdot Entropija(\check{Zaid\check{z}iama}_{lietinga}) - \frac{5}{14} \cdot (Entropija \check{Zaid\check{z}iama}_{debesuota}); \quad (3)$$

$$Entropija(\check{Zaid\check{z}iama}_{saul\check{e}ta}) = -\frac{2}{5} \cdot \log_2 \frac{2}{5} - \frac{3}{5} \cdot \log_2 \frac{3}{5} = 0,971; \quad (4)$$

$$Entropija(\check{Zaid\check{z}iama}_{lietinga}) = -\frac{4}{4} \cdot \log_2 \frac{4}{4} - \frac{0}{5} \cdot \log_2 \frac{0}{5} = 0; \quad (5)$$

$$Entropija(\check{Zaid\check{z}iama}_{debesuota}) = -\frac{3}{5} \cdot \log_2 \frac{3}{5} - \frac{2}{5} \cdot \log_2 \frac{2}{5} = 0,971; \quad (6)$$

$$Nauda(\check{Zaid\check{z}iama}, Vaizdas) = 0,94 - \frac{5}{14} \cdot 0,971 - \frac{4}{14} \cdot 0 - \frac{5}{14} \cdot 0,971 = 0,246. \quad (7)$$

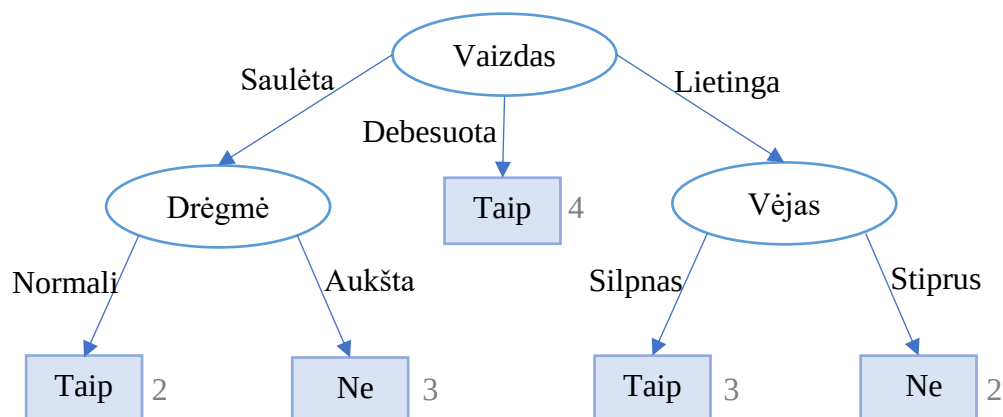
Kitų atributų naudos pateikiamos (8), (9) ir (10) formulėse:

$$Nauda(\check{Zaid\check{z}iama}, Temperatūra) = 0,0289; \quad (8)$$

$$Nauda(\check{Zaid\check{z}iama}, Drėgmė) = 0,1515; \quad (9)$$

$$Nauda(\check{Zaid\check{z}iama}, Vėjas) = 0,048. \quad (10)$$

Galutinis suformuotas medis pateiktas 3 paveiksle.

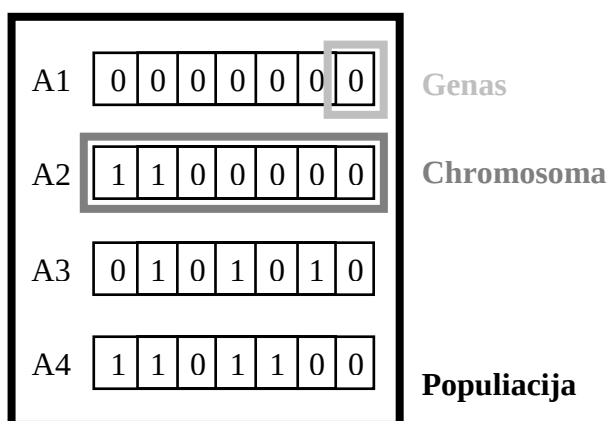


3 pav. Sprendimų medžio formavimo, remiantis atributų nauda (sudaryta autoriaus remiantis Hssina et al., 2014)

Apskaičiavus visų atributų naudas, akivaizdu, kad atributas *Vaizdas* turi didžiausią naudingumą – 0,246, tad jis ir pasirenkamas kaip šaknies mazgas. Šis atributas turi tris galimas reikšmes, todėl iš šaknies išeina trys šakos: saulėta, lietinga ir debesuota. Toliau išskaidžius reikšmes į šias tris šakas, skaičiuojant naudingumą, ieškomi kiti mazgai.

2.2. Genetiniai algoritmai

Genetiniai algoritmai imituoja natūralios evoliucijos procesą ir dažnai yra pasitelkiami duomenų gavyboje, čia jie naudojami siekiant optimizuoti kitų rūšių duomenų gavybos algoritmų parametrus (Agarwal ir Tech, 2015). Algoritmas pradedamas nuo sprendimų rinkinio, vadinamo populiacija, kuris naudojamas suformuoti naujai populiacijai, kuri tikimasi, kad bus geresnė už pradinę. Populiaciją sudaro chromosomos, kurios pačios sudarytos iš genų, kurie turi savo reikšmes, šios reikšmės yra skaitinės. Populiacijos chromosomas galima aprašyti keliais būdais: dvejetainiu, kai visi elementai yra lygūs 0 arba 1, tikrųjų verčių arba slankiojo kablelio, kai elementai aprašomi skaičiais su kableliu, pvz.: 0,1 ir sveikųjų skaičių, kaip 1, 2, 3. Dvejetainių verčių populiacijos pavyzdys pateiktas 4 paveiksle.



4 pav. Genetinio algoritmo populiacijos, chromosomos ir geno pavyzdys su dvejetainėmis vertėmis

Pagrindiniai genetinių algoritmų žingsniai:

1. Pradedama nuo atsitiktinai sugeneruotos populiacijos iš n elementų inicijavimo, kuris vykdomas atsitiktiniu arba euristiniu būdu;
2. Apskaičiuojama kiekvieno elemento tinkamumo (angl. *fitness*) funkcija populiacijoje;
3. Kartojami tokie veiksmai, kol bus sukurta n palikuonių:
 - a. Išrenkama tėvinių elementų pora iš dabartinės populiacijos;
 - b. Pora kryžminama, kad susidarytų du palikuonys;
 - c. Gauti palikuonys mutuoja ir susidarę elementai pridedami prie naujos populiacijos;
 - d. Apskaičiuojamas naujų elementų tinkamumas.
4. Paliekama išgyventi n geriausių elementų sekančiai kartai;

5. Einama į 3 žingsnį.

Tėviniai elementai išrenkami iš esamos populiacijos tam, kad iš jų būtų galima sukurti palikuonis kitoms kartoms. Šis žingsnis yra labai svarbus, nes tinkamai atrinkus tėvinius elementus iš jų galima gauti geresnius palikuonis. Khalid Jebari, Mohammed Mediafi ir Abdelaziz Elmoujahid (2013) išskiria netgi šešis galimus tėvinių elementų išrinkimo būdus: ruletės rato (angl. *the roulette wheel selection*), universalų stochastinį (angl. *the stochastic universal sampling*), linijinį reitingo (angl. *the linear rank selection*), eksponentinį reitingo (angl. *the exponential rank selection*), turnyro (angl. *the tournament selection*) ir nukirtimo (angl. *the truncation selection*). Ruletės rato išrinkimo pagrindinė ypatybė yra tokia, kad kiekvienam populiacijos elementui suteikiama išrinkimo tikimybė, kuri lygi tinkamumo funkcijai. Universalus stochastinis ir linijinio reitingo išrinkimas yra ruletės rato variantai, kuriuose siekiama sumažinti susilieimo tikimybę, kad nebūtų atrinktas tas pats elementas vietoje dviejų tėvinių. Eksponentinio reitingo išrinkimas yra toks pat kaip ir minėti aukščiau, tik tikimybės didinamos eksponentiškai. Turnyro išrinkimo būdas tėviniams elementams yra paremtas atsitiktiniu k elementų išrinkimu, kai išrinkti elementai reitinguojami pagal tinkamumo funkcijas ir pasirenkamas pats tinkamiausias. Nukirtimo išrinkimo būdu populiacijos elementai rūšiuojami pagal jų tinkamumą ir tik geriausieji atrenkami kaip tėviniai.

Įvairios kryžminimo technikos naudojamos siekiant kuo greičiau rasti optimalų sprendimą, kad reikėtų generuoti kuo mažiau kartų. Standartiniame kryžminime išskiriama dešimt technikų: 1 taško (angl. *1-point*), k taškų (angl. *k-point*), maišytas (angl. *shuffle*), sumažinto surogato (angl. *reduced surrogate*), vienodas (angl. *uniform*), vidurkio (angl. *average*), diskretus (angl. *discrete*), plokščias (angl. *flat*), euristinis / tarpinis (angl. *heuristic / intermediate*) ir pagrįstas statistika prisitaikantis nevienodas (angl. *statistics-based adaptive nonuniform*).

1 taško kryžminimo metu tėviniuose elementuose atsitiktinai pasirenkamas vienas taškas, jame chromosomos yra perskeliamos ir vienos atskelta dalis prijungiama prie kitos likusios po atskėlimo dalies. K taškų kryžminimas vyksta taip pat, kaip ir 1 taško, tik atsitiktinai pasirenkami keli skėlimo taškai. Maišytame kryžminime vienoduose taškuose sumaišomi chromosomų genai, po to vykdomas 1 taško kryžminimas ir genai vėl maišomi, kad grįžtų į pradines savo pozicijas. Sumažinto surogato kryžminime peržiūrimi visi tėvų genai, ieškant taškų, kuriuose perkirtus tėvų genai skirtųsi, iš rastų taškų atsitiktinai pasirenkamas vienas ir vykdomas 1 taško kryžminimas. Vienodame kryžminime palikuoniams atsitiktinai parenkami genai iš pirmojo ar antrojo tėvinio elemento, kai pirmasis palikuonis išsirenka, o antrasis pasiima likusius. Vidurkio kryžminime iš dviejų tėvų gaunamas tik vienas palikuonis, apskaičiavus tėvinių elementų genų vidurkius. Diskretus būdas taip pat generuoja tik vieną palikuonį, atsitiktinai parenkant genus. Plokščias kryžminimas panašus į diskretųjį, tačiau

pasirinktas genas turi būti mažiausias arba didžiausias. Euristinis / tarpinis ir pagrįstas statistika prisitaikantis nevienodas kryžminimai yra gerokai sudėtingesni, jie palikuonies kūrimui pasitelkia sudėtingus skaičiavimus.

Nitasha'os Soni ir Tapas'o Kumar'o straipsnyje (2014) išskiriami devyni mutacijos žingsnio tipai: įterpimo (angl. *insert*), inversijos (angl. *inversion*), grumtynių (angl. *scramble*), sukeitimo (angl. *swap*), apvertimo (angl. *flip*), keitimosi (angl. *interchanging*), keitimo atvirkštiniu (angl. *reversing*), vienodumo (angl. *uniform*) ir šliaužimo (angl. *creep*).

Atrankos metu išrenkama, kuriuos iš naujos populiacijos narių reikia palikti kitoje kartoje, o kuriuos atmesti. Išskiriami du dažniausiai naudojami atrinkimo būdai: paremtas amžiumi ir paremtas tinkamumu. Amžiumi paremtoje atrankoje visada yra išmetami nariai, kurie kartoje yra seniausieji, neatsižvelgiant į jų tinkamumą. Paremtoje tinkamumu atrankoje naujai gauti palikuonys išstumia narius, kurių tinkamumas buvo pats mažiausias. Šiuo atveju, jei populiacijoje yra keli nariai su vienodu tinkamumu, iš jų gali būti pašalinamas bet kuris, atsitiktinumo principu.

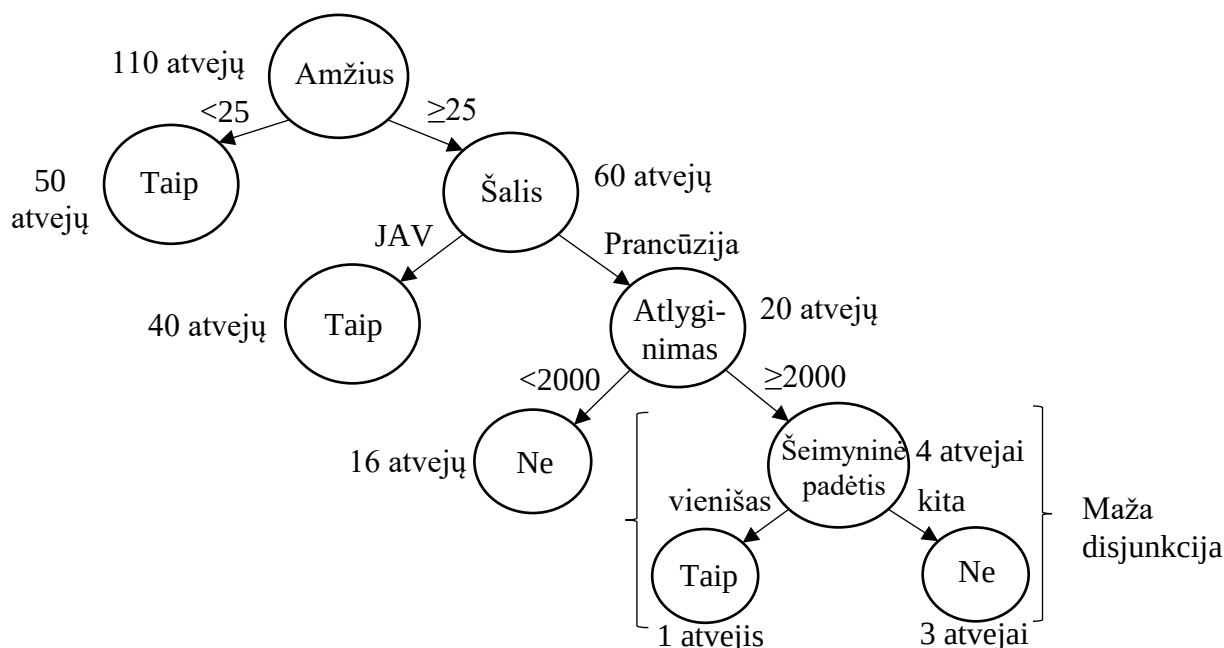
1.3. Mišrūs sprendimų medžių / genetiniai algoritmai

Mišrių sprendimų medžių / genetinių algoritmų metodo (angl. *hybrid decision tree/genetic algorithm method*) pagrindinė idėja – nedidelių disjunkcijų naudojimas duomenų gavyboje, kai nedidelės disjunkcijos klasifikuojamos pagal genetinį algoritmą, o didelės – sprendimų medžio algoritmą (Carvalho ir Freitas 2004).

Šiuo metodu klasifikuojant žinias jos išreiškiamos taisyklių rinkiniu, kurios nurodomos sąlygos forma: JEI < sąlyga > TADA < prognozė (klasė) >. Aprašyti taisyklės logine forma pasirinkta, nes toks pateikimas yra suprantamas intuityviai. Paprastai loginiu požiūriu nustatytos taisyklės išreiškiamos disjunkcijos forma, kai kiekviena taisyklė reiškia disjunkciją, o taisyklės sąlyga konjunkciją. Disjunkcijos operacija, tai loginė sudėtis, kuri išreiškiama žodžiu „arba“ ir jos rezultatas yra teigiamas, jeigu bent vieno operando reikšmė yra teisinga. Tuo tarpu konjunkcija – loginė daugyba, kuri išreiškiama jungtuku „ir“, o jos rezultatas yra teigiamas tik tada, jeigu abiejų operandų reikšmės yra teisingos.

Mažų disjunkcijų, kaip taisyklių, koncepcija pavaizduota 5 paveiksle. Šiame paveiksle pavaizduota sprendimų medžio dalis, kurioje vaizduojamas suaugusių asmenų duomenų pasiskirstymas pagal tam tikras sąlygas. Pateikto paveikslo apačioje dešinėje esantys paskutiniai mazgai gali būti laikomi mažomis (smulkiomis) disjunkcijomis, nes juose numatytas mažas sąlygas atitinkančių atvejų skaičius: vienas ir trys. Tuo tarpu didelė (stambia) disjunkcija gali būti laikomi viršutiniai medžio mazgai, kuriuose vaizduojamas didelis sąlygas atitinkančių atvejų skaičius, kaip

50 atvejų, kai tiriamųjų amžius yra mažesnis nei 25 metai ir 60 atvejų, kai tiriamųjų amžius lygus arba didesnis nei 25 metai.



5 pav. Mažos disjunkcijos pavyzdys sprendimų medyje (sudaryta autoriaus remiantis Carvalho ir Freitas, 2004)

Mišrūs sprendimų medžių/genetiniai algoritmai taisyklių ieško dviem etapais:

1. Vykdomas sprendimų medžio indukcijos algoritmas, dar žinomas kaip C4.5. Jo metu gautas medis paverčiamas taisyklėmis, kai kiekviena taisyklė išreiškiama disjunkcijos forma. Kiekviena iš gautų taisyklių yra apibūdinama kaip didelė ar nedidelė disjunkcija, priklausomai nuo to koks jai priklausančių pavyzdžių kiekis, šiam skaidymui pirmiausia nustatoma riba ir lyginama ar priklausančių disjunkcijai atvejų skaičius mažesnis už ribą, ar ne.
2. Anksčiau išskirtoms mažoms disjunkcijoms vykdomas genetinis algoritmas.

Genetinį algoritmą sudaro trys pagrindiniai elementai: JEIGU sąlyga (angl. *IF*), konjunkcija, kuri susideda iš pirminių taisyklių atributų ir tinkamumo funkcija, kuri vertina taisyklės kokybę. Tuo tarpu algoritmas mažų disjunkcijų rinkinius gali formuoti dviem būdais:

1. Siejant kiekvieno medžio lapo pavyzdžius, tačiau taip gaunami rinkiniai turi po mažą kiekį pavyzdžių. Šiuo atveju algoritmą reikia paleisti du kartus, kad būtų sugrupuoti visi rinkiniai.
2. Grupuojant visus pavyzdžius, priklausančius visiems lapų mazgams į vieną rinkinį. Šiuo atveju algoritmą užtenka paleisti vieną kartą.

Taip pat svarbu išskirti, kad genetinis algoritmas, nors ir papildo taisyklių iš simuliacijos rezultatų gavimą, tačiau turi kelis trūkumus:

- Genetinis algoritmas yra lėtas, lyginant jį su sprendimų medžio metodu.

- Tradiciniai genetiniai operatoriai, tokie kaip kryžminis ir mutacijos, yra taikomi aklaui, be siūlomo galimo sprendimo kokybės optimizavimo.

2.4. K-vidurkių algoritmai

K-vidurkių (angl. *K-means*) algoritmai yra ganėtinai populiarius duomenų gavyboje dėl savo paprastumo įgyvendinti apytikslius sprendimus (Oyelade, Oladipupo ir Obagbuwa, 2010). Pagrindinės šio tipo algoritmų populiarumo priežastys: paprastumas, nesudėtingas įgyvendinimas, lengvas dydžio keitimas, greitas suliejimas ir gebėjimas prisitaikyti prie išskaidytų duomenų. K-vidurkių algoritmas remiasi grupavimo technika ir grupuoja elementus taip, kad kiekvienas elementas priklausytų tai grupei, kurios centras jam yra artimiausias. K-vidurkių algoritmo rezultatas atrodo kaip taškų sugrupavimas erdvėje, kurioje taškai išdėstomi pagal naudojamus parametrus.

K-vidurkių algoritmas susideda iš penkių etapų:

1. Numatyti grupių skaičių, į kurias bus skirstomi elementai, ir patį duomenų rinkinį.
2. Numatyti pirmąsias grupes: paimant pirmuosius pavyzdžius ar atsitiktinius elementus.
3. Apskaičiuoti kiekvienos grupės aritmetinį vidurkį.
4. Kiekvieną duomenų rinkinio elementą paskirti į grupę, kai elementai paskiriami artimiausiai grupei pagal atstumo matą, kaip pavyzdžiui Euklido atstumą.
5. Perkelti kiekvieną elementą iš naujo į grupę, kuri atrodo labiausiai panaši. Perkėlus visus elementus iš naujo apskaičiuoti kiekvienos grupės aritmetinį vidurkį.

Didžiausias k-vidurkių algoritmo trūkumas, kad grupių skaičius, kiekvienos jų centras ir atstumas turi būti žinomi iš anksto, prieš pradedant algoritmo vykdymą, o algoritmo vykdymo metu elementai jau priskiriami šioms grupėms. Taigi netinkamai pasirinktos grupės gali sąlygoti netikslius duomenų analizės rezultatus. Metodo privalumai: nesudėtingas įgyvendinimas, nes svarbiausia dalis būna numatyta iš anksto, belieka tik grupėms priskirti elementus, gebėjimas prisitaikyti prie išskaidytų duomenų ir geras pasirodymas tiek su mažais, tiek ir su dideliais duomenų kiekiais.

2.5. Neuroninių tinklų algoritmai

Priyanka Gaur neuroninius tinklus (angl. *neural network*) apibūdina kaip smegenų metaforą informacijos apdorojimui (2012). Neuroniniai tinklai laikomi labai perspektyviais dėl savo gebėjimo „mokytis“ iš duomenų, griežtų prielaidų nenaudojimo ir gebėjimo apibendrinti. Duomenų gavyboje neuroninių tinklų metodas naudojamas klasifikavimui, grupavimui, funkcijų gavimui, modelio atpažinimui ir prognozavimui. Neuroninių tinklų grafiniame atvaizdavime naudojami du pagrindiniai elementai: neuronai ir sujungimai tarp neuronų. Išskiriami pagrindiniai neuroninių tinklų privalumai:

didelis tikslumas, tolerancija triukšmui, nepriklausomumas nuo anksčiau išsikeltos prielaidos, lengvas palaikymas.

Neuroniniai tinklai yra užprogramuoti arba apmokyti:

- Saugoti, atpažinti ir pagal asociacijas gauti modelių ar duomenų bazių elementus.
- Spręsti optimizavimo problemas.
- Filtruoti triukšmą iš duomenų.
- Kontroliuoti netiksliai apibrėžtas problemas.
- Įvertinti funkcijas.

Mokslinėje literatūroje išskiriama, kad duomenų gavyba naudojant neuroninių tinklų algoritmus susideda iš keturių etapų (Singh Gill ir Ashima, 2014):

1. Modelio atpažinimo – modelis randamas pagal tai ką siekiama atlikti su duomenimis arba iš duomenų istorijos. Neuroniniai tinklai laikomi labai tiksliai atpažįstančiais modelius net jeigu duomenys yra labai triukšmingi, turi daug smarkiai nutolusių, netikslių reikšmių.
2. Klasifikavimo – atliekamas standartinis duomenų klasifikavimas į numatytąsias klases, kur duomenys klasifikuojami pagal norimas savybes.
3. Grupavimo – atliekamas standartinis duomenų skaidymas į panašių duomenų grupes.
4. Labai nutolusių reikšmių atradimas – leidžia pašalinti elementus, kurie nepritampa prie nei vienos iš grupių.

Neuroniniai tinklai gali dirbti tik su skaitiniais duomenimis, tad kitokio tipo duomenis prieš pradėdant vykdyti neuroninių tinklų algoritmus būtina konvertuoti į skaitines reikšmes, tam dažniausiai naudojamos atitikmenų lentelės, kuriose nurodoma koks skaičius atitinka kokį simbolį. Dar vienas didelis neuroninių tinklų algoritmų trūkumas – sudėtingas įgyvendinimas ir gautų rezultatų interpretavimas.

2.6. Duomenų gavybos algoritmų tipų palyginimas

Siekiant išsirinkti su kuriuo iš algoritmų tipų bus dirbama toliau, svarbu juos palyginti, tad 3 lentelėje vaizduojamas anksčiau aptartų sprendimų medžių, genetinių, mišrių sprendimų medžių / genetinių, K-vidurkių ir neuroninių tinklų duomenų gavybos algoritmų tipų palyginimas. Algoritmų tipai lyginami pirmiausia atsižvelgiant į tai kokius duomenų gavybos metodus naudoja: grupavimą, klasifikavimą, asociacijas, regresiją. Taip pat trumpai aprašomas kiekvieno algoritmo tipo veikimo principas, jo privalumai, trūkumai, bei žinoma galimybės gauti taisykles.

3 lentelė. Duomenų gavybos algoritmų tipų palyginimas

Algoritmo tipas Lyginimo kriterijus	Sprendimų medžių	Genetiniai	Mišrūs sprendimų medžių / genetiniai	K-vidurkių	Neuroninių tinklų
Grupavimas	+	+	+	+	+
Klasifikacija	+	+	+	+	+
Asociacijos	+		+		+
Regresija	+		+		+
Veikimo principas	Klasifikuojama naudojantis šakomis, kurios išeina po kelias iš vieno mazgo	Atrankos, kryžminimo ir mutacijos procesais formuojama geresnė populiacija	Didelės disjunkcijos klasifikuojamos pagal sprendimų medį, mažos – genetinį algoritmą	Grupuoja elementus taip, kad kiekvienas priklausytų grupei, kurios centras artimiausias	Visus elementus sujungia tarpusavyje pagal jų ryšius
Privalumai	Lengva suprasti, apdoroja įvairius duomenis	Nustatoma kiekvienos chromosomos svarba, jos gerumas, algoritmai ganėtinai greiti	Papildo sprendimų medžio gautas žinias smulkių taisyklių analize	Lengvas įgyvendinimas, gebėjimas prisitaikyti prie išsklaidytų duomenų	Geba mokytis iš duomenų, prognozuoti pagal duomenis, apima taisyklių visumą, ne tik kaip atskirus elementus
Trūkumai	Smarkiai sulėtėja su dideliais duomenų kiekiais	Neatsižvelgia į taisyklių visumą	Labai lėtas, genetiniai operatoriai taikomi akiai	Nežinomas reikiamų grupių kiekis, jis spėjamas	Gali dirbti tik su skaitiniais duomenimis, sunku interpretuoti
Taisyklių išgavimo galimybės	Jo pagrindinė savybė gauti taisykles, kai taisykle vadinamas kiekvienas kelias nuo šaknies iki lapo	-	Taisyklės gaunamos panašiai kaip ir naudojant sprendimų medžius	Galima išžvelgti tam tikras taisykles pagal gautas grupes	Nėra lengva interpretuoti rezultatus

2.7. Apibendrinimas ir rezultatai

Apžvelgus duomenų gavybos algoritmų tipus paaiškėjo, kad dažniausiai naudojami yra penki algoritmų tipai, tai sprendimų medžiai, genetiniai algoritmai ir iš jų sudaryti mišrūs algoritmai, K-vidurkių bei neuroninių tinklų algoritmai.

Sprendimų medis, tai grafikas, kuris vaizduoja visas galimas sąlygas ir jų rezultatus. Sprendimų medžiai gali būti klasifikuojami pagal naudojamų atributų kiekį į vienerūšius, kai vienoje sąlygoje

naudojamas tik vienas atributas ir nevienarūšius, kai naudojami keli atributai. Pagrindiniai sprendimų medžių privalumai yra nesudėtinga ir lengvai suprantama jų struktūra, galimybė naudoti įvairaus tipo duomenis bei dirbti su nedideliu duomenų kiekiu. Pagrindinis trūkumas – algoritmų vykdymo laikas tiesiogiai priklauso nuo medžio dydžio, kitaip dar vadinamo jo gyliu.

Genetiniai algoritmai iš pradinės sprendinių populiacijos formuoja naują, geresnę. Populiacijos formavimui išrenkami tėviniai elementai su kuriais vykdomas kryžminimas ir mutacija, taip gaunant naujus palikuonis. Galiausiai papildžius populiaciją atrenkamos naujos populiacijos chromosomos. Genetiniuose algoritmuose vertinamas taisyklės veiksmingumas, jis vertinamas tinkamumo funkcija ir nurodo taisyklės svarbą ir tinkamumą. Pagrindinis genetinių algoritmų tikslas yra optimizuoti turimą populiaciją į geriausią jos versiją.

Mišriuose sprendimų medžių ir genetiniuose algoritmuose naudojamos nedidelės disjunkcijos, kai rezultatas teigiamas, jei bent vieno operando reikšmė yra teisinga. Šių algoritmų metu pirmiausia vykdomas sprendimų medžio algoritmas, kai išskiriamos didelės ir mažos disjunkcijos. Sekančiame žingsnyje mažoms disjunkcijoms taikomas genetinis algoritmas, kurio metu taisyklės gaunamos iš sąlygų, kurių rezultatai yra pirminio medžio lapai. Šie algoritmai papildo taisyklių gavimą, tačiau jie yra gana lėti, o metodai taikomi aklaui, be galimo sprendimo kokybės optimizavimo.

K-vidurkių algoritmai yra nesudėtingai įgyvendinami, jie grupuoja elementus pagal tai, kurios grupės aritmetinis centras yra artimiausias. Pirmiausia sukuriamos pirminės grupės ir apskaičiuojami jų aritmetiniai vidurkiai, po to duomenys priskiriami grupėms, o vidurkiai perskaičiuojami. Didžiausias algoritmų trūkumas, kad pradžioje reikia numatyti grupes aklaui, po to gali prireikti kardinalių pakeičių.

Neuroninių tinklų algoritmai yra daug pranašesni už kitus algoritmus, nes jie gali mokytis iš duomenų, filtruoti triukšmą, įvertinti funkcijas ir kita, tačiau yra labai sudėtingi. Taip pat neuroniniai tinklai gali dirbti tik su skaitiniais duomenimis, tad būtina prieš darbą kito formato pradinius duomenis konvertuoti.

Atsižvelgiant į algoritmų tipų palyginimą tolimesniam darbui pasirinkta dirbti su sprendimų medžių algoritmais. Didžiausią įtaką šiam pasirinkimui daro didžiausias sprendimų medžių algoritmų privalumas – vizualus taisyklių išreiškimas, kuo sprendimų medžiai yra gerokai pranašesni už kitus algoritmų tipus. Nusprendus, kad bus dirbama su sprendimų medžių algoritmais galima pasirinkti ir metodą, kuriuo bus vadovaujama. Kadangi genetiniai algoritmai yra naudojami visuose methoduose, palyginus juos buvo pasirinktas grupavimo metodas, nes jame nereikia iš anksto numatyti klasių ir dirbama vienu metu tik su vienu elementu.

3. SIŪLOMAS DUOMENŲ GAVYBOS METODAS VERSLO PROCESŲ SIMULIACIJOJE

Šiame skyriuje apžvelgiamas siūlomas metodas žinioms iš verslo procesų simuliacijos išgauti pasitelkus duomenų gavybos algoritmą. Norint panaudoti duomenų gavybos algoritmus pirmiausia svarbu nustatyti su kokiais duomenų rinkiniais algoritmai gali būti vykdomi, tuomet nustatyti, kaip po simuliacijos gautą informaciją reikia konvertuoti į algoritmams tinkamus duomenis. Taigi apžvelgiamas ir duomenų paruošimas prieš duomenų gavybos taikymą. Galiausiai pateiktas pats duomenų gavybos algoritmas – CART sprendimų medžių algoritmas bei siekiamas įvesties (verslo proceso simuliacijos parametru) ir išvesties (sugrupuotų simuliacijos rezultatų, po duomenų gavybos algoritmo panaudojimo) duomenų vizualus susiejimas.

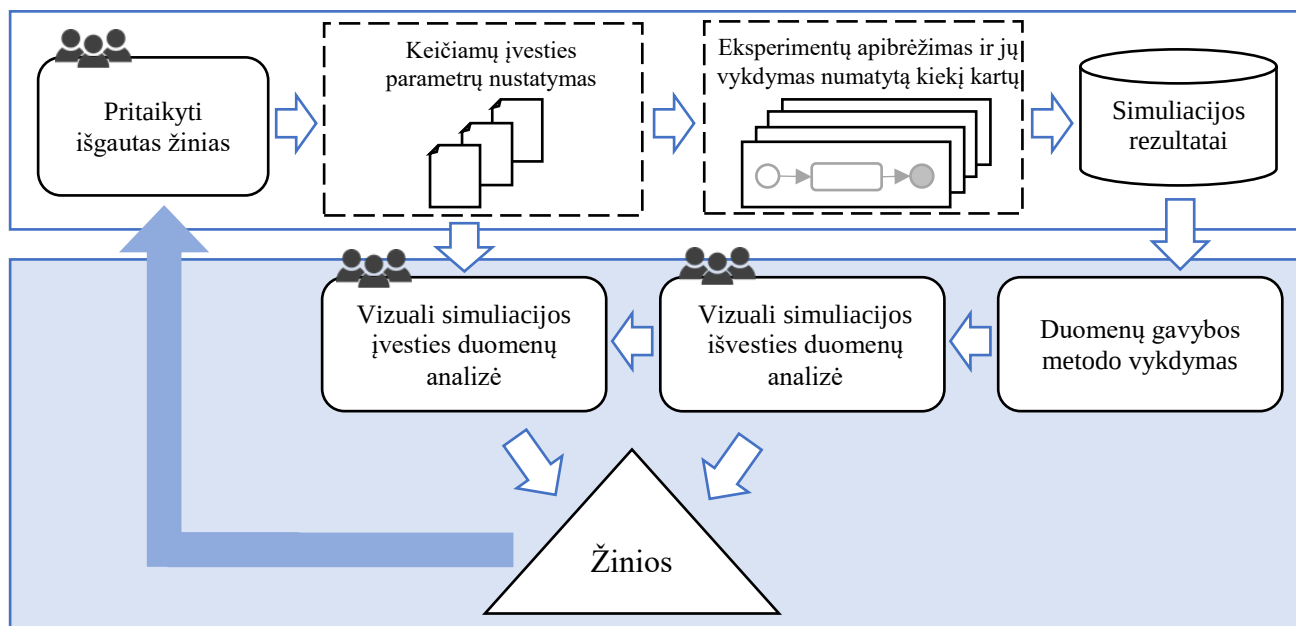
3.1. Siūlomas metodas

Niclas'as Feldkamp'as, Soeren'as Bergmann'as ir Steffen'as Strassburger'is pasiūlė analizuoti verslo procesų simuliacijos modelio elgesį, keičiant simuliacijos parametrus ir galiausiai randant ryšius tarp rezultatų ir pradinių įeigos duomenų, kurie gali atskleisti anksčiau nepastebėtas žinias (2015). Tokiame modelyje galima išskirti dvi pagrindines analizės dalis. Pirmą dalį – tai didelio simuliacijos įrašų kiekio generavimas, kai vykdamas simuliaciją keičiami jos parametrai, o antroji – simuliacijos rezultatų analizė ir žinių gavimas susiejant įvesties ir sugrupuotus išvesties duomenis. Tokio tipo analizės esminė idėja yra naudojant intuityvius vizualinius duomenų gavybos rezultatus iš dalies automatizuoti simuliacijos rezultatų analizę, taip palengvinant analitiko darbą, kai duomenų kiekiai yra per dideli, kad būtų galima analizuoti po vieną konkretų įrašą. Magistro darbe bus remiamasi šių autorių siūloma idėja, tad toliau ji aprašoma detaliau.

Simuliacijos rezultatų analizės metodas susideda iš septynių pagrindinių žingsnių:

1. Keičiamų įvesties parametru nustatymas.
2. Eksperimentų apibrėžimas ir jų vykdymas numatytą kartų kiekį, taip sukaupiant simuliacijos rezultatus.
3. Duomenų gavybos metodo vykdymas su gautais simuliacijos rezultatais.
4. Vizuali simuliacijos išvesties duomenų analizė.
5. Vizuali simuliacijos įvesties duomenų analizė, pavyzdžiui grupuojant.
6. Ryšių tarp įvesties ir išvesties duomenų nustatymas, kurie leistų interpretuoti duomenis ir interpretacijas transformuoti į žinias.
7. Pritaikyti išgautas žinias kitose simuliacijose, taip vykdamas nenutrūkstamas iteracijas, kurios leistų optimizuoti pradinį verslo procesą.

Pilnas metodo modelis pateiktas 6 paveiksle, kur baltos spalvos sritis žymi simuliacijos įrašų generavimą, o mėlyna – simuliacijos rezultatų analizę ir žinių išgavimą.



6 pav. Duomenų gavybos panaudojimo verslo procesų simuliacijoje metodas (sudaryta autoriaus remiantis Feldkamp et al., 2015)

Šiame darbe bus vykdoma tik analitinė metodo dalis: simuliacijos rezultatų analizė, panaudojant duomenų gavybos metodą, sugrupuotų duomenų vizualizacija, įeigos ir išeigos ryšių nustatymas bei bandymas iš šių ryšių gauti žinias.

3.2. Verslo procesų simuliacijos rezultatų paruošimas naudoti duomenų gavybos algoritmuose

Norint verslo procesų simuliacijos rezultatams pritaikyti duomenų gavybos algoritmus, pirmas uždavinys – apdoroti duomenis. Analizei bus naudojami diskrečių įvykių simuliacijos rezultatų duomenys, kurie aprašo kiekvieno dalyvaujančio įvykio rezultatus atskirai. Taigi vienos simuliacijos rezultatai būna sudaryti iš tiek duomenų, kiek sekoje yra įvykių ir įvesties parametrų. Pavyzdžiui, jei simuliacijoje naudojami trys parametrai ir trys įvykiai, tai įvestį sudarys trijų duomenų rinkinys, o išvestį trys rinkiniai, kur kiekvienas turės po tris duomenis, be to, visi šie rinkiniai grupuojant bus naudojami kaip atskiri objektai.

Paruošti duomenis duomenų gavybos algoritmams, išskiriami keturi žingsniai (Alasadi ir Bhaya, 2017):

1. Duomenų išvalymas (angl. *data cleaning*) – trūkstančių duomenų, triukšmo, duomenų nuokrypio ir nenuoseklumo suradimas ir pašalinimas.

2. Duomenų integracija (angl. *data integration*) – duomenų iš kelių šaltinių apjungimas į nuoseklią duomenų saugyklą.
3. Duomenų transformavimas (angl. *data transformation*) – duomenų pakeitimas į tinkamus naudoti duomenų gavybos algoritmuose.
4. Duomenų sumažinimas (angl. *data reduction*) – duomenų rinkinio sumažinimas, siekiant gauti geresnius duomenų gavybos rezultatus.

Duomenų išvalymo metu radus trūkstamą duomenų reikšmę galima elgtis dviem būdais: nenaudoti rinkinio duomenų analizėje arba užpildyti reikšmę. Reikšmės užpildymas galimas įvairiais būdais: ją įrašant rankiniu būdu (nenaudojama dideliuose duomenų rinkiniuose), užpildyti iš anksto numatyta konstanta, apskaičiuotu iš kitų rinkinių reikšmės vidurkiu ar labiausiai tikėtina reikšme. Triukšmo šalinimo metu randamos atsitiktinės klaidos ar nuokrypiai, kuriuos aptikus galima pasinaudoti trimis metodais: dėžės (angl. *binning*), regresijos arba grupavimo. Dėžės metodas mažina duomenų triukšmą koreguodamas reikšmes atsižvelgiant į jų „kaimynų“ reikšmes. Regresijos metodas taiko tiesinę regresiją, o grupavimo metodas suskirsto duomenis į grupes pagal panašumą, taip aptinkant nuokrypius, kurie netinka nei vienai grupei.

Duomenų transformavimas gali būti vykdomas lyginimo (angl. *smoothing*), agregacijos (angl. *aggregation*) apibendrinimo (angl. *generalization*) ir normalizacijos (angl. *normalization*) būdais. Lyginimo metu iš duomenų yra pašalinamas triukšmas. Agregacijos metu duomenys apibendrinami pasitelkiant statistines metrikas, kaip reikšmių vidurkius, medianas ir dispersijas, kai duomenų gavybos algoritmuose jau naudojami agreguoti duomenys. Apibendrinimo metu žemesnio lygio duomenys pakeičiami aukštesnio abstrakcijos lygio duomenimis pasinaudojus hierarchinėmis sąvokomis, pavyzdžiui tikslus amžius gali būti apibendrinamas tam tikromis grupėmis, kaip vaikai, vidutinio amžiaus asmenys ir senjorai. Normalizacijos metodas duomenų reikšmes paverčia reikšmėmis iš specifinio intervalo, pavyzdžiui nuo 0 iki 1.

Duomenų sumažinimui gali būti naudojamos penkios technikos: duomenų kubo agregacija (angl. *data cube aggregation*), atributų pogrupio pasirinkimas (angl. *attribute subset selection*), matmenų mažinimas (angl. *dimensionality reduction*), skaičiaus mažinimas (angl. *numerosity reduction*), duomenų diskretizavimas ir hierarchijos sąvokų generavimas (angl. *data discretization and concept hierarchy generation*). Naudojant duomenų kubo agregacijos techniką duomenys agreguojami į duomenų kubą, taip sumažinant duomenų kiekį, bet neatsisakant svarbios informacijos, pavyzdžiui turint atskirus duomenis apie kiekvienų metų ketvirčių pardavimus, ketvirčiai gali būti agreguoti gaunant metinius pardavimus. Atributų pogrupio pasirinkimo metu atsisakoma tam tikrų funkcijų, dimensijų ar nesvarbių atributų. Matmenų mažinimo technika naudoja duomenų kodavimą, siekiant

sumažinti duomenų rinkinio dydį, kai vėliau duomenys dekoduojami. Skaičiaus mažinimo metu duomenys pakeičiami ar suliejami su mažesniu duomenų rinkiniu. Duomenų diskretizacijos ir hierarchijos sąvokų generavimo technika naudojama pakeisti atributų duomenų reikšmes aukštesnio abstrakcijos lygio sąvokomis ar intervalais.

Verslo procesų simuliacijos rezultatai gali būti labai įvairių reikšmių, kurios priklauso nuo paties proceso ir nurodytų parametrų. Pavyzdžiui rezultatuose gali būti gaunamos tokios reikšmės: ar simuliuotas įvykis įgyvendintas sėkmingai, kiek laiko truko procesas, kiek laiko teko laukti reikiamų resursų, kokia galutinė proceso kaina ir panašiai. Duomenų gavyboje gali būti naudojamos tiek skaitinės, tiek tekstinės reikšmės. Tačiau sprendimų medžio algoritmas skaitines reikšmes tiksliau interpretuoja ir nustato išsamesnes taisykles, grąžina ne tik „taip“ arba „ne“ atsakymą, bet gali nusakyti ir intervalą.

Atsižvelgiant į analizuotus duomenų paruošimo būdus, kad su jais būtų galima naudoti sprendimų medžių grupavimą, išskirti tinkami naudoti metodai:

- Duomenų išvalymo metu radus įrašę trūkstančių reikšmių reikėtų šių įrašų tiesiog netraukti į galutinį duomenų rinkinį.
- Esant poreikiui transformuoti duomenis, kad būtų sumažintas duomenų išsisklaidymas ir klasifikuojant aiškiau išsiskirtų esminė informacija, panaudoti agregavimą arba apibendrinimą.
- Duomenų sumažinimui panaudoti atributų pogrupio pasirinkimo techniką, kad būtų atsisakyta tam tikrų nesvarbių funkcijų, dimensijų ar atributų.

3.3. Sprendimų medžio algoritmo panaudojimas simuliacijos rezultatų duomenų gavyboje

Darbe bus naudojamas CART sprendimų medžio algoritmas. CART generuoja dvejetainius medžius, kuriuose kiekvienas mazgas skaidomas į dvi šakas. CART buvo pasirinktas remiantis Sonia'os Singh ir Manoj Giri straipsnyje pateiktu populiariausių algoritmų, ID3, CART ir C4.5, palyginimu (2014). Autoriai nurodo, kad CART prieš kitus algoritmus yra pranašesnis dėl kelių savybių:

1. Gali būti naudojamas ne tik su skaitiniais kintamaisiais, bet ir kategorijomis.
2. Nustato svarbiausius kintamuosius ir pašalina nesvarbius.
3. Gali aptikti ir valdyti smarkiai nutolusias reikšmes.

Nors prieš generuojant sprendimų medį yra vykdomas duomenų apdorojimas, tačiau ne visada pavyksta išvengti netikslių duomenų grupėse. Siekiant pašalinti tokius nuokrypius ir klaidas, naudojamas sprendimų medžio genėjimas (angl. *pruning*). Genėjimas būna dviejų tipų: atliekamas po medžio kūrimo – genėjimas po, ir atliekamas medžio kūrimo metu – genėjimas realiu laiku. Genėjimas pradedamas vykdyti nuo žemiausio medžio lygio – lapų, ir tęsiamas į viršų, link medžio mazgo.

3.4. Ryšys tarp pradinių simuliacijos parametrų ir gautų simuliacijos rezultatų

Sekanti užduotis po simuliacijos rezultatų grupavimo – rasti ryšius tarp prieš simuliaciją nustatytų įvesties parametrų ir po jos gautų rezultatų. Nustačius, į kurias grupes pateko simuliacijos rezultatai su atitinkamais parametrais, būtų galima pastebėti, kurie parametrai simuliuojamame verslo procese yra jautriausiai reaguojantys į pokyčius. Pavyzdžiui, jei keičiant tam tikrą parametą simuliacijos rezultatai pasiskirsto po daugelį grupių, akivaizdu, kad šis parametras stipriai veikia visą procesą.

Ryšiui tarp parametrų ir rezultatų reikšmių nustatyti N. Feldkamp'as, S. Bergmann'as ir S. Strassburger'is (2015) siūlo naudoti stačiakampes diagramas (angl. *boxplots*). Kai kiekviena iš stačiakampių medianų rodo įvesties parametro kvartilų (atsitiktinio dydžio skirstinio skaitinių charakteristikų) poziciją vienoje grupėje. Stačiakampėms diagramoms atvaizduoti reikia nustatyti penkis taškus, kurie nusako grupės elementų reikšmes:

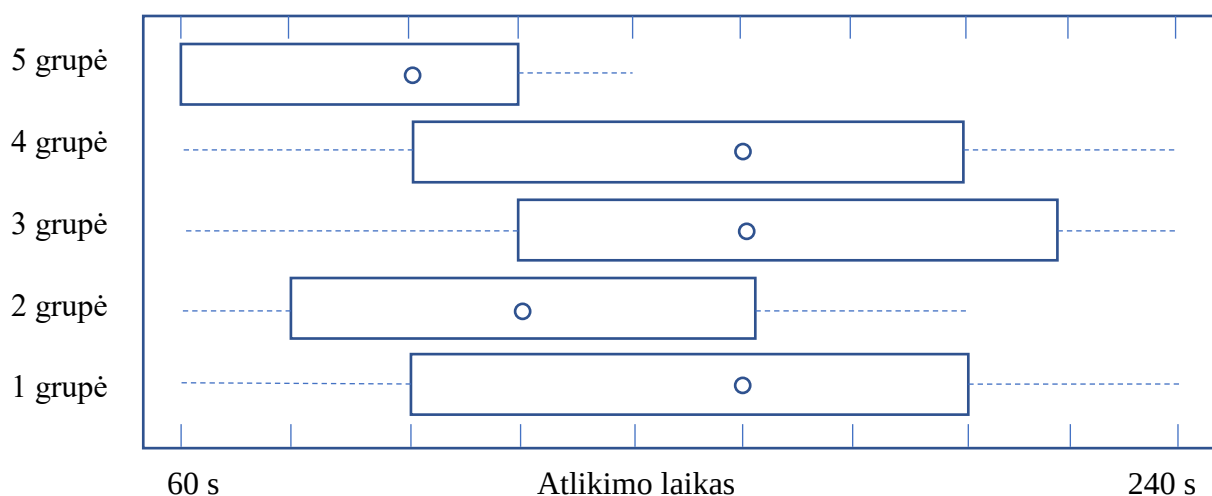
1. Mažiausią reikšmę grupėje.
2. Grupės medianą.
3. Grupės pusės nuo mažiausios vertės iki medianos medianą.
4. Grupės pusės nuo medianos iki didžiausios vertės medianą.
5. Didžiausią reikšmę grupėje.

Suradus šiuos penkis taškus duomenų rinkinys suskaidomas į keturias dalis, nuo vieno taško iki kito. Kiekvienoje stačiakampės diagramos dalyje yra po 25 % visų reikšmių. Toks reikšmių pasiskirstymo grafinis atvaizdavimas padeda vizualiai nustatyti grupių reikšmių priklausomybę nuo pasirinkto parametro.

Autoriai siūlo stačiakampių diagramų atvaizdavimą, kai grupės vaizduojamos vertikaliajoje ašyje, o parametro reikšmės horizontalioje. Pačiame stačiakampyje yra vaizduojami 50 % visų rezultatų, o stačiakampiui iš kairės ir dešinės, linijomis, pavaizduojami likę rezultatai po 25 %. Stačiakampyje dar vaizduojamas duomenų pasiskirstymo taškas, kuris žymi, kad iki taško ir už jo yra po 25 % reikšmių.

Nesunku pastebėti, kad kuo mažesnis stačiakampio ar linijų iš jo šonų plotis, tuo toje srityje duomenys yra mažiau išsisklaidę ir atvirkščiai, kuo didesnis plotas, tuo duomenų koncentracija jame

mažesnė. Jei stačiakampiai tarpusavyje visai ar beveik nepersidengia, tai rodo, kad parametras turi didelę įtaką grupių pasiskirstymui. Stačiakampių diagramų pavyzdys, kuris vaizduoja penkių grupių pasiskirstymą pagal užduočių atlikimo laiko parametą, pateiktas 7 paveiksle.



7 pav. Stačiakampių diagramų vaizdavimas, susiejant įvykdymo laiko parametą su rezultatų pasiskirstymu grupėse (sudaryta autoriaus remiantis Feldkamp et al., 2015)

Iš 7 paveikslo galima teigti, kad penktoje grupėje užduočių atlikimo laikas buvo fiksuotas intervale nuo 60 iki 120 sekundžių. Vadinasi, trumpas atlikimo laikas nusako geresnį sistemos veikimą. Taip pat, pagal stačiakampių diagramų išsidėstymą, galima pastebėti, kad į penktąją grupę patekusios reikšmės labiausiai skiriasi iš visų grupių, joje esančių reikšmių ilgiausias atlikimo laikas siekia tik 120 s, kai beveik visose kitose grupėse atlikimo laikas pasiskirsto nuo mažiausios iki didžiausios nustatytos laiko reikšmės.

Tačiau svarbiausia pastebėti, kad visų grupių stačiakampiai bent iš dalies persidengia su kitais stačiakampiais. Atsižvelgiant į stačiakampių diagramų tarpusavio persidengimą, galima teigti, kad atlikimo laikas nėra vienintelis įvesties parametras, kuris turi įtakos grupių pasiskirstymui. Taigi norint nustatyti reikšmingesnius parametrus gautiems rezultatams, reikėtų nubraižyti diagramas, kurios vaizduotų ryšius su kitais parametrais. Kuriose diagramose stačiakampiai mažiausiai persidengtų, tie parametrai būtų patys jautriausi pokyčiams.

Iš šio pradinių ir rezultatų duomenų susiejimo pateikimo būdo galima pastebėti, kad susiejus parametą su gautomis grupėmis atskleidžiama jo įtaka verslo procesui. Nustačius jautriausius parametrus, galima spręsti, kaip juos koreguoti, siekiant gauti reikiamus proceso rezultatus. Taigi tokia susiejimo idėja ir bus taikoma tolimesniame tyrime.

3.5. Apibendrinimas ir rezultatai

Aprašius siūlomą duomenų gavybos iš simuliacijos rezultatų metodą, nustatyta, kaip prieš atliekant duomenų gavybą bus apdorojami pradiniai duomenys, kaip jie bus panaudojami pačiame sprendimų medžių algoritme ir kaip vaizduojamas ryšys, tarp pradinių įvestų parametrų ir sudarytų

rezultatų grupių. Nustatyta, kad metodą sudarys keturi pagrindiniai žingsniai: duomenų gavybos metodo taikymas apdorotiems simuliacijos rezultatams, vizuali simuliacijos išvesties duomenų analizė, vizuali simuliacijos įvesties, parametrų analizė, įvesties ir išvesties ryšių nustatymas, kuris leis interpretuoti duomenis ir interpretacijas transformuoti į žinias.

Apibrėžta, kad prieš taikant duomenų gavybos metodą duomenys bus apdorojami juos išvalant, integruojant, transformuojant ir sumažinant. Duomenų išvalymas nuo nepilnų įrašų bus vykdomas atsisakant tokių įrašų, nes metodas vykdomas su dideliu duomenų kiekiu, tai apsunkina galimybę trūkstamas reikšmes užpildyti. Transformacija reikalinga sumažinti duomenų išsisklaidymui, kad grupuojant nebūtų įtraukiami smarkiai nutolę rezultatai, tam bus taikomas apibendrinimas arba agregavimas. Duomenų sumažinimas bus taikomas tam, kad būtų atsisakyta nesvarbių atributų, jam pasitelkiama atributų pogrupio pasirinkimo technika.

Išskirta, kad sprendimų medžio algoritme svarbiausi žingsniai yra informacijos naudos apskaičiavimas randant entropiją ir genėjimas, kuris padeda aptikti nuokrypius ir juos pašalinti.

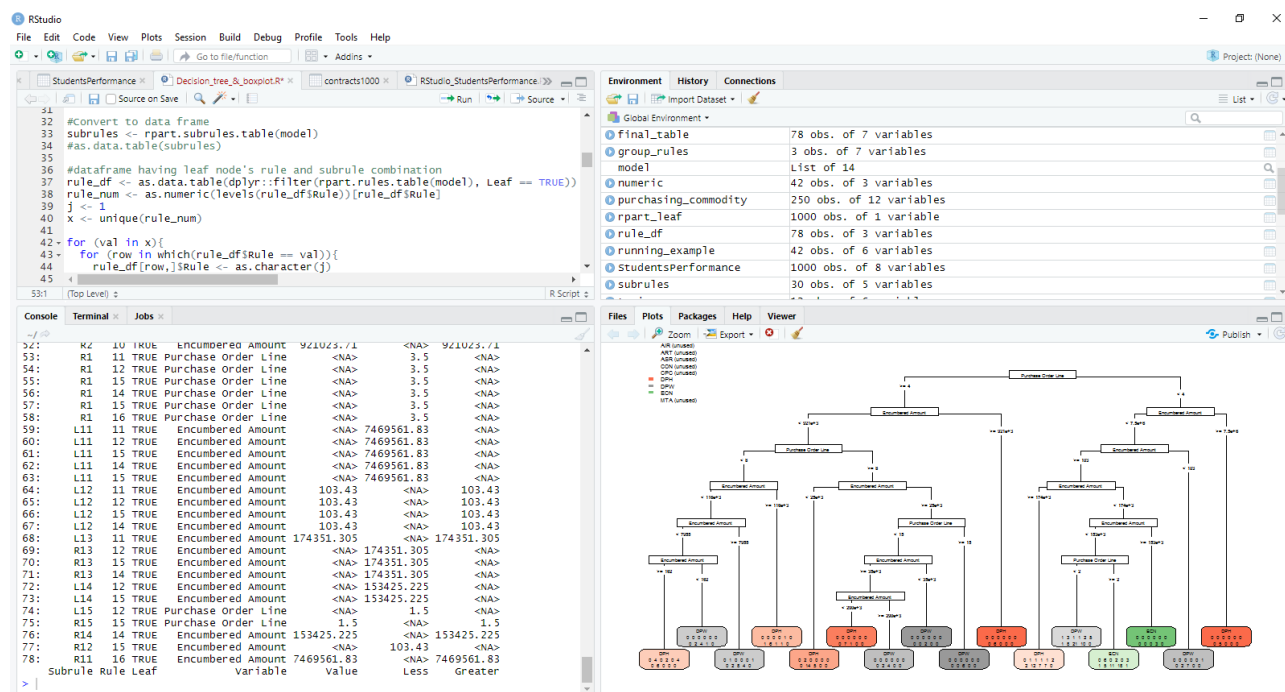
Siekiant vizualiai pateikti pradinių įvesties parametrų ryšį su gautais simuliacijos rezultatais, bus pasitelkta grafinė, po duomenų gavybos sudarytų rezultatų grupių pasiskirstymo kiekvieno parametro atžvilgiu, vizualizacija. Iš šios vizualizacijos bus galima pastebėti, kurie parametrai turi didžiausią įtaką simuliacijos rezultatų pokyčiams bei kokius parametrus nustačius gaunami geriausi rezultatai.

4. SIŪLOMO DUOMENŲ GAVYBOS METODO REALIZACIJA

Šiame skyriuje pateikiamas metodo realizacijai naudotų priemonių aprašas, metodo realizacijos aprašymas bei eksperimentinis tyrimas. Pirmiausia buvo pasirinkta metodo realizacijai atlikti tinkamiausia programinė įranga – *RStudio*. Prieš keliant duomenis į pasirinktą programinę įrangą, jie buvo apdoroti. *RStudio* standartinėmis funkcijomis iš įkeltų duomenų sugeneruotas sprendimų medis, iš sudarytų grupių gautos taisyklės. Pradinis duomenų rinkinys papildytas nauju atributu, kuriame nurodyta, kuriai grupei jis priklauso, tai išskaičiuojama pagal anksčiau gautas taisykles. Iš atnaujinto duomenų rinkinio generuojamos stačiakampių diagramos. Galiausiai, pagal atlikto eksperimento rezultatus, pateiktas galimas rezultatų interpretavimas.

4.1. Duomenų gavybai atlikti pasirinktos priemonės

Duomenų gavybai atlikti buvo pasirinkta naudoti *RStudio* programinę įrangą. *RStudio* – tai atviro kodo *R* kalbos programavimo aplinka, kuri apima *R*, *knitr*, *rmarkdown* ir žymėjimo kalbas (angl. *Markup languages*) (Gandrud, 2016). *R* – tai statistinė kalba, kuri leidžia rinkti ir analizuoti duomenis. *Knitr* ir *rmarkdown* – tai *R* kalbos paketai, kurie leidžia sujungti statistinę analizę ir rezultatų vaizdavimą. *RStudio* aplinkoje yra keli pagrindiniai komponentai: komandų rašymo langas, sintaksę išryškinantis redaktorius, įrankiai, skirti atvaizduoti grafikai. *RStudio* programinės įrangos aplinkos vaizdas pateiktas 8 paveiksle.



8 pav. *RStudio* programinės įrangos aplinka

Pasirinkta programinė įranga gali skaityti *csv*, *xls*, *xlsx*, *sav*, *dta*, *por*, *sas* ir *stata* failų formatus, taip pat yra galimybė įkelti ir *xes* formato failus. Daugiausia viešai prieinamų duomenų rinkinių pavyzdžių galima rasti *csv* ir *xes* formatų failais, tačiau bandymai parodė, kad iš *xes* failų nuskaitytų

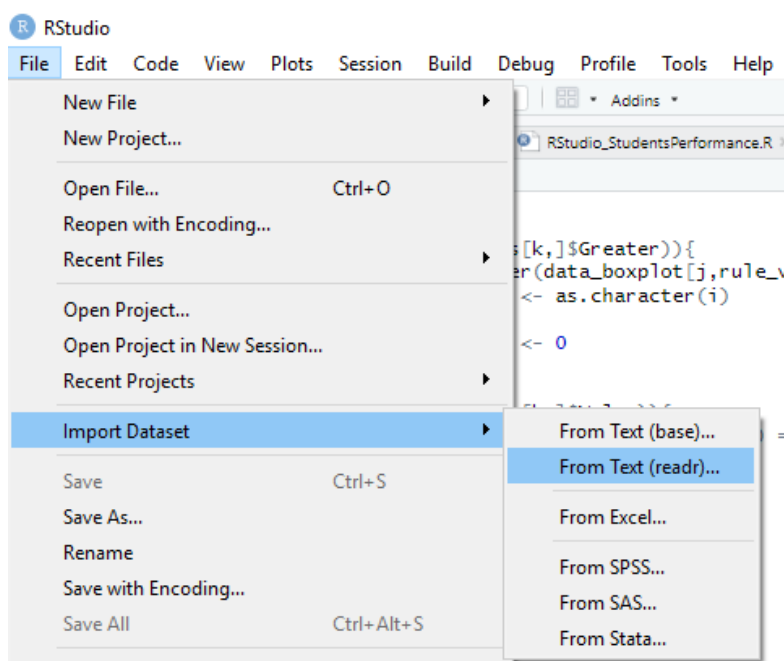
duomenų nepavyksta sugeneruoti sprendimų medžio. Taigi nuspręsta metodui išbandyti naudoti csv formato failuose saugomus duomenų rinkinius. Bet kuriame duomenų rinkinyje, kurį galėtų perskaityti *RStudio*, privalo būti naudojama tam tikra įrašų struktūra:

1. Nurodyti įrašų atributų pavadinimai, csv failuose jie nurodomi pirmoje eilutėje.
2. Atskirai įrašytos kiekvieno įrašo reikšmės, csv faile įrašai pateikiami atskirose eilutėse.

4.2. Duomenų rinkinio įkėlimas į *RStudio*

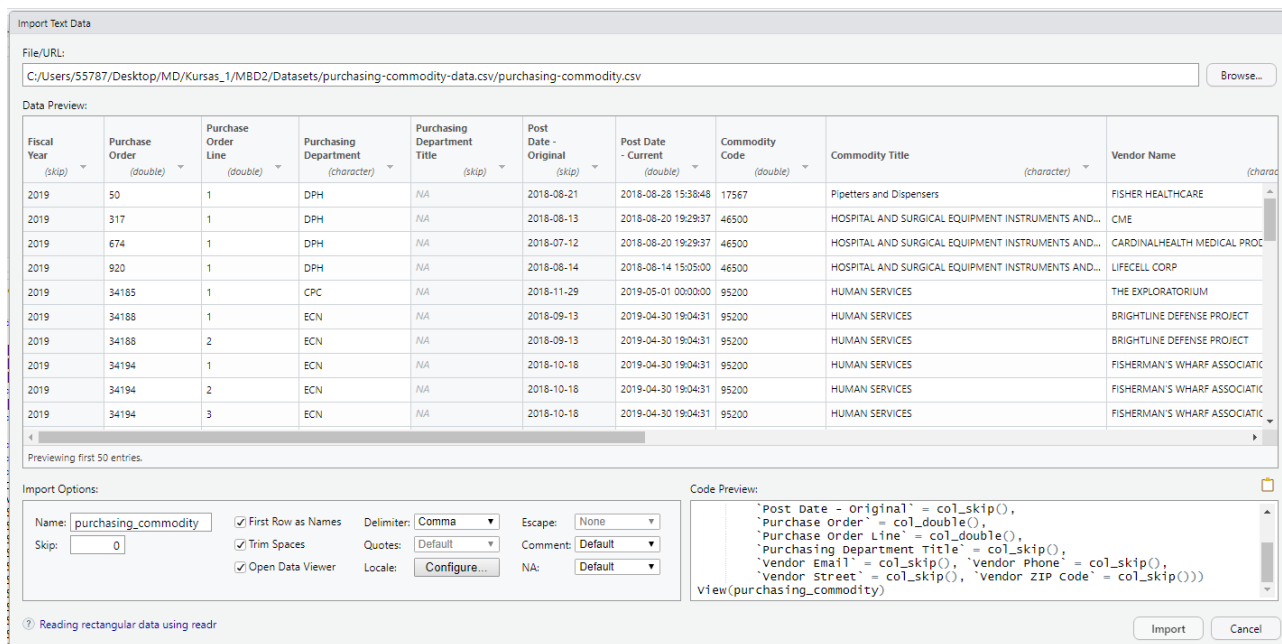
RStudio programavimo aplinkoje csv formato failus galima įkelti dviem būdais:

1. Iškvietus komanda `read.csv(failo pavadinimas)`.
2. Įkeliant failą iš meniu *File -> Import Dataset -> From text (readr)* (žr. 9 pav.). Šį būdą naudinga pasirinkti, jei norima prieš įkeliant duomenis juos peržiūrėti ir pakoreguoti.



9 pav. CSV formato failo įkėlimas *RStudio* aplinkoje

Įkeliant duomenų rinkinį būtina nurodyti koku skiriamuoju ženklu csv formato faile kiekvienas parametras yra atskiriamas, dažniausiai tai būna kablelis arba kabliataškis. Taip pat duomenų įkėlimo metu galima pakoreguoti rinkinį: pažymėti atributus, kurių nenorima įkelti arba pakoreguoti atributų tipus. Duomenų rinkinio įkėlimo lango vaizdas, *RStudio* programinėje įrangoje, pateiktas 10 paveiksle.



10 pav. Duomenų rinkinio įkėlimo lango vaizdas, RStudio programinėje įrangoje

Įkeliant failą su įrašais galima patikrinti ar visiems parametrų automatinio būdu nustatytas tinkamas tipas, jei įmanoma, naudinga kuo didesnę kiekį parametrų nustatyti į *double* – realiųjų skaičių tipą, nes metodas su šio tipo atributais veikia greičiau nei su įvairių simbolių tipo atributais, tokiais kaip *character*. Pavyzdžiui parametrai gali būti automatiškai nustatyti kaip *character* tipo, nors jų reikšmės iš tiesų yra skaitinės, kaip 001, tokį atributą nustačius skaitiniu, jis konvertuojamas į realųjį skaičių – 1.

4.3. Metodui realizuoti reikalingi paketai

Didelė dalis RStudio funkcionalumo gali būti pasiekama tik naudojantis šią programinę įrangą papildančiais paketais. Rekomenduojama naudoti kiek įmanoma mažiau paketų, taip sumažinant kodo klaidų riziką paketų atsinaujinimo atveju. Paketą galima įdiegti naudojant komandą - *install.packages(„paketo pavadinimas“)*. Įdiegti paketo nepakanka, būtina jį įkelti prieš naudojant, tai galima padaryti naudojant komandą *library(„paketo pavadinimas“)*. Paketus pakartotinai kiekvienos sesijos metu reikia įkelti tam, kad visi įdiegti paketai nebūtų užkraunami tuomet, kai jų nereikia, taip sumažinamas sistemos apkrovimas.

Įgyvendinti metodui reikalingi sprendimų medžio generavimo, taisyklių išgavimo ir grafikų vaizdavimo RStudio paketai. Iš viso buvo panaudoti šeši paketai:

1. *rpart* – sprendimų medžio generavimui, pritaikant CART algoritmą.
2. *rpart.plot* – sprendimų medžio vizualiam atvaizdavimui.
3. *rpart.utils* – taisyklių išgavimui ir jų konvertavimui į struktūrizuotas lenteles.

4. *data.table* – konvertuoti duomenų masyvą į struktūrizuotą duomenų lentelę, kuriai galima taikyti įvairias funkcijas.
5. *dplyr* – darbui su lentelių įrašais: reikšmių filtravimui pagal nurodytas sąlygas, dviejų lentelių sujungimui.
6. *ggplot2* – diagramų generavimui ir vizualiam atvaizdavimui.

4.4. Metodo realizacijos aprašymas

Pilną metodo realizaciją galima suskaidyti į šešias dalis:

1. *RStudio* darbo aplinkos paruošimas, sukeliant reikiamus tolimesniam darbui paketus ir įkeliant į ją duomenis.
2. Sprendimų medžio generavimas iš įkeltų duomenų.
3. Taisyklių gavimas, pagal suformuotą sprendimų medį.
4. Duomenų įrašų papildymas nauju atributu, kuris nurodo į kurią sprendimų medžio grupę įrašas pateko.
5. Stačiakampių diagramų generavimas.
6. Labiausiai grupių pasiskirstymą įtakojančių parametrų reikšmių ir galutinių simuliacijos rezultatų ryšio nustatymas, siekiant nustatyti naudingas simuliacijos modelio korekcijas.

Ruošiant *RStudio* darbo aplinką būtina įkelti pradinis duomenis, kuriuos būtų galima apdoroti taikant sistemos funkcijas. Prieš naudojant funkcijas, būtina įdiegti reikiamus funkcijų paketus. Funkcijų pakete yra saugomas tam tikrų funkcijų ir duomenų rinkinys, kurio neįdiegus, sistema negalės identifikuoti ir įvykdyti atitinkamų užklausų.

Sprendimų medžio kūrimui pasirinkta naudoti *rpart* funkciją, kuri naudoja CART algoritmą. Generuojant sprendimų medį galima nurodyti ar grupavimui reikia naudoti visus įrašų atributus, ar tik kelis. Taip pat būtina nurodyti, kuris atributas turi būti naudojamas medžio lapų reikšmėse, kaip taisyklės rezultatas. Sprendimų medžio algoritmas CART nustato, kurie iš nurodytų, galimų naudoti grupuojant, atributai yra svarbiausi ir grupavimui naudoja tik juos, tam, kad įrašai būtų sugrupuojami remiantis tik reikšmingais kriterijais.

Grupavimo taisyklės nustatomos trimis etapais:

1. Nustatant kiekvienos taisyklės dalies reikšmių sąrašą, kuris aprašo kiekvieną reikšmių skaidymo sąlygą.
2. Nustatant sąrašą, kokios dalys sudaro kiekvieną taisyklę.
3. Sujungiant taisyklių ir jų dalių sąrašus.

Taisyklės dalimi vadinama viena reikšmių skaidymo sąlyga, kuri nurodo, kad šią sąlygą tenkinantys įrašai patenka nuo medžio šaknies arba mazgo į kitą mazgą arba lapą. Taisyklės dalis gali būti aprašyta trijų tipų sąlygomis: reikšme, mažiau už atitinkamą reikšmę ir daugiau už atitinkamą reikšmę. Taisyklę sudarančių dalių sąrašė, taisyklėmis vadinamos šakos nuo medžio šaknies iki medžio lapo, kai sąrašė kiekvienai taisyklei išskirta tiek įrašų, kiek skirtingų dalių ją sudaro. Galiausiai abu šie sąrašai sujungiami, kai prie taisyklių sąrašo įrašų pridedama atitinkama taisyklių dalių įrašų informacija.

Duomenų įrašų papildymas nauju atributu yra ilgiausiai užtrunkanti metodo realizacijos dalis, nes tikrinamas įrašo atitikimas kiekvienai taisyklei, kol randama tinkama. Nustatyti į kurią sprendimų medžio grupę pradinis įrašas pateko, parašytas algoritmas, kuris tikrina po vieną taisyklę. Taisyklės tikrinimo metu lyginama ar įrašo atitinkamos reikšmės tenkina visas taisyklės dalių sąlygas, jeigu visos sąlygos tenkinamos, ta taisyklė įrašoma į naują įrašo atributą, jeigu ne – tikrinama kita taisyklė. Be to, kai kurios sąlygos susideda iš kelių galimų reikšmių, pavyzdžiui sprendimų medžio generavimo metu algoritmas nustato, kad į vieną grupę patenka reikšmės, kurias gali įtakoti keli asmenys. Patikrinus, kad tarp nustatytų taisyklių yra ir kombinuotų, galinčių susidėti iš kelių galimų reikšmių, jos atskiriamos nuo kitų taisyklių ir tikrinama ar įrašas atitinka bent vieną iš galimų alternatyvų. Duomenų įrašų papildymo algoritmas vykdomas tol, kol visiems įrašams nurodoma, į kurią sprendimų medžiu nustatytą grupę jie pateko.

Generuojant stačiakampių diagramas naudojamas pradinis duomenų masyvas, kuris papildytas ketvirtajame metodo žingsnyje. Diagramoms kurti naudojama *ggplot RStudio* funkcija, kur x ašyje visada nurodomas naujasis grupės atributas, o y ašyje – bet kuris kitas duomenų masyvo atributas. Patartina, kad palyginimui naudojamas atributas būtų skaitinis, tuomet stačiakampių diagrama vaizduojama intervalais, o jei naudojamas atributas turi tik kelias galimas reikšmes – stačiakampių diagrama vaizduoja tik konkrečias reikšmes.

Nustatant ryšį tarp reikšmingų parametrų ir galutinių simuliacijos rezultatų, kurie pateko į stačiakampių diagramą, nurodoma į kurią parametro galimų reikšmių grupę jie pateko, kai pirmąją grupę vadinamos reikšmės nuo mažiausios iki medianos, o antrąją – nuo medianos iki didžiausios. Priskyrus grupes sugeneruojamos diagramos, kurios nurodo įrašų galutinių reikšmių priklausomybę nuo parametro reikšmės. Diagramose vaizduojamos tikslios galutinės reikšmės esant atitinkamai parametro reikšmei bei regresinė priklausomybė. Pagal priklausomybės diagramas galima pastebėti su kokiomis parametro reikšmėmis gaunama daugiau tinkamesnių rezultatų, turint šias žinias paaiškėja, kaip reikėtų koreguoti atitinkamus parametrus simuliaciniame modelyje, norint gauti tinkamesnius simuliacijos rezultatus.

4.5. Metodą aprašantis pseudokodas

Kaip minėta ankstesniame skyriuje, metodo vykdymą galima suskaidyti į šešias dalis, nuo aplinkos paruošimo iki nustatymo, kaip parametras įtakoja rezultatus. Žemiau pateikiamas pseudokodas, kuris sukurtas pagal įgyvendintą metodo programinį kodą bei nusako kiekvienos iš metodo dalių veikimą. Komentarai žymimi grotelėmis – #.

1. *RStudio* darbo aplinkos paruošimas pateikiamas 11 paveiksle:

```
install.packages(„paketo pavadinimas“)  
library(paketo pavadinimas)  
duomenų lentelė <- failo pavadinimas
```

11 pav. Pseudokodas: *RStudio* aplinkos paruošimas

2. Sprendimų medžio generavimas ir atvaizdavimas pateikiamas 12 paveiksle:

```
sprendimų medis <- rpart(  
  `rezultatų stulpelio pavadinimas` ~ .,  
  # Norint generuoti sprendimų medį ne su visais parametrais, bet tik nurodytais reikia  
  # naudoti tokią sintaksę: `rezultatų stulpelio pavadinimas` ~ `parametras1` +  
  `parametras2`...,  
  data = duomenų lentelės kintamasis,  
  control = papildomi nustatymai medžio formavimui)  
rpart.plot(sprendimų medis)
```

12 pav. Pseudokodas: sprendimų medžio generavimas ir atvaizdavimas

3. Taisyklių gavimas pateikiamas 13 paveiksle:

```
taisyklių dalys <- rpart.subrules.table(sprendimų medis)  
taisyklės <- as.data.table(dplyr::filter(rpart.rules.table(sprendimų medis), Leaf ==  
  TRUE))  
kiekvienam i, kai i = unikalios taisyklės vykdyti  
  priskirti taisyklei numerį nuo 1 iki unikalių taisyklių kiekio  
baigti  
taisyklės ir jų dalys <- merge(x = taisyklės, y = taisyklių dalys, by = "Subrule", all.x =  
  TRUE, allow.cartesian = TRUE)
```

13 pav. Pseudokodas: taisyklių gavimas

4. Duomenų įrašų papildymas grupę nusakančiu atributu pateiktas 14 ir 15 paveiksluose:

kiekvienam j , kai $1 < j < \text{sprendimų medis įrašų kiekį}$ **vykdyti**

kiekvienam i , kai $1 < j < \text{unikalių taisyklių kiekį}$ **vykdyti**

taisyklės dalys $\leftarrow \text{dplyr::filter}(\text{taisyklės ir jų dalys}, \text{Rule} == i)$

unikalios taisyklės dalys $\leftarrow \text{unique}(\text{taisyklės dalys}\$Subrule)$

jei unikalios taisyklės dalys kiekis $< \text{taisyklės dalys kiekis}$ **vykdyti**

kiekvienam k , kai $k = \text{unikalios taisyklės dalys}$ **vykdyti**

jei taisyklių dalių kiekis, kurios atitinka tą pačią unikalią sąlygą > 1

vykdyti

taisyklės dalys $\leftarrow \text{taisyklės dalys} [-c(\text{neunikalios taisyklės}),]$

baigti

baigti

tą pačią sąlygą atitinkančios kelios taisyklės dalys $\leftarrow \text{dplyr::anti_join}(\text{visos taisyklės dalys}, \text{tik unikalios taisyklės dalys}, \text{by} = \text{"Subrule"})$

baigti

jei tą pačią sąlygą atitinkančios kelios taisyklės dalys turi įrašų **vykdyti**

kiekvienam m , kai $m = \text{unikalios tą pačią sąlygą atitinkančios kelios taisyklės dalys}$ **vykdyti**

kiekvienam l , kai $l = \text{tą pačią sąlygą atitinkančios kelios taisyklės dalys}$ pagal sąlygą **vykdyti**

jei įrašas atitinka bent vieną iš sąlygų

sprendimų medis\$klasė $\leftarrow i$

priešingu atveju pereiti prie kitos taisyklių grupės

baigti

baigti

baigti

jei tik unikalios taisyklės dalys turi įrašų **vykdyti**

kiekvienam k , kai $1 < k < \text{tik unikalios taisyklės dalys}$ **vykdyti**

jei egzistuoja tik unikalios taisyklės dalys $[k,]\$Less$ **vykdyti**

jei įrašas $< \text{tik unikalios taisyklės dalys } [k,]\$Less$

sprendimų medis\$klasė $\leftarrow i$

priešingu atveju pereiti prie kitos taisyklių grupės

baigti

14 pav. Pseudokodas: duomenų įrašų papildymas grupę nusakančiu atributu I dalis

```

priešingu atveju jei egzistuoja tik unikalios taisyklės dalys [k,]$Greater
vykdyti
    jei įrašas > tik unikalios taisyklės dalys [k,]$Greater
        sprendimų medis$klasė <- i
    priešingu atveju pereiti prie kitos taisyklių grupės
baigti
priešingu atveju jei egzistuoja tik unikalios taisyklės dalys [k,]$Value
vykdyti
    jei įrašas == tik unikalios taisyklės dalys [k,]$Value
        sprendimų medis$klasė <- i
    priešingu atveju pereiti prie kitos taisyklių grupės
baigti
priešingu atveju
    sprendimų medis$klasė <- 0
baigti
baigti
baigti
baigti

```

15 pav. Pseudokodas: duomenų įrašų papildymas grupę nusakančiu atributu II dalis

5. Stačiakampių diagramų generavimas pateikiamas 16 paveiksle:

```

stačiakampių diagrama <- ggplot(sprendimų medis, aes(x = class, y = `parametras`)) +
geom_boxplot(outlier.shape = NA)
intervalas = mažiausia reikšmė stačiakampių diagramoje, didžiausia reikšmė stačiakampių
diagramoje
stačiakampių diagrama + coord_cartesian(ylim = rėžiai)

```

16 pav. Pseudokodas: stačiakampių diagramų generavimas

6. Labiausiai grupių pasiskirstymą įtakančių parametrų reikšmių ir galutinių simuliacijos rezultatų ryšio nustatymas bei vizualus vaizdavimas pateikiamas 17 paveiksle:

```
pusė <- median(intervalas)
```

kiekvienam i , kai $1 < i < \text{sprendimų medis}$ įrašų kiekį vykdyti

jei sprendimų medis $[i, \text{parametras}] \geq \text{intervalas}[\text{mažiausia reikšmė}]$ ir sprendimų medis $[i, \text{parametras}] < \text{pusė}$

```
sprendimų medis[i,]$pusė <- "1"
```

baigti

priešingu atveju jei sprendimų medis $[i, \text{parametras}] \leq \text{intervalas}[\text{didžiausia reikšmė}]$ ir sprendimų medis $[i, \text{parametras}] \geq \text{pusė}$

```
sprendimų medis[i,]$pusė <- "2"
```

baigti

baigti

```
ggplot(data=sprendimų medis, aes(x = `parametras`, y = `rezultatas`, color = pusė)) +  
geom_line() + geom_point() + geom_smooth(method = "lm")
```

17 pav. Pseudokodas: labiausiai grupių pasiskirstymą įtakančių parametru reikšmių ir galutinių simuliacijos rezultatų ryšio nustatymas

Pseudokode pasvirusiu šriftu vaizduojamos funkcijos, jų nustatymai, kintamųjų pavadinimai ar kiti programiniame kode būtini elementai, kurie negali būti keičiami.

4.6. Apibendrinimas ir rezultatai

Pagal anksčiau aprašytą siūlomą metodą buvo pasirinktos jam įgyvendinti tinkamos priemonės, programinė įranga – *Rstudio*. Aprašytos duomenų failo įkėlimo į programinės įrangos aplinką galimybės. Trumpai pristatyti naudojami funkcijų paketai, kokios jų galimybės naudojamos metodei įgyvendinti. Pateikiamas išsamus realizacijos aprašymas, kuriame išskirti šeši etapai: darbo aplinkos paruošimas, sprendimų medžio generavimas, taisyklių gavimas, duomenų įrašų papildymas nauju atributu, kuris nurodo į kurią sprendimų medžio grupę įrašas pateko, stačiakampių diagramų generavimas ir labiausiai grupių pasiskirstymą įtakančių parametru reikšmių ir galutinių simuliacijos rezultatų ryšio nustatymas. Taip pat aprašomas programos pseudokodas kiekvienam iš realizacijos etapų.

5. Eksperimentiniai tyrimai

Eksperimentiniai tyrimai buvo atliekami su įvairios struktūros pradiniais duomenimis, siekiant išsiaiškinti ar pasiūlytas metodas tinka taisyklių nustatymui iš duomenų rinkinių, kurie imituoja verslo procesų simuliacijos metu gautus duomenis. Atliekant tyrimus buvo siekiama parodyti realizuoto metodo galimybes bei nustatyti tobulintinas sritis. Eksperimentai buvo atliekami su palyginti nedideliais duomenų kiekiais, tokie kiekiai buvo pasirinkti dėl naudotos techninės įrangos ribotų galimybių. Darbe pateikiami du iš atliktų eksperimentų, kurie labiausiai atspindi galimus galutinių rezultatų skirtumus ir įmanomas nustatytų taisyklių bei ryšių interpretacijas. Eksperimentų atlikimo metu naudoti duomenys paimti iš viešai skelbiamų duomenų rinkinių *kaggle.com* tinklalapyje. Remiantis žiniomis, gautomis atlikus pirmuosius du eksperimentus, atliktas trečiasis eksperimentas, kuriame naudojami sukurto verslo proceso simuliacijos modelio rezultatai.

5.1. I eksperimentas

I eksperimentas buvo atliekamas su duomenų rinkiniu, kurį sudarė skaitinės ir tekstinės reikšmės. Siekta nustatyti kaip metodas veikia su įvairaus tipo duomenimis. Pirmieji bandymai buvo labai nesėkmingi, nes naudoti duomenys buvo pernelyg skirtingi, todėl algoritmas sukrovosi keliasdešimt minučių kol galiausiai programinė įranga „užstrigdavo“. Po nesėkmingų bandymų duomenų imtis buvo sumažinta, o įrašai pasirenkami pagal bendras savybes.

5.1.1. Naudojamas duomenų rinkinys

I eksperimentui pasirinktame duomenų rinkinyje pateikta San Fransisko miesto prekių ir paslaugų pirkimo procese numatytų parametrų informacija nuo 2007 iki 2019 fiskalinių metų. Duomenų rinkinys paimtas iš *kaggle.com*, originalus jo pavadinimas – *SF Purchasing Commodity Data*.

Rinkinyje iš viso buvo virš dviejų milijonų įrašų, tačiau siekiant optimalaus skaičiavimų greičio iš jų buvo paimti tik 2 000 įrašų. Duomenų kiekis sumažintas išrinkus tik penkias prekių kategorijas ir pašalinus įrašus su neteisingai nurodytomis pirkimo kainomis (neigiamomis bei lygiomis nuliui), taip pat atsisakius įrašų su pačiomis didžiausiomis kainomis, nes jie pernelyg nutolę nuo kituose įrašuose nurodytų sumų. Naudoto duomenų rinkinio failo antraštės eilutė ir penki pirmieji įrašai pateikti 4 lentelėje.

4 lentelė. I eksperimente naudoto duomenų rinkinio antraštė ir 5 pirmieji įrašai

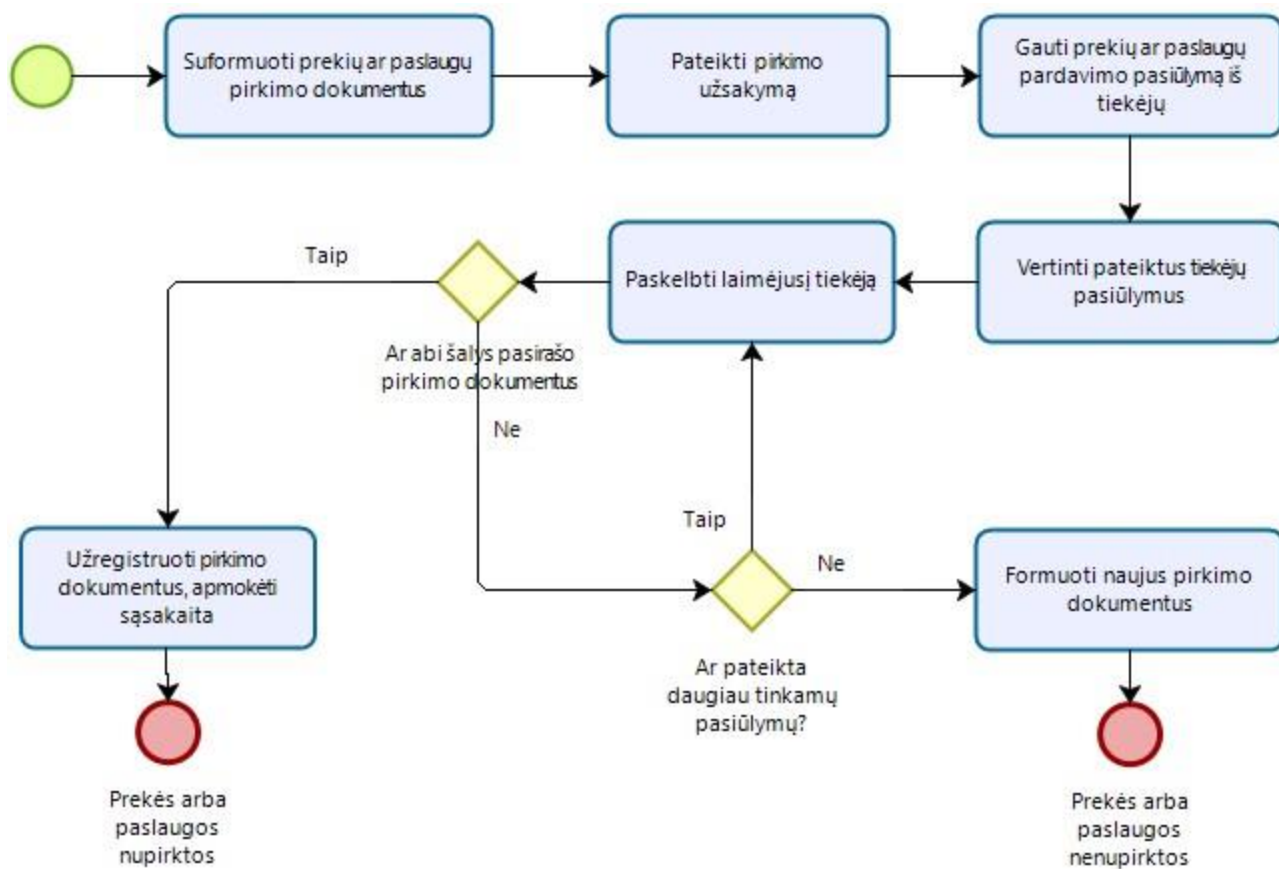
Purchase Order	Purchase Order Line	Purchasing Department	Post Date - Current	Commodity Code	Vendor State	Encumbered Quantity	Encumbered Amount
202844	1	DPH	2019-05-03T19:10:59.000	3100	CA	0	31920
306897	1	PUC	2019-05-03T19:10:59.000	3778	CA	1	3892.98
306557	1	HSA	2019-05-03T13:10:54.000	3100	CA	0	325.5
291333	1	DPW	2019-05-03T00:00:00.000	3100	CA	1	135.22
291333	2	DPW	2019-05-03T00:00:00.000	3100	CA	3	145.79

Įrašus aprašo iš viso 19 atributų, tačiau šis kiekis sumažintas iki 7 naudingiausių metodui parametų ir 1 rezultatų atributo. Elementų kiekį sumažinti galima ne tik pačiame faile, bet ir įkeliant duomenų rinkinį galima pasirenkti, kurių parametų nereikia įkelti. Galutiniame duomenų rinkinyje pasirinkti naudoti šie parametrai:

1. *Purchase order* – pirkimo dokumento numeris, kuris susideda iš 10 skaitmenų, pvz.: 0000000050.
2. *Purchase order line* – pirkimo užsakymo eilutė, kuri nurodo perkamų paslaugų ar prekių tipą, ji susideda iš 2 skaitmenų, pvz.: 01.
3. *Purchasing Department* – pirkimą vykdomasis departamentas, kuris nurodo organizacinį San Fransisko miesto ir apskrities vienetą, žymimas trumpiu, pvz.: DPH.
4. *Post Date – Current* – data, kada pirkimo dokumentas buvo išsiųstas apskaitos sistemai, nurodomi metai, mėnuo, diena, valandos minutės, sekundės ir milisekundės pvz.: 2018-08-28T15:38:48.000.
5. *Commodity Code* – prekės kodas, kuris susideda iš 5 skaitmenų, pvz.: 17567.
6. *Vendor State* – organizacijos valstija, iš kurios perkamos prekės ar paslaugos, rašoma santrumpa, pvz.: MA.
7. *Encumbered Quantity* – perkamų prekių ar paslaugų kiekis, kai nediskrečioms prekėms jis nurodomas kaip 0, rašomas skaičiais, pvz.: 0.

Rezultatas pateikiamas atribute *Encumbered Amount*, kuris nurodo prekių ar paslaugų sumą doleriais, pavyzdžiui: 11193.3.

Verslo proceso modelis pateiktas 18 paveiksle.

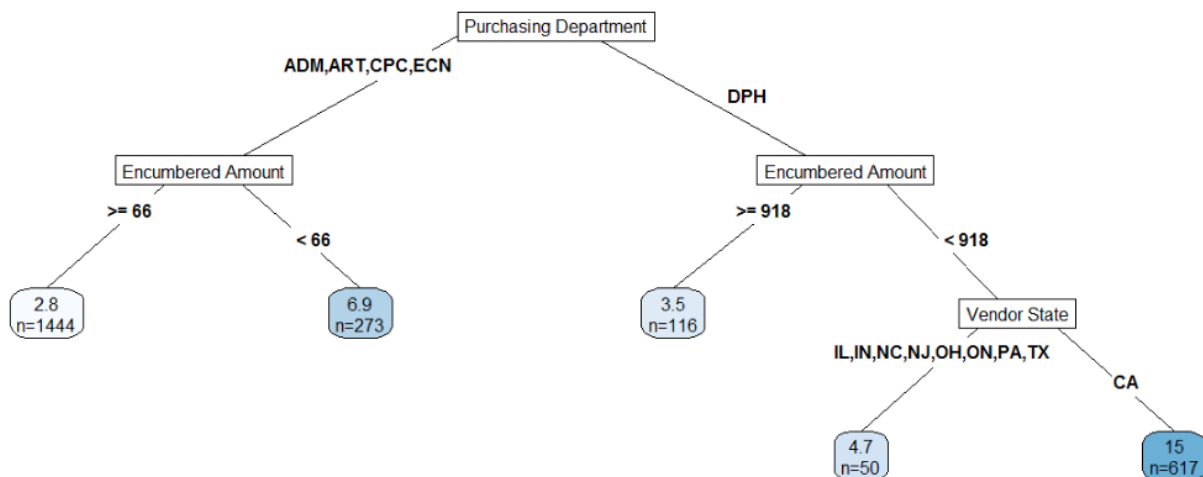


18 pav. I eksperimento verslo proceso modelis

Žingsnyje „Suformuoti prekių ir paslaugų pirkimo dokumentus“ nurodomi parametrai: *Purchase order line*, *Purchasing Department*, *Commodity Code*, *Encumbered Quantity*. Žingsnyje „Užregistruoti pirkimo dokumentus, apmokėti sąskaitą“ nurodomi parametrai: *Purchase order*, *Post Date – Current*, *Vendor State*. Simuliacijos rezultatas įrašytas *Encumbered Amount*. Duomenų rinkinys formuojamas tik iš tų įrašų, kurie pasiekė sėkmingą pabaigą, kai prekės arba paslaugos buvo nupirktos.

5.1.2. Eksperimento eiga ir rezultatai

Prieš pradedant eksperimentą sukeliama reikiama paketai ir nurodomas duomenų rinkinys, kuris toliau bus naudojamas. Pirmiausia sugeneruojamas sprendimų medis, kuris šio eksperimento metu formuojamas ne iš visų rinkinyje esančių atributų. Medžio lapams, kurie interpretuojami kaip rezultatų grupės, pasirenkamas atributas *Purchase Order Line*. Skaidymo sąlygoms, kurios interpretuojamos kaip taisyklių sudedamosios dalys, pasirinkti atributai: *Purchasing Department*, *Encumbered Amount* ir *Vendor State*. Sugeneruotas sprendimų medis pateiktas 19 paveiksle.



19 pav. I eksperimento sprendimų medis

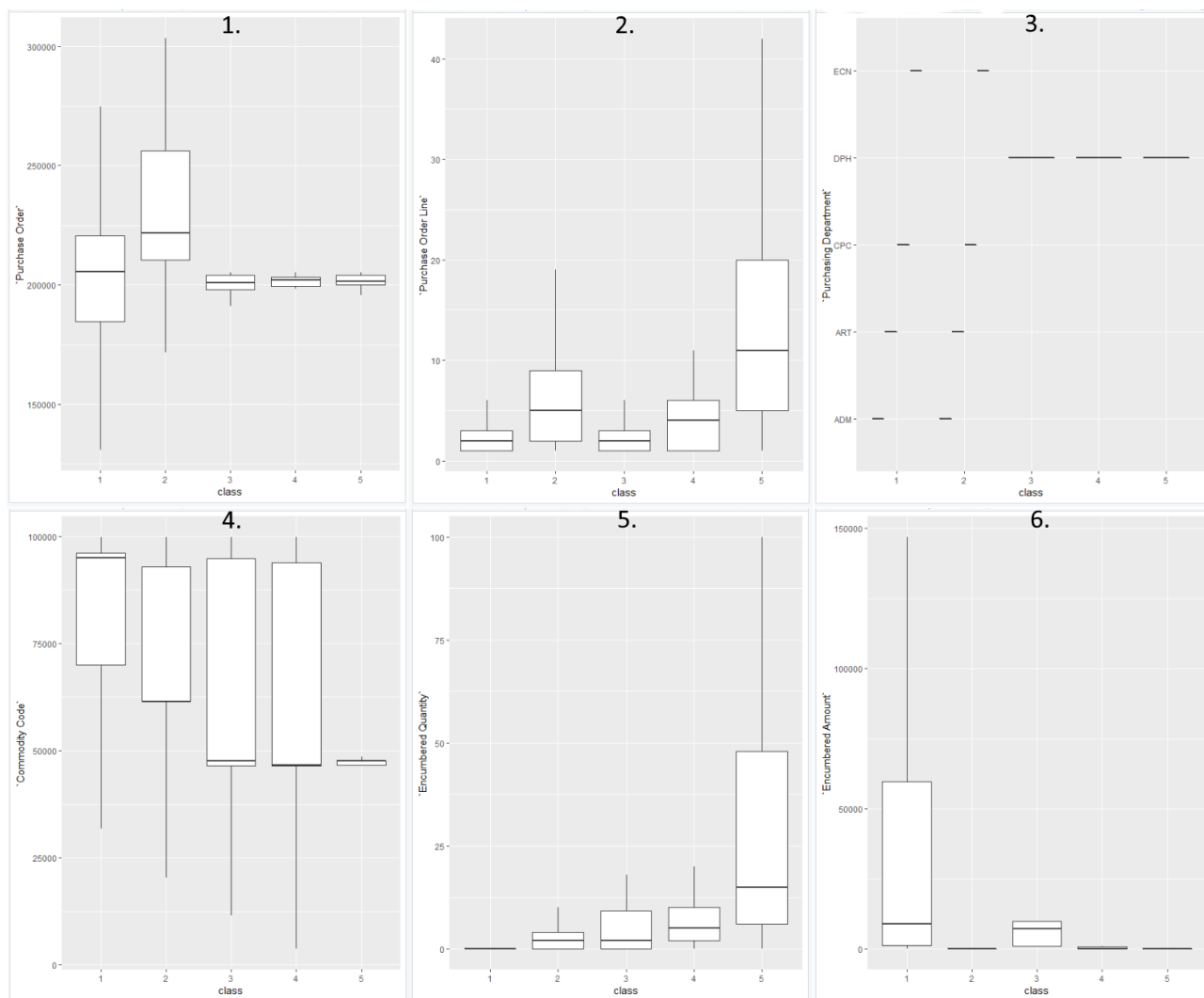
Pagal sugeneruotą sprendimų medį pirma nustatomos taisyklių dalys, medžio mazguose panaudotos skaidymo sąlygos, sąrašą iš viso sudaro 18 įrašų. Po to suformuojamas taisyklės sudarančių taisyklių dalių sąrašas, nustatyta 12 įrašų. Galiausiai šie sąrašai sujungiami į vieną, kurio bendras įrašų kiekis – 25. Visi taisyklių gavimo etapai vaizduojami 20 paveiksle.

Subrule	Variable	Value	Less	Greater	Rule	Subrule	Leaf
1:	L1 Purchasing Department	ADM	<NA>	<NA>	1:	1	L1 TRUE
2:	L1 Purchasing Department	ART	<NA>	<NA>	2:	1	L2 TRUE
3:	L1 Purchasing Department	CPC	<NA>	<NA>	3:	2	L1 TRUE
4:	L1 Purchasing Department	ECN	<NA>	<NA>	4:	2	R2 TRUE
5:	L2 Encumbered Amount	65.99	<NA>	65.99	5:	3	R1 TRUE
6:	L3 Encumbered Amount	917.75	<NA>	917.75	6:	3	L3 TRUE
7:	L4 Vendor State	IL	<NA>	<NA>	7:	4	R1 TRUE
8:	L4 Vendor State	IN	<NA>	<NA>	8:	4	R3 TRUE
9:	L4 Vendor State	NC	<NA>	<NA>	9:	4	L4 TRUE
10:	L4 Vendor State	NJ	<NA>	<NA>	10:	5	R1 TRUE
11:	L4 Vendor State	OH	<NA>	<NA>	11:	5	R3 TRUE
12:	L4 Vendor State	ON	<NA>	<NA>	12:	5	R4 TRUE
13:	L4 Vendor State	PA	<NA>	<NA>			
14:	L4 Vendor State	TX	<NA>	<NA>			
15:	R1 Purchasing Department	DPH	<NA>	<NA>			
16:	R2 Encumbered Amount	<NA>	65.99	<NA>			
17:	R3 Encumbered Amount	<NA>	917.75	<NA>			
18:	R4 Vendor State	CA	<NA>	<NA>			

Subrule	Rule	Leaf	Variable	Value	Less	Greater
1:	L1	1 TRUE	Purchasing Department	ADM	<NA>	<NA>
2:	L1	1 TRUE	Purchasing Department	ART	<NA>	<NA>
3:	L1	1 TRUE	Purchasing Department	CPC	<NA>	<NA>
4:	L1	1 TRUE	Purchasing Department	ECN	<NA>	<NA>
5:	L1	2 TRUE	Purchasing Department	ADM	<NA>	<NA>
6:	L1	2 TRUE	Purchasing Department	ART	<NA>	<NA>
7:	L1	2 TRUE	Purchasing Department	CPC	<NA>	<NA>
8:	L1	2 TRUE	Purchasing Department	ECN	<NA>	<NA>
9:	L2	1 TRUE	Encumbered Amount	65.99	<NA>	65.99
10:	R2	2 TRUE	Encumbered Amount	<NA>	65.99	<NA>
11:	R1	3 TRUE	Purchasing Department	DPH	<NA>	<NA>
12:	R1	4 TRUE	Purchasing Department	DPH	<NA>	<NA>
13:	R1	5 TRUE	Purchasing Department	DPH	<NA>	<NA>
14:	L3	3 TRUE	Encumbered Amount	917.75	<NA>	917.75
15:	R3	4 TRUE	Encumbered Amount	<NA>	917.75	<NA>
16:	R3	5 TRUE	Encumbered Amount	<NA>	917.75	<NA>
17:	L4	4 TRUE	Vendor State	IL	<NA>	<NA>
18:	L4	4 TRUE	Vendor State	IN	<NA>	<NA>
19:	L4	4 TRUE	Vendor State	NC	<NA>	<NA>
20:	L4	4 TRUE	Vendor State	NJ	<NA>	<NA>
21:	L4	4 TRUE	Vendor State	OH	<NA>	<NA>
22:	L4	4 TRUE	Vendor State	ON	<NA>	<NA>
23:	L4	4 TRUE	Vendor State	PA	<NA>	<NA>
24:	L4	4 TRUE	Vendor State	TX	<NA>	<NA>
25:	R4	5 TRUE	Vendor State	CA	<NA>	<NA>

20 pav. I eksperimento taisyklių gavimo etapai

Toliau kiekvienas pradinį duomenų įrašas papildomas nauju atributu, kuris nurodo į kurią grupę šis įrašas patenka, o tikrinimas vykdomas pagal kiekvieną iš grupę sudarančių taisyklių. Priskyrus grupės atributą prie įrašų galima formuoti stačiakampių diagramas. Generuojant diagramas pašalinami smarkiai nutolę duomenys, kurie nepatenka į stačiakampę diagramą, taip gaunant tik aktualų vaizdą. Šiuo atveju diagramas naudinga generuoti tik su skaitiniais parametrais, nes kitu atveju bus vaizduojami tik pavieniai taškai. Stačiakampių diagramų atvaizdavimas pateiktas 21 paveiksle.



21 pav. I eksperimento stačiakampių diagramų atvaizdavimas

Kiekviena iš stačiakampių diagramų suteikia tam tikrų žinių, kurias vėliau galima bandyti pritaikyti verslo proceso simuliacijoje, pavyzdžiui:

1. Pirmoji diagrama vaizduoja įrašų pasiskirstymą atsižvelgiant į atributą *Purchase order*. Galima pastebėti, kad daugiausia įrašų susikcentravę ties užsakymo numeriu apie 200 000, tiesa, nuo šio atributo duomenys labai silpnai priklausomi, nes visi stačiakampiai persidengia.
2. Antroji diagrama vaizduoja pasiskirstymą pagal *Purchase order Line*. Pagrindinės žinios, kurias galima gauti iš šios diagramos, kad didžioji dalis įrašų yra žemiau 20 pirkimo užsakymo eilutės, tad reikėtų koncentruotis į juos.
3. Trečioji diagrama vaizduoja jau minėtą konkrečių taškų parodymą, jei priklausomybei nurodyti pasirinktas ne skaitinis atributas.
4. Ketvirtoji diagrama pateikia ryšį su *Commodity Code*. Persidengiantis stačiakampių vaizdas rodo, kad nuo prekės kodo nepriklauso kokios taisyklės įrašams yra taikomos.
5. Penktojoje diagramoje vaizduojamas įrašų pasiskirstymas pagal *Encumbered Quantity*. Visų parametro reikšmių įrašai patenka į penktąją grupę, kitose grupėse duomenų pasiskirstymas

nepalyginamai mažesnis, tad būtų naudinga daugiau pasigilinti į penktosios grupės taisyklės ir sužinoti kuo ši grupė skiriasi nuo kitų. Taip pat iš diagramos galima pastebėti, kad didžioji dalis visų įrašų yra žemiau parametro reikšmės 25.

6. Šešta diagrama vaizduoja įrašų pasiskirstymą pagal *Encumbered Amount*. Matoma, kad šioje pačios brangiausios prekės pateko į pirmą ir trečią grupes, tačiau pats duomenų pasiskirstymas nuo parametro nelabai priklauso.

Apibendrinant galima teigti, kad nei vienas parametras nedaro didelės įtakos duomenų pasiskirstymui, taigi šiam duomenų rinkiniui nebūtų lengva nustatyti kokios korekcijos padėtų patobulinti verslo proceso simuliaciją. Tokioje situacijoje būtų naudinga duomenis papildomai apdoroti, pašalinant didelius duomenų nuokrypius, kurie gali trukdyti gauti naudingus rezultatus, ir pritaikyti metodą pakartotinai.

5.2. II eksperimentas

II eksperimentas buvo atliekamas su duomenų rinkiniu, kuriame visos reikšmės skaitinės. Metodus su skaitinėmis reikšmėmis nuo pirmojo bandymo buvo vykdomas sėkmingai, reikšmių apdorojimas smarkiai greitesnis, lyginant su I eksperimentu. Be to, pavyko eksperimentą įgyvendinti algoritmui paliekant analizuoti visus parametrus.

5.2.1. Naudojamas duomenų rinkinys

Atlikti antrajam tyrimui pasirinktas nesunkiai suprantamas ir smarkiai apdorotas pradinių duomenų rinkinys, kad būtų galima pademonstruoti metodo galimybes. Duomenų rinkinys, kaip ir pirmasis, paimtas iš *kaggle.com*, originalus jo pavadinimas – *Graduate Admissions*. Rinkinyje iš viso yra 500 įrašų. Duomenys aprašo absolventų iš Indijos priėmimo į magistrantūros studijų programas tikimybę, priklausomai nuo kelių parametrų. Naudoto duomenų rinkinio failo antraštės eilutė ir penki pirmieji įrašai pateikti 5 lentelėje.

5 lentelė. II eksperimente naudoto duomenų rinkinio antraštė ir 5 pirmieji įrašai

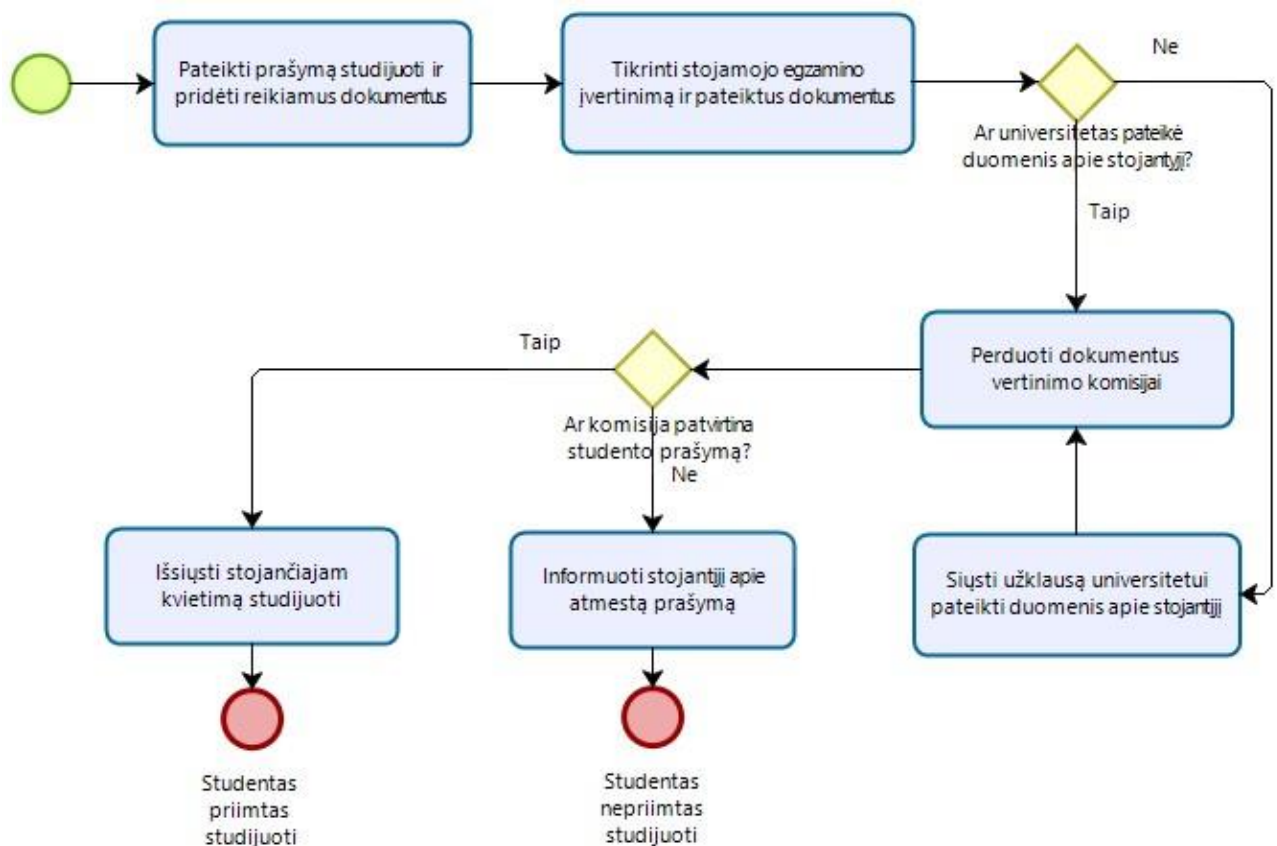
GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
337	118	4	4.5	4.5	9.65	1	0.92
324	107	4	4	4.5	8.87	1	0.76
316	104	3	3	3.5	8	1	0.72
322	110	3	3.5	2.5	8.67	1	0.8
314	103	2	2	3	8.21	0	0.65

Įrašus aprašo devyni atributai, iš jų atsisakyta tik vieno, kuris žymi eilės numerį. Visi įrašus aprašantys parametrai yra skaitiniai, o tuščių reikšmių nėra, tad duomenų apdoroti nereikėjo. Galutiniame duomenų rinkinyje naudojami parametrai:

1. *GRE Score* – stojamojo egzamino įvertinimas, iki 340.
2. *TOEFL Score* – TOEFL anglų kalbos testo įvertinimas, iki 120.
3. *University Rating* – universiteto reitingas, iš 5 balų.
4. *SOP* – studento motyvacinio laiško stiprumas, iš 5 balų.
5. *LOR* – rekomendacinio laiško stiprumas, iš 5 balų.
6. *CGPA* – bakalauro studijų įvertinimų vidurkis, iki 10.
7. *Research* – darbo patirtis su moksliniais tyrimais, 0 arba 1.

Gaunamas rezultatas pateikiamas atribute *Chance of Admit*, kuris rodo priėmimo į studijas tikimybę, nuo 0 iki 1.

Verslo proceso modelis pateiktas 22 paveiksle.



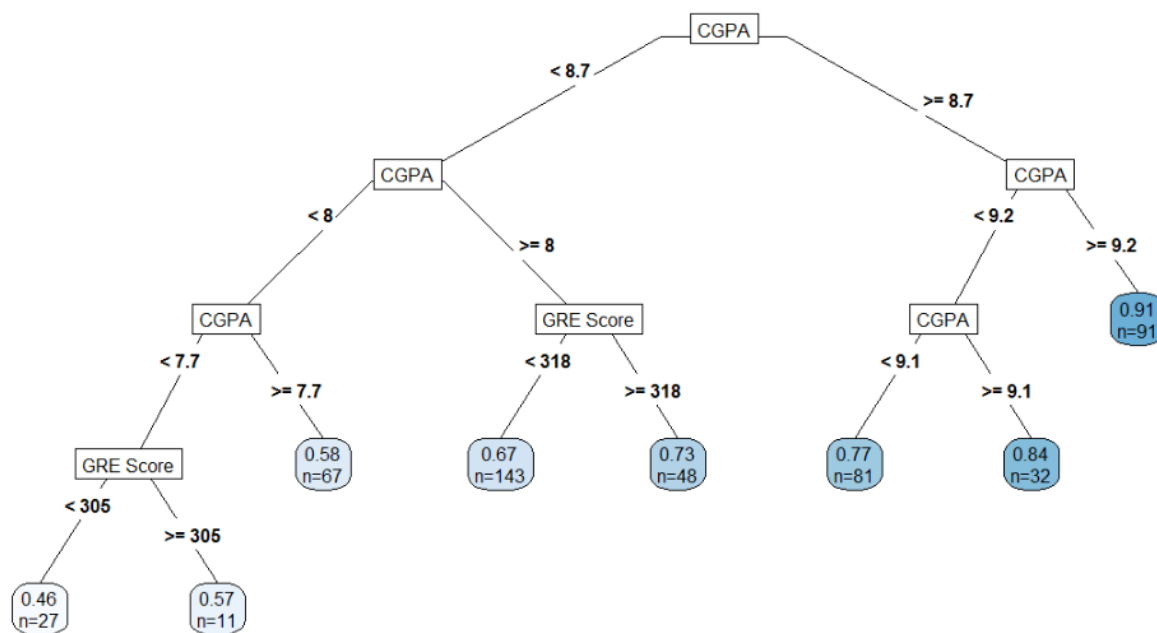
22 pav. II eksperimento verslo proceso modelis

Žingsnyje „Pateikti prašymą studijuoti ir pridėti reikiamus dokumentus“ nurodomi parametrai: *GRE Score*, *TOEFL Score*, *Research*. Žingsnyje „Perduoti dokumentus vertinimo komisijai“

nurodomi parametrai: *University Rating*, *SOP*, *LOR*, *CGPA*. Simuliacijos rezultatas įrašytas *Chance of Admit*, kuris nurodo studentų priėmimo tikimybę pagal priimtų ir nepriimtų studentų skaičius.

5.2.2. Eksperimento eiga ir rezultatai

II eksperimento metu sprendimų medis generuojamas pagal visus atributus, o medžio lapams, kuriuose formuojamos grupės, nurodytas atributas *Chance of Admit*. Sugeneruotas sprendimų medis pateiktas 23 paveiksle.



23 pav. II eksperimento sprendimų medis

Galima pastebėti, kad formuojant sprendimų medį CART algoritmas naudingiausiu skaidymui atributu net 5 kartus pasirinko *GRE Score*. Taigi vien iš sprendimų medžio galima pastebėti, kuris parametras turi didžiausią įtaką priėmimo į magistrantūros studijas tikimybei. Toliau pagal suformuotą sprendimų medį nustatyta 14 taisyklių dalių, 25 įrašai taisyklės sudarančių taisyklių dalių sąrašė, 25 įrašai išliko ir sujungus šiuos sąrašus. Taisyklių gavimo etapai vaizduojami 24 paveiksle.

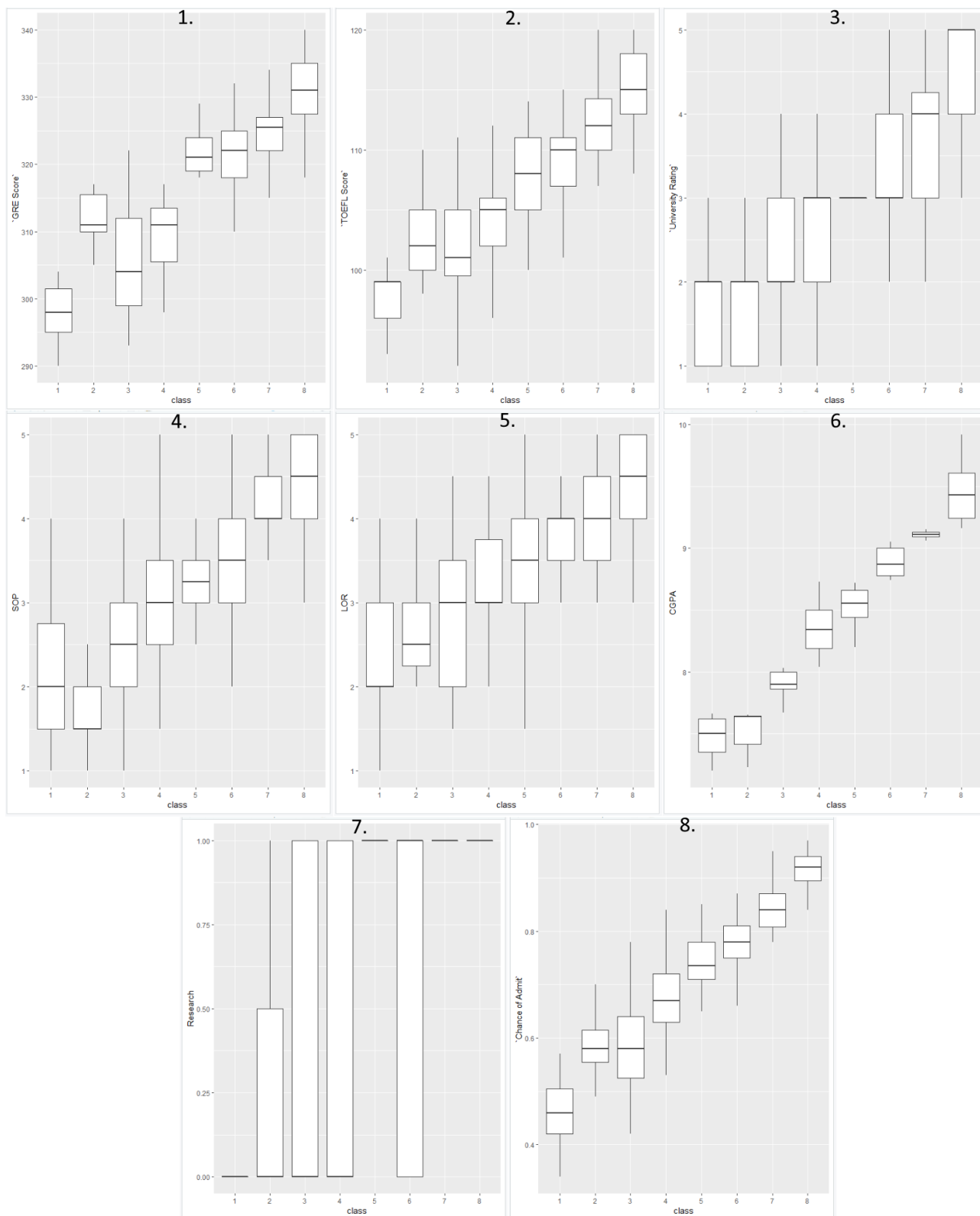
Subrule	Variable	Value	Less	Greater
1:	L1	CGPA	<NA>	8.735
2:	L2	CGPA	<NA>	8.035
3:	L3	CGPA	<NA>	7.665
4:	L4	GRE Score	<NA>	304.5
5:	L5	GRE Score	<NA>	317.5
6:	L6	CGPA	<NA>	9.155
7:	L7	CGPA	<NA>	9.055
8:	R1	CGPA	8.735	<NA>
9:	R2	CGPA	8.035	<NA>
10:	R3	CGPA	7.665	<NA>
11:	R4	GRE Score	304.5	<NA>
12:	R5	GRE Score	317.5	<NA>
13:	R6	CGPA	9.155	<NA>
14:	R7	CGPA	9.055	<NA>

Rule	Subrule	Leaf
1:	1	L1 TRUE
2:	1	L2 TRUE
3:	1	L3 TRUE
4:	1	L4 TRUE
5:	2	L1 TRUE
6:	2	L2 TRUE
7:	2	L3 TRUE
8:	2	R4 TRUE
9:	3	L1 TRUE
10:	3	L2 TRUE
11:	3	R3 TRUE
12:	4	L1 TRUE
13:	4	R2 TRUE
14:	4	L5 TRUE
15:	5	L1 TRUE
16:	5	R2 TRUE
17:	5	R5 TRUE
18:	6	R1 TRUE
19:	6	L6 TRUE
20:	6	L7 TRUE
21:	7	R1 TRUE
22:	7	L6 TRUE
23:	7	R7 TRUE
24:	8	R1 TRUE
25:	8	R6 TRUE

Subrule	Rule	Leaf	Variable	Value	Less	Greater
1:	L1	1 TRUE	CGPA	<NA>	8.735	<NA>
2:	L1	2 TRUE	CGPA	<NA>	8.735	<NA>
3:	L1	3 TRUE	CGPA	<NA>	8.735	<NA>
4:	L1	4 TRUE	CGPA	<NA>	8.735	<NA>
5:	L1	5 TRUE	CGPA	<NA>	8.735	<NA>
6:	L2	1 TRUE	CGPA	<NA>	8.035	<NA>
7:	L2	2 TRUE	CGPA	<NA>	8.035	<NA>
8:	L2	3 TRUE	CGPA	<NA>	8.035	<NA>
9:	L3	1 TRUE	CGPA	<NA>	7.665	<NA>
10:	L3	2 TRUE	CGPA	<NA>	7.665	<NA>
11:	L4	1 TRUE	GRE Score	<NA>	304.5	<NA>
12:	R4	2 TRUE	GRE Score	304.5	<NA>	304.5
13:	R3	3 TRUE	CGPA	7.665	<NA>	7.665
14:	R2	4 TRUE	CGPA	8.035	<NA>	8.035
15:	R2	5 TRUE	CGPA	8.035	<NA>	8.035
16:	L5	4 TRUE	GRE Score	<NA>	317.5	<NA>
17:	R5	5 TRUE	GRE Score	317.5	<NA>	317.5
18:	R1	6 TRUE	CGPA	8.735	<NA>	8.735
19:	R1	7 TRUE	CGPA	8.735	<NA>	8.735
20:	R1	8 TRUE	CGPA	8.735	<NA>	8.735
21:	L6	6 TRUE	CGPA	<NA>	9.155	<NA>
22:	L6	7 TRUE	CGPA	<NA>	9.155	<NA>
23:	L7	6 TRUE	CGPA	<NA>	9.055	<NA>
24:	R7	7 TRUE	CGPA	9.055	<NA>	9.055
25:	R6	8 TRUE	CGPA	9.155	<NA>	9.155

24 pav. II eksperimento taisyklių gavimo etapai

Pagal gautas taisykles kiekvienam pradinio duomenų rinkinio įrašui pridėjus papildomą grupės atributą buvo sugeneruotos *boxplot* diagramos pagal kiekvieną iš atributų. Visos stačiakampių diagramos vaizduojamos 25 paveiksle.

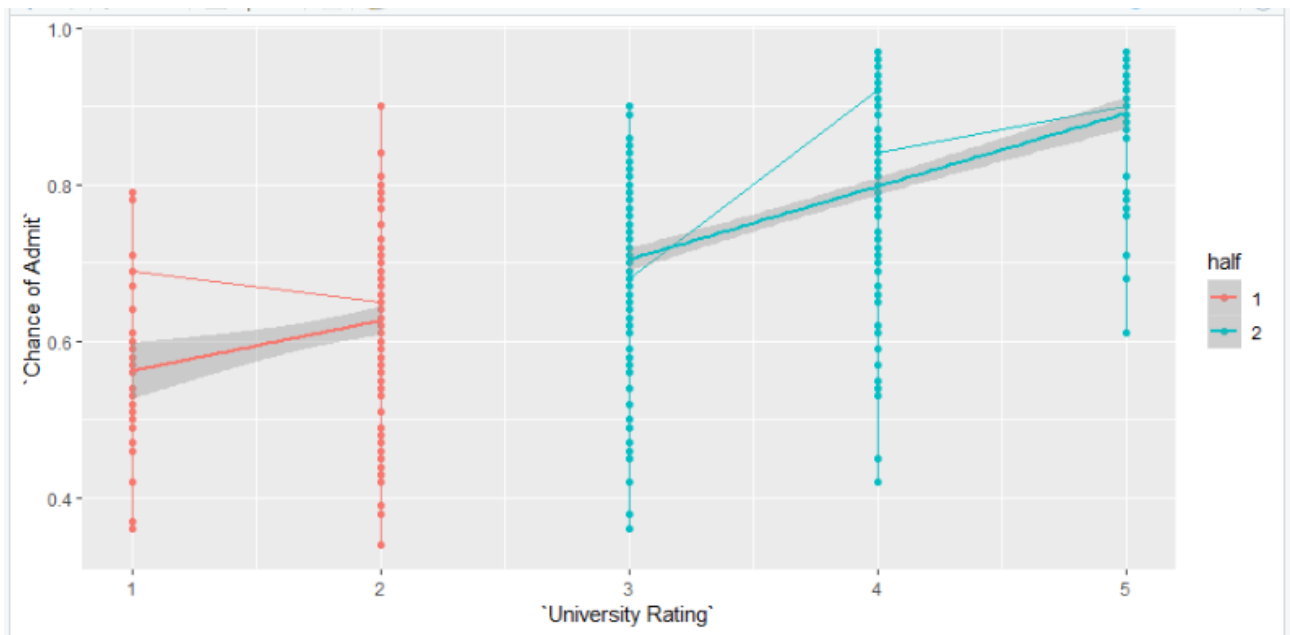


25 pav. II eksperimento stačiakampių diagramų atvaizdavimas

Iš 25 paveikslo galima pastebėti, kad visose diagramose, priklausomai nuo parametro didėjančios reikšmės, didėja ir priėmimo į studijas tikimybė. Be to, pagal kiekvieną iš šių diagramų galima daryti tam tikras išvadas apie analizuojamus duomenis:

1. Pirmoji diagrama vaizduoja įrašų pasiskirstymą grupėse pagal *GRE Score* parametą. Kadangi visi stačiakampiai persidengia bent su dviem kitais stačiakampiais, tai galima teigti, kad šis parametras neturi daug įtakos galutiniam rezultatui.
2. Antroji diagrama vaizduoja pasiskirstymą pagal *TOEFL Score* parametą. Praktiškai visi stačiakampiai persidengia vieni su kitais, taigi akivaizdu, kad ir nuo šio atributo nelabai priklauso rezultatas.
3. Trečioji diagrama pateikia ryšį su *University Rating* parametru. Šioje diagramoje galima įžvelgti tam tikrą tendenciją, kad kylant reitingui, proporcingai kyla ir tikimybė. Taip pat galima atkreipti dėmesį, kad ties 5 grupe nesusiformavo stačiakampė diagrama, taip nutiko dėl to, kad tam nepakako duomenų pasiskirstymo, dauguma jų susikoncentravę ties reitingu 3, kitur atsidūrė tik pavienės reikšmės.
4. Ketvirtoji diagrama vaizduoja *SOP*, tačiau ji nesuteikia daug žinių.
5. Penktoji diagrama pateikia duomenų pasiskirstymą grupėse pagal *LOR*. Stačiakampiai taip pat persidengia ir negalima pastebėti naudingos informacijos.
6. Šeštojoje diagramoje matomas aiškiausias atsiskyrimas tarp grupių, jos praktiškai nepersidengia. Šioje diagramoje rezultatai paskirstomi *CGPA* atžvilgiu, kuris buvo naudojamas kaip dažniausia sprendimų medžio skaidymo sąlyga, todėl toks stačiakampių išsidėstymas nestebina. Labiausiai išsiskiria 1 ir 2 grupės, kurios stipriai persidengia, iš to galima spręsti, kad į šias grupes patenkantys įrašai stipriai priklauso ir nuo kitų atributų.
7. Septinta diagrama, kuri rodo ryšį su *Research*, mažiausiai informatyvi, kadangi joje stačiakampiai užima arba visą plotą, arba tik pavienius taškus.
8. Aštuntoji diagrama vaizduoja įrašų pasiskirstymą grupėse pagal *Chance of Admit*. Šios diagramos svarbiausios suteikiamos žinios, kad sprendimų medžio algoritmu suformuotos grupės nurodo ne tik konkrečiai tikimybę, bet atsižvelgia į visą duomenų rinkinį, nes akivaizdžiai matomi persidengimai tarp stačiakampių.

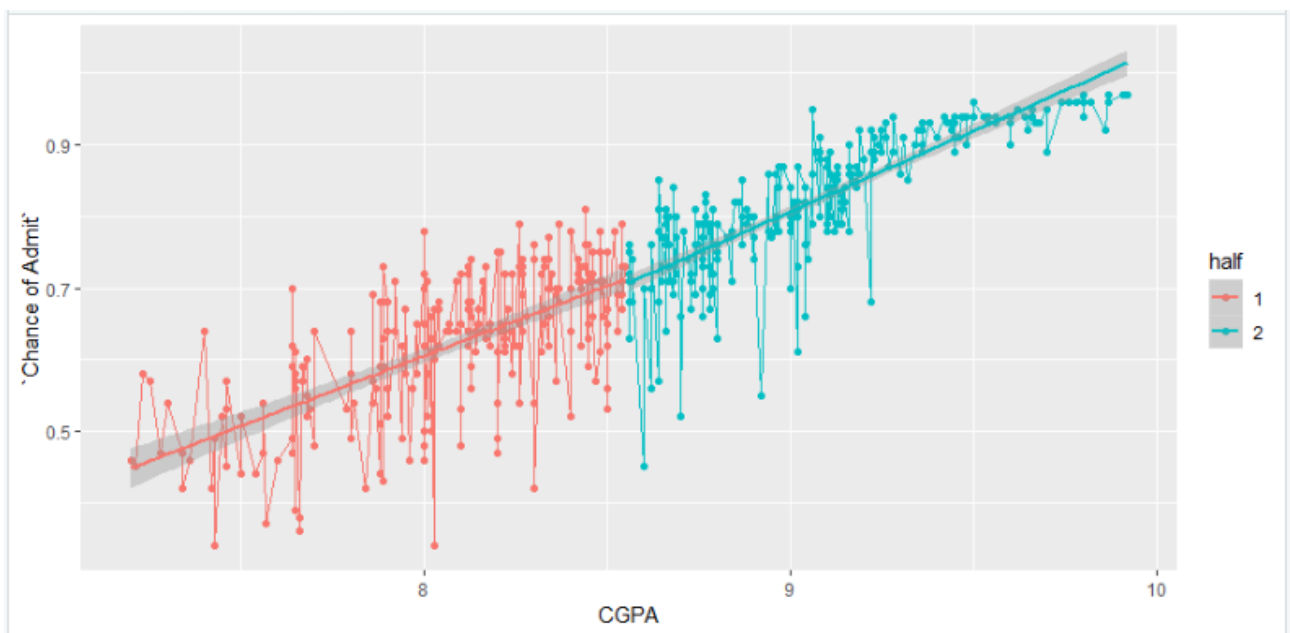
Apibendrinant galima teigti, kad norint smarkiai pakeisti simuliacijos rezultatus tokiaame verslo procese reikėtų koreguoti žingsnius, kurie susiję su *University Rating* ir *CGPA*. Siekiant nustatyti su kokiomis parametro reikšmėmis gaunama daugiau tinkamesnių rezultatų, sugeneruojamos priklausomybių diagramos. *University Rating* priklausomybių diagrama vaizduojama 26 paveiksle.



26 pav. Rezultatų priklausomybė nuo parametro *University Rating*

Priklausomybių diagramoje raudonai žymimos reikšmės, kurios pateko į pirmąją grupę, nuo mažiausios parametro *University Rating* vertės iki medianos, o mėlynai – patekusios į antrąją grupę, nuo medianos iki didžiausios vertės. Pilkame fone esančios linijos žymi regresiją – parametro vidurkio priklausomybę nuo rezultatų. Šių duomenų rinkinio atveju tinkamesni rezultatai yra tada, kai gaunama kuo didesnė *Chance of Admit* reikšmė, iš diagramos galima pastebėti, kad antros grupės įrašai nurodo tinkamesnius rezultatus. Taigi sugeneruota diagrama parodo, kad norint gauti geresnius rezultatus reikia didinti *University Rating* parametro reikšmes.

CGPA priklausomybių diagrama vaizduojama 27 paveiksle.



27 pav. Rezultatų priklausomybė nuo parametro *CGPA*

Rezultatų priklausomybės nuo parametro *CGPA* diagrama sugeneruota tokiu pat principu kaip ir 17 paveiksle vaizduota diagrama. Iš diagramos galima pastebėti, kad geresni rezultatai pasiekiami su antroje grupėje esančiomis parametro reikšmėmis. Taigi priklausomybės diagrama parodo, kad didinant *CGPA* parametro reikšmes, galima gauti geresnius rezultatus.

5.3. III eksperimentas

Remiantis dviem aukščiau aprašytais eksperimentais buvo sukurtas verslo proceso simuliacijos modelis, kuriam nurodyti šeši kintantys parametrai. Atsižvelgiant į metodo darbo su skaitiniais ir tekstiniais duomenimis rezultatai, nuspręsta visas reikšmes naudoti tik skaitines. Procesas buvo simuliuojamas 100 kartų, kai kiekvieną kartą žymima, kokie buvo įvesties parametrai ir koks gautas simuliacijos rezultatas. Kadangi iš Niclas'o Feldkamp'o, Soeren'o Bergmann'o ir Steffen'o Strassburger'o siūlomos idėjos šiame darbe atliekama tik duomenų analizės dalis, duomenų rinkinys buvo suformuotas neautomatizuotu būdu, surašant kiekvieną iš simuliacijoje naudotų parametrų ir gautų rezultatų kaip atskirus duomenų rinkinio įrašus. Dėl neautomatizuoto darbo į duomenų rinkinį nebuvo įtraukiami netikslūs rezultatai, todėl papildomai apdoroti duomenų nereikėjo.

5.3.1. Naudojamas duomenų rinkinys

Atlikti trečiam tyrimui naudoti iš vykdytos simuliacijos gauti duomenys. Iš viso surinkta 100 įrašų, kurių kiekvienas apibūdina vienos simuliacijos eigą. Simuliacijai sukurtas užsakymų vykdymo procesas, kuris aprašo užsakymo eigą nuo pirkėjo užsakymo pateikimo, užsakomų prekių tikrinimo ir rengimo iki pirkėjo informavimo, kad užsakymas įgyvendintas. Simuliacijos metu skaičiuojama kiek iš pateiktų 500 užsakymų buvo įgyvendinta per 24 valandas bei koks buvo darbuotojų užimtumas. Be to, pagal šiuos duomenis išvedamas naujas rezultatas – veiklos naudingumas, kuris susideda iš įgyvendintų užsakymų procentais ir darbuotojų užimtumo procentais, kai šie rezultatai turi lygų naudingumo koeficientą. Galutinio duomenų rinkinio antraštės eilutė ir penki pirmieji įrašai pateikiami 6 lentelėje.

6 lentelė. III eksperimente naudoto duomenų rinkinio antraštė ir 5 pirmieji įrašai

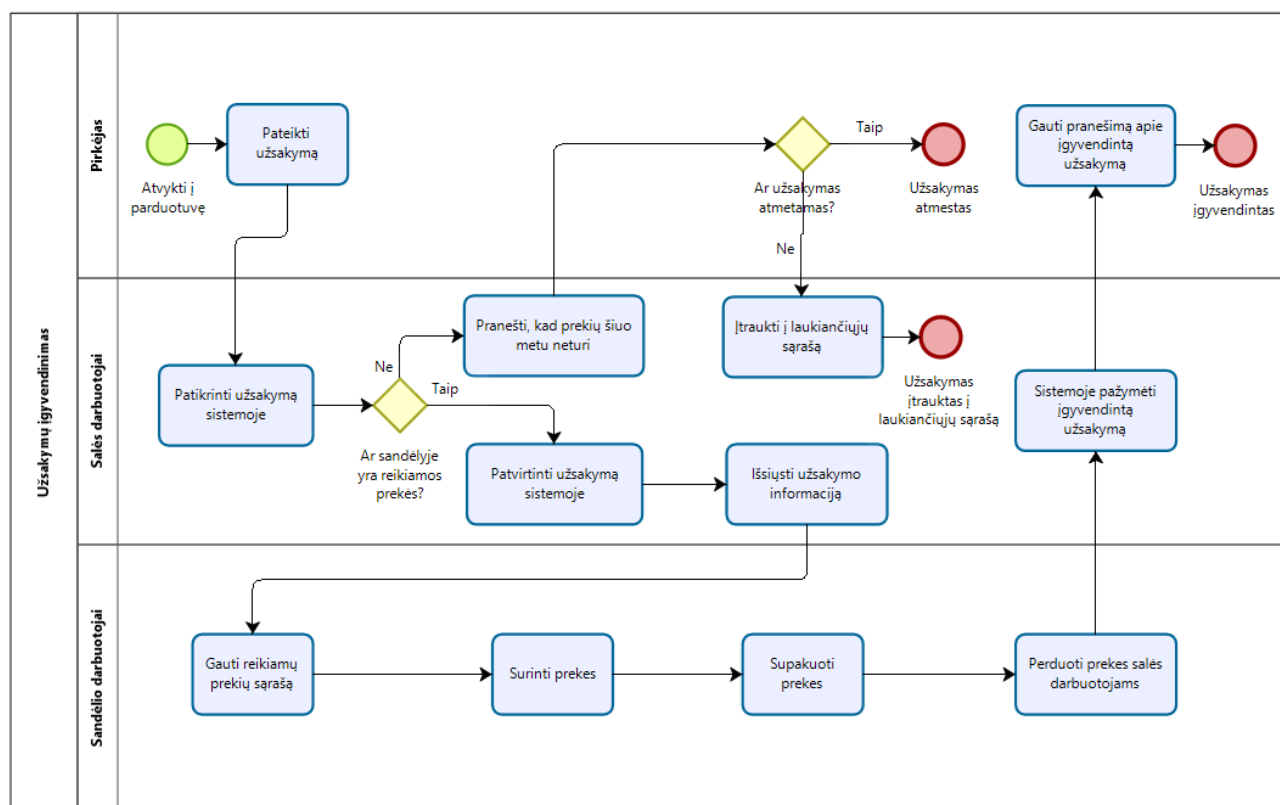
Darbuotojų kiekis	Priėmimo laikas	Priimančios darbuotojai	Vykdymo laikas	Vykdančios darbuotojai	Įgyvendinta proc.	Darbuotojų užimtumas	Veiklos naudingumas
13	2	1	20	4	40,80 %	93,78 %	67,29%
10	2	1	20	4	26,80 %	81,39 %	54,10%
10	2	1	17	3	46,80 %	90,69 %	68,75%
9	2	1	16	2	66,40 %	89,75 %	78,08%
9	2	1	16	2	66,40 %	89,75 %	78,08%

Įrašus aprašo aštuoni atributai, penki iš jų nurodo naudotus parametrus, trys – gautus rezultatus. Visi atributai yra skaitiniai, vieni žymi realiuosius skaičius, kiti procentinę dalį. Naudoti parametrai:

1. *Darbuotojų kiekis* – darbuotojų skaičius įmonėje, kurie dirba su užsakymais (salės ir sandėlio darbuotojų), reikšmės nuo 7 iki 20
2. *Priėmimo laikas* – vidutinis užsakymo priėmimo laikas, per kurį salės darbuotojai patikrina ar prekių yra sandėlyje ir informaciją apie užsakymą perduoda sandėlio darbuotojams, reikšmės nuo 2 iki 15 minučių.
3. *Priimančys darbuotojai* – užsakymui priimti reikalingas darbuotojų skaičius, nuo 1 iki 2.
4. *Vykdymo laikas* – vidutinis užsakymo vykdymo laikas, per kurį sandėlio darbuotojai surenka, supakuoja ir perduoda prekes, reikšmės nuo 15 iki 30
5. *Vykdantys darbuotojai* – užsakymui įvykdyti reikalingas darbuotojų skaičius, nuo 1 iki 5.

Rezultatai pateikiami trimis atributais: *Įgyvendinta proc.*, *Darbuotojų užimtumas* ir *Veiklos naudingumas*. *Įgyvendinta proc.* nurodo, kiek procentų iš 500 pateiktų užsakymų buvo įgyvendinta, kai skaičiuojama pagal įgyvendintų užsakymų kiekį. *Darbuotojų užimtumas* nurodo, kiek laiko procentais darbuotojai dirbo. *Veiklos naudingumas* rodo bendrą įgyvendintų užsakymų ir darbuotojų užimtumo įvertinimą, kai abi sudedamosios dalys sudaro po 50 % naudingumo.

Verslo proceso simuliacijos modelis pateiktas 28 paveiksle.

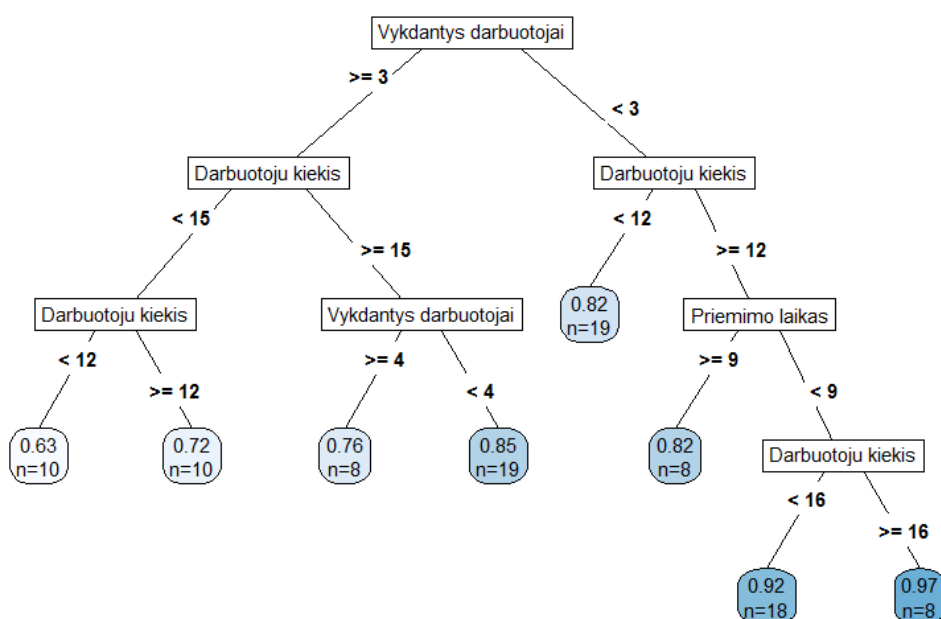


28 pav. III eksperimento verslo proceso simuliacijos modelis

Parametras *Darbuotojų kiekis* nurodomas bendrai visam procesui. Žingsnyje „Išsiųsti užsakymo informaciją“ nurodomi parametrai: *Priėmimo laikas* ir *Priimančys darbuotojai*. Žingsnyje „Perduoti prekes salės darbuotojams“ nurodomi parametrai: *Vykdymo laikas* ir *Vykdančys darbuotojai*. Šie parametrai nurodo bendrą visų salės arba sandėlio darbuotojų veiklų laiką ir reikalingų darbuotojų skaičių. Duomenų rinkinys formuojamas tik iš tų įrašų, kurie pasiekė sėkmingą pabaigą, kai užsakymas įgyvendinamas.

5.3.2. Eksperimento eiga ir rezultatai

III eksperimento metu sprendimų medis generuojamas pagal visus atributus, kai grupių reikšmės medžio lapuose nurodytas atributas *Veiklos naudingumas*, kuris apima ir įgyvendintų darbų, ir darbuotojų užimtumo rezultatus. Sugeneruotas sprendimų medis pateiktas 29 paveiksle.



29 pav. III eksperimento sprendimų medis

Pagal suformuotą sprendimų medį nustatyta 14 taisyklių dalių, 25 įrašai taisyklių ir jų dalių sąrašė, 25 įrašai išliko ir sujungus šiuos sąrašus. Taisyklių gavimo etapai vaizduojami 30 paveiksle.

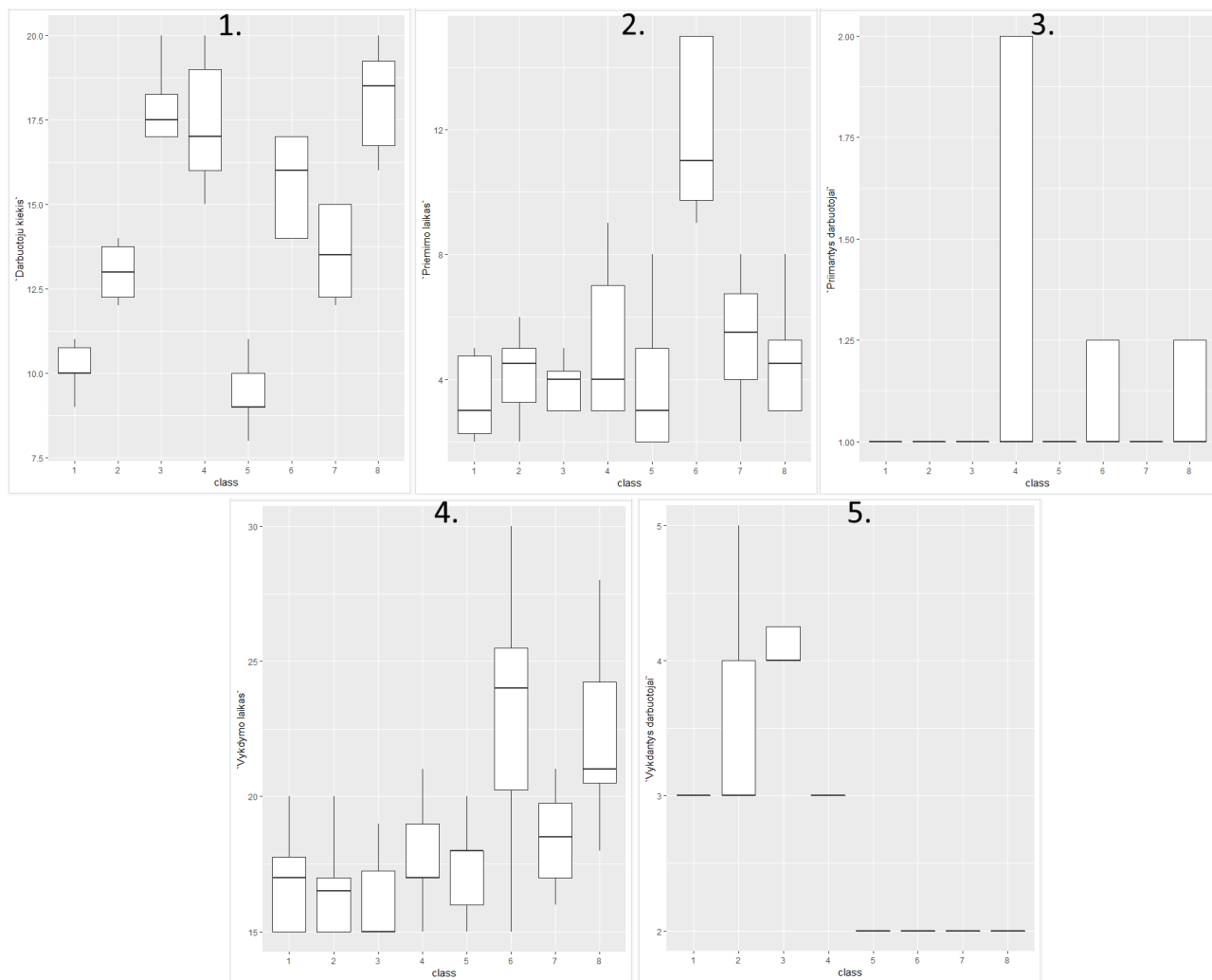
Subrule	variable	value	Less	Greater
1	L1 vykdančys darbuotojai	2.5	<NA>	2.5
2	L2 darbuotojų kiekis	<NA>	14.5	<NA>
3	L3 darbuotojų kiekis	<NA>	11.5	<NA>
4	L4 vykdančys darbuotojai	3.5	<NA>	3.5
5	L5 darbuotojų kiekis	<NA>	11.5	<NA>
6	L6 Priėmimo laikas	8.5	<NA>	8.5
7	L7 darbuotojų kiekis	<NA>	15.5	<NA>
8	R1 vykdančys darbuotojai	<NA>	2.5	<NA>
9	R2 darbuotojų kiekis	14.5	<NA>	14.5
10	R3 darbuotojų kiekis	11.5	<NA>	11.5
11	R4 vykdančys darbuotojai	<NA>	3.5	<NA>
12	R5 darbuotojų kiekis	11.5	<NA>	11.5
13	R6 Priėmimo laikas	<NA>	8.5	<NA>
14	R7 darbuotojų kiekis	15.5	<NA>	15.5

Rule	Subrule	Leaf
1:	1	L1 TRUE
2:	1	L2 TRUE
3:	1	L3 TRUE
4:	2	L1 TRUE
5:	2	L2 TRUE
6:	2	R3 TRUE
7:	3	L1 TRUE
8:	3	R2 TRUE
9:	3	L4 TRUE
10:	4	L1 TRUE
11:	4	R2 TRUE
12:	4	R4 TRUE
13:	5	R1 TRUE
14:	5	L5 TRUE
15:	6	R1 TRUE
16:	6	R5 TRUE
17:	6	L6 TRUE
18:	7	R1 TRUE
19:	7	R5 TRUE
20:	7	R6 TRUE
21:	7	L7 TRUE
22:	8	R1 TRUE
23:	8	R5 TRUE
24:	8	R6 TRUE
25:	8	R7 TRUE

Subrule	Rule	Leaf	variable	value	Less	Greater
1:	L1	1 TRUE	vykdančys darbuotojai	2.5	<NA>	2.5
2:	L1	2 TRUE	vykdančys darbuotojai	2.5	<NA>	2.5
3:	L1	3 TRUE	vykdančys darbuotojai	2.5	<NA>	2.5
4:	L1	4 TRUE	vykdančys darbuotojai	2.5	<NA>	2.5
5:	L2	1 TRUE	Darbuotojų kiekis	<NA>	14.5	<NA>
6:	L2	2 TRUE	Darbuotojų kiekis	<NA>	14.5	<NA>
7:	L3	1 TRUE	Darbuotojų kiekis	<NA>	11.5	<NA>
8:	R3	2 TRUE	Darbuotojų kiekis	11.5	<NA>	11.5
9:	R2	3 TRUE	Darbuotojų kiekis	14.5	<NA>	14.5
10:	R2	4 TRUE	Darbuotojų kiekis	14.5	<NA>	14.5
11:	L4	3 TRUE	vykdančys darbuotojai	3.5	<NA>	3.5
12:	R4	4 TRUE	vykdančys darbuotojai	<NA>	3.5	<NA>
13:	R1	5 TRUE	vykdančys darbuotojai	<NA>	2.5	<NA>
14:	R1	6 TRUE	vykdančys darbuotojai	<NA>	2.5	<NA>
15:	R1	7 TRUE	vykdančys darbuotojai	<NA>	2.5	<NA>
16:	R1	8 TRUE	vykdančys darbuotojai	<NA>	2.5	<NA>
17:	L5	5 TRUE	Darbuotojų kiekis	<NA>	11.5	<NA>
18:	R5	6 TRUE	Darbuotojų kiekis	11.5	<NA>	11.5
19:	R5	7 TRUE	Darbuotojų kiekis	11.5	<NA>	11.5
20:	R5	8 TRUE	Darbuotojų kiekis	11.5	<NA>	11.5
21:	L6	6 TRUE	Priėmimo laikas	8.5	<NA>	8.5
22:	R6	7 TRUE	Priėmimo laikas	<NA>	8.5	<NA>
23:	R6	8 TRUE	Priėmimo laikas	<NA>	8.5	<NA>
24:	L7	7 TRUE	Darbuotojų kiekis	<NA>	15.5	<NA>
25:	R7	8 TRUE	Darbuotojų kiekis	15.5	<NA>	15.5

30 pav. III eksperimento taisyklių gavimo etapai

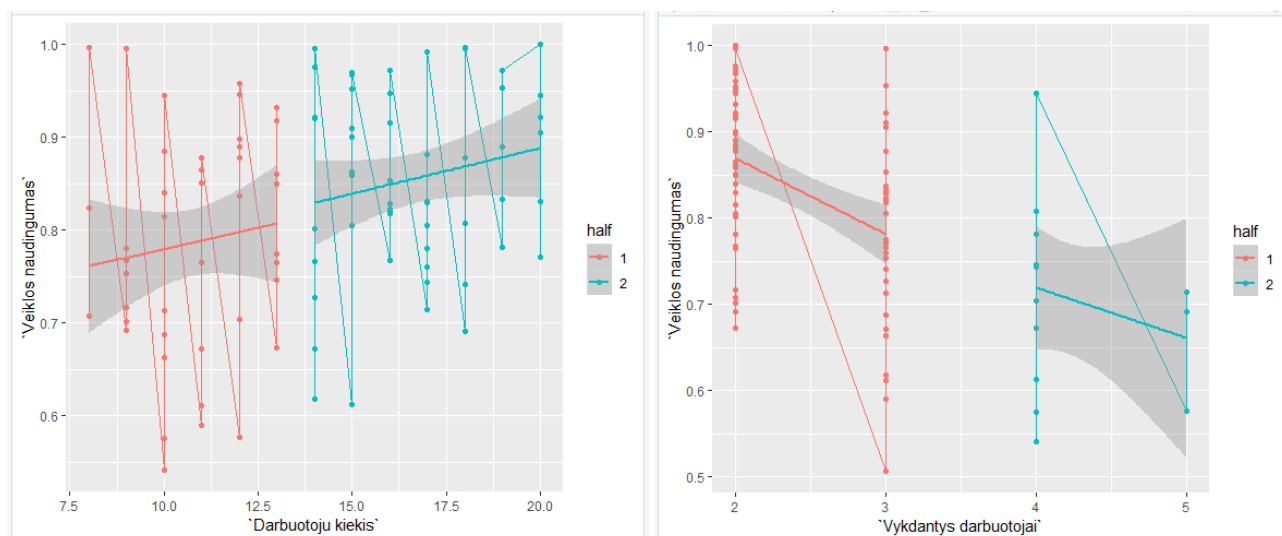
Pagal gautas taisykles kiekvienam duomenų rinkinio įrašui buvo nustatyta grupė, į kurią įrašas patenka, pagal grupes, kiekvienam iš parametrų, sugeneruotos *boxplot* diagramos. Visos stačiakampių diagramos vaizduojamos 31 paveiksle.



31 pav. III eksperimento stačiakampių diagramų atvaizdavimas

Didžioji dalis gautų diagramų rodo, kad parametrų reikšmės didelės įtakos galutiniam rezultatui neturi. Ypač 2, 3 ir 4 diagramose beveik visi stačiakampiai persidengia tarpusavyje didžiąja dalimi savo ploto. Tačiau iš 1 ir 5 diagramų galima pastebėti didesnę grupių pasiskirstymą pagal parametrų reikšmes. 1 diagrama vaizduoja įrašų pasiskirstymą pagal parametą *Darbuotojų kiekis*, joje stačiakampiai persidengia, tačiau yra smarkiai pasisklaidę vertikaliajoje ašyje, panašu, kad šis parametras turi didžiausią įtaką galutiniam rezultatui. Tuo tarpu iš 5 diagramos, kuri pateikia pasiskirstymą pagal parametą *Vykdanys darbuotojai*, galima pastebėti, kad grupės pasiskirsto į dvi dalis, į paskutines keturias grupes patenka įrašai, kurių reikšmės lygios 2, o į pirmas keturias – likusios reikšmės. Taigi iš vizualaus ryšio tarp parametrų ir sprendimų medžio suklasifikuotų grupių nustatyta, kad didžiausią įtaką rezultatams turi *Darbuotojų kiekis* ir *Vykdanys darbuotojai*. Siekiant nustatyti

su kokiomis parametro reikšmėmis gaunamos didesnės rezultatų *Veiklos naudingumas* reikšmės, sugeneruojamos priklausomybių diagramos. Priklausomybės diagramos pateiktos 32 paveiksle.



32 pav. Rezultatų priklausomybė nuo parametų *Darbuotojų kiekis* ir *Vykdytys darbuotojai*

Pagal priklausomybių vizualizaciją galima pastebėti, kad siekiant gauti aukštesnę rezultatų atributo *Veiklos naudingumas* reikšmę *Darbuotojų kiekis* parametro reikšmę reikia padidinti, o *Vykdytys darbuotojai* – sumažinti. Remiantis šiomis žiniomis buvo atlikti 5 simuliacijos bandymai, kur didintas pirmasis parametras, 5 bandymai, kur mažintas antrasis ir 5 bandymai, kai keičiami abu parametrai, rezultatai pateikti 7 lentelėje.

7 lentelė. Nustatytų, labiausiai rezultatus įtakančiais, parametų tyrimo rezultatai

Darbuotojų kiekis	Priėmimo laikas	Priimančys darbuotojai	Vykdymo laikas	Vykdytys darbuotojai	Įgyvendinta proc.	Darbuotojų užimtumas	Veiklos naudingumas
10	2	1	20	4	26,80 %	81,39 %	54,10 %
11	2	1	20	4	26,80 %	74,44 %	50,62 %
12	2	1	20	4	40,20 %	100 %	70,10 %
13	2	1	20	4	40,80 %	92,69 %	66,75 %
14	2	1	20	4	40,80 %	86,43 %	63,62 %
14	2	1	20	5	27,20 %	72,82 %	50,01 %
14	2	1	20	4	40,80 %	86,43 %	63,62 %
14	2	1	20	3	54,40 %	86,42 %	70,41 %
14	2	1	20	2	95,20 %	100 %	97,60 %
14	2	1	20	1	100 %	99,21 %	99,61 %
10	2	1	20	5	26,80 %	100 %	63,40 %
11	2	1	20	4	26,80 %	74,44 %	50,62 %
12	2	1	20	3	53,60 %	99,99 %	76,80 %
13	2	1	20	2	81,60 %	92,71 %	87,16 %
14	2	1	20	1	100 %	99,21 %	99,61 %

Atsižvelgiant į bandymų rezultatus galima teigti, kad šių parametrų numatyti pakeitimai iš tiesų generuoja geresnį galutinį simuliacijos rezultatą. Taip pat galima pastebėti, kad koreguojant parametras *Vykdančys darbuotojai* rezultatai kinta greičiausiai. Tačiau šio proceso atveju nebūtų tikslo keisti tik vykdančių užsakymų darbuotojų skaičių, nerealistiška būtų manyti, kad kelių darbuotojų darbą įstengs atlikti vienas darbuotojas, per tokį pat terminą. Taigi sprendžiant verslo proceso optimizavimo klausimą tiksliausia būtų derinti abu parametrus, kol rezultatas patenkins poreikius.

5.4. Apibendrinimas ir rezultatai

Atlikti eksperimentai parodė, kad pasiūlytas metodas tinka taisyklių nustatymui iš verslo procesų simuliacijos rezultatų. Taip pat, remiantis eksperimento rezultatais, galima teigti, kad šis metodas padeda interpretuoti simuliacijos rezultatus bei atitinkamų parametrų ryšį su rezultatais. Remiantis pateiktomis eksperimentų rezultatų interpretacijomis, galima nustatyti, kokius parametrus ir kaip reikėtų koreguoti simuliacijos procese, kad būtų gaunami norimi simuliacijos rezultatai. Taigi pasiūlytas metodas leidžia gauti daugiau žinių apie simuliuojamą procesą bei šias žinias panaudoti proceso tobulinimui.

Vykdam metodą su skirtingais duomenų rinkiniais nustatyti ir metodo trūkumai:

1. Metode naudojami algoritmai lėtai apdoroja didelę duomenų imtį. Visi skaičiavimai atliekami imant po vieną įrašą iš duomenų rinkinio, tad algoritmo vykdymo laikas tiesiogiai priklauso nuo iteracijų kiekio.
2. Metodas labai lėtai veikia su tekstiniais duomenimis. Nėra aišku ar visais atvejais pavyksta juos sėkmingai suklasifikuoti, nes dažnai bandant tai atlikti programinė įranga nustodavo funkcionuoti, tačiau tai susiję su techninės įrangos galimybėmis, todėl rezultatas nėra iki galo aiškus.
3. Metodo rezultatai pateikia mažiau naudingų žinių, kai sprendimų medžio generavimo etape nurodoma, kuriuos konkrečiai parametrus reikia naudoti grupavimui.

DARBO REZULTATAI

Magistro baigiamojo darbo literatūros analizės skyriuje apžvelgtos su darbu susijusios temos:

- Verslo procesai: kuo skiriasi statiniai, dinaminiai, siekiantys tikslo ir priklausantys nuo konteksto verslo procesai bei kaip procesus valdyti.
- Verslo procesų modeliavimas: klasifikavimas ir modeliavimo kalbos.
- Verslo procesų analizė ir optimizavimas.
- Verslo procesų simuliacija: jos privalumai ir trūkumai, vykdymo etapai, verslo proceso modelio ir simuliacijos modelio skirtumai.
- Duomenų gavybos metodai: grupavimo, klasifikavimo, asociacijų, regresijos.

Taisyklių iš verslo procesų simuliacijos rezultatų gavimo algoritmų tipų analizėje buvo apžvelgti penki dažniausiai naudojami algoritmų tipai: sprendimų medžių, genetiniai, mišrūs sprendimų medžių / genetiniai, K-vidurkių ir neuroninių tinklų algoritmai. Pasirinkta, kad tolimesniems tyrimams bus naudojamas duomenų gavybos sprendimų medžio algoritmas, nes jo rezultatai leidžia ne tik vaizdžiai pateikti, bet ir akivaizdžiai pastebėti rezultatų pasiskirstymą.

Pasiūlytas duomenų gavybos iš simuliacijos rezultatų metodas, kurio metu bus atliekama duomenų gavyba, tuomet įvesties ir išvesties duomenų ryšys pateikiamas vizualiai. Taip pat nustatyta, kaip pradinius duomenis reikės apdoroti, kad juos būtų galima naudoti duomenų gavyboje. Aprašytas sprendimų medžio algoritmo veikimas, kuris bus naudojamas duomenų gavyboje, pateiktas jo generavimo pavyzdys. Pasirinktas naudoti CART sprendimų medžio algoritmas, kurio didžiausias pranašumas prieš kitus algoritmus – nutolusių reikšmių aptikimas. Apžvelgtas įvesties ir išvesties duomenų ryšio vaizdavimas, kai kiekvieno įvesties parametro atžvilgiu atvaizduojamas rezultatų grupių pasiskirstymas, nuspręsta toliau remtis šio pavyzdžio idėja.

Realizuotas siūlomas metodas, kuris įgyvendintas naudojant *RStudio* programinę įrangą. Aprašyti metodo realizacijos žingsniai: aplinkos paruošimas, sprendimų medžio generavimas, taisyklių gavimas, duomenų įrašų papildymas nustatytos grupės atributu, stačiakampių diagramų generavimas ir labiausiai įrašų pasiskirstymą grupėse įtakančių parametrų ryšio su rezultatais vizualizavimas. Atlikti eksperimentiniai tyrimai, pateikti jų rezultatai bei gautų rezultatų interpretavimas.

IŠVADOS

1. Atlikus su verslo procesais susijusių mokslinių darbų literatūros analizę, nustatyta, kad verslo procesų simuliacijos rezultatų analizė nėra plačiai ištirta tema ir pasirinkta konkreti tolimesnio tyrimo sritis – žinių gavimas iš simuliacijos rezultatų.
2. Duomenų gavybos algoritmų tipų galimybių, privalumų ir trūkumų analizė parodė, kad tinkamiausias tipas, kuris turi pakankamai daug galimybių, yra gana tikslus ir padeda valdyti stipriai išsiskiriančius duomenis – sprendimų medžių.
3. Pasiūlytas duomenų gavybos metodas leidžia, atsižvelgiant į simuliacijos įvesties parametrus ir gautas rezultatų grupes, gauti žinių apie simuliuojamą verslo procesą.
4. Atlikti eksperimentiniai tyrimai parodė, kad realizuotas metodas gali išgauti žinias iš verslo procesų simuliacijos rezultatų ir nustatyti labiausiai rezultatus įtakančius parametrus.

TOLIMESNIŲ TYRIMŲ KRYPTYS

Realizuojant pasiūlytą metodą ir atliekant tyrimą buvo pastebėtos galimos tolimesnių tyrimų kryptys, kurios padėtų pritaikyti duomenų gavybos metodus verslo procesų simuliacijoje:

1. Pasiūlyto metodo išplėtimas pritaikant jį neapdorotiems, realiems verslo procesų simuliacijos, duomenų rinkiniams.
2. Pasiūlyto metodo algoritmo, kuris pradinių duomenų įrašams priskiria grupes, tyrimas su įvairesnės struktūros ir apimties duomenų rinkiniais.
3. Pasiūlyto metodo išplėtimas įvairesnėmis vizualinėmis rezultatų interpretacijomis.
4. Pasiūlyto metodo išplėtimas automatiškai sistemos vykdomomis įžvalgomis, kaip galima tobulinti verslo proceso simuliaciją.
5. Tyrimas, ar pritaikius šį metodą ir pastebėjus verslo proceso simuliacijos galimas korekcijas, atlikus pakeitimus simuliacijos rezultatai tampa tikslesni.
6. Taisyklių, gautų pritaikant duomenų gavybos metodus verslo proceso simuliacijos rezultatams, panaudojimas įvairiose srityse.

LITERATŪROS ŠALTINIAI

- Agarwal, R. and Tech, S. M. (2015). Genetic Algorithm in Data Mining. *International Journal of Advanced Research in Computer Science and Software Engineering*, 5(9), 631–634.
- Aguilar-Savén, R. S. (2004). Business process modelling: Review and framework. *International Journal of Production Economics*, 90(2), 129–149.
- Alasadi, S. A. and Bhaya, W. S. (2017). Review of data preprocessing techniques in data mining. *Journal of Engineering and Applied Sciences*, 12(16), 4102–4107.
- Allweyer, T. (2016). *BPMN 2.0 : Introduction to the Standard for Business Process Modeling* (2nd ed.). Norderstedt: Books on Demand.
- Bhargava, N., Sharma, G., Bhargava, R. and Mathuria, M. (2013). Decision Tree Analysis on J48 Algorithm for Data Mining. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(6), 1114–1119.
- Bhat, J. M. and Deshmukh, N. (2005). Methods for Modeling Flexibility in Business Processes. *Workshop on Business Process Modeling, Design and Support (BPMDS05), Proceedings of CAiSE05 Workshops*, 1–8.
- Caramiaux, B. and Tanaka, A. (2013). Machine Learning of Musical Gestures. *International Conference on New Interfaces for Musical Expression 2013 (NIME)*, 513–518.
- Carvalho, D. R. and Freitas, A. A. (2004). A hybrid decision tree/genetic algorithm method for data mining. *Information Sciences*, 163(1–3), 13–35.
- Feldkamp, N., Bergmann, S. and Strassburger, S. (2015). Knowledge Discovery in Manufacturing Simulations. *Proceedings of the 3rd ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, 3–12.
- Freitas, A. A. (2003). A Survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery. *Advances in Evolutionary Computing*, 819–845.
- Gabryelczyk, R. and Jurczuk, A. (2017). Does Experience Matter? Factors Affecting the Understandability of the Business Process Modelling Notation. *Procedia Engineering*, 182, 198–205.
- Gandrud, C. (2016). *Reproducible Research with R and R Studio* (2nd Editio). New York: Chapman and Hall/CRC.
- Gaur, P. (2012). Neural Networks in Data Mining. *International Journal of Electronics and Computer*

Science Engineering, 1(3), 1449–1453.

Ghasemi, S., Ghasemi, M. and Ghasemi, M. (2011). Knowledge Discovery in Discrete Event Simulation Output Analysis. *Communications in Computer and Information Science*, 241, 108–120.

Greenwood, D. (2016). What “goal-oriented” BPM means, from a business perspective. Retrieved January 25, 2018, from <https://www.whitestein.com/blog/what-goal-oriented-bpm-means-business-perspective>

Gupta, S. (2015). A Regression Modeling Technique on Data Mining. *International Journal of Computer Applications*, 116(9), 975–8887.

Hammer, M. and Champy, J. (1993). Reengineering the corporation: A manifesto for business revolution. *Business Horizons*, 36(5), 90–91.

Hssina, B., Merbouha, A., Ezzikouri, H. and Erritali, M. (2014). A comparative study of decision tree ID3 and C4.5. *International Journal of Advanced Computer Science and Applications, Special Issue on Advances in Vehicular Ad Hoc Networking and Applications*, 4(2), 13–19.

Jebari, K., Mediafi, M. and Elmoujahid, A. (2013). Parent Selection Operators for Genetic Algorithms. *International Journal of Engineering Research & Technology (IJERT)*, 2(11), 1141–1145.

Kellner, M. I., Madachy, R. J. and Raffo, D. M. (1999). Software process simulation modeling: Why? What? How? *Journal of Systems and Software*, 46(2), 91–105.

Kueng, P. and Kawalek, P. (1997). Goal-based business process models: creation and evaluation. *Business Process Management Journal*, 3(1), 17–38.

Kumbhare, T. A. and Chobe, S. V. . (2014). An Overview of Association Rule Mining Algorithms. *International Journal of Computer Science and Information Technologies*, 5(1), 927–930.

Lapouchnian, A. and Yu, E. (2016). Exploring Context Sensing in the Goal-Driven Design of Business Processes. *2016 IEEE 18th Conference on Business Informatics (CBI)*, 45–54.

Law, A. M. (2009). How to build valid and credible simulation models. *Proceedings of the 2009 Winter Simulation Conference (WSC)*, 37(2), 24–33.

Malinova, M. and Mendling, J. (2013). A Qualitative Research Perspective on BPM Adoption and the Pitfalls of Business Process Modeling. *International Conference on Business Process Management*, 77–88.

- Melao, N. and Pidd, M. (2000). A conceptual framework for understanding business processes and business process modelling. *Information Systems Journal*, 10(2), 105–129.
- Mili, H., Tremblay, G., Jaoude, G. B., Mili, H., Tremblay, G., Bou Jaoude, G. and Boussaidi, G. El. (2010). Business Process Modeling Languages: Sorting Through the Alphabet Soup. *ACM Computing Surveys*, 43(1), 1–56.
- Oyelade, O. J., Oladipupo, O. O. and Obagbuwa, I. C. (2010). Application of k-Means Clustering algorithm for prediction of Students' Academic Performance. (*IJCSIS*) *International Journal of Computer Science and Information Security*, 7(1), 292–295.
- Patel, N. and Hlupic, V. (2001). Dynamic business process modelling (BPM) for business process change. *International Journal of Simulation: Systems, Science & Technology*, 2(2), 65–76.
- Radeschütz, S., Schwarz, H. and Niedermann, F. (2015). Business impact analysis - a framework for a comprehensive analysis and optimization of business processes. *Computer Science - Research and Development*, 30(1), 69–86.
- Raibova, I. (2009). Business rules specification and business processes modelling. *Agricultural Economics (Zemědělská Ekonomika)*, 55(1), 20–24.
- Razavian, M. and Khosravi, R. (2008). Modeling Variability in Business Process Models Using UML. *Fifth International Conference on Information Technology: New Generations (Itng 2008)*, 82–87.
- Rima, A. and Vasilecas, O. (2015). Verslo procesų simuliacija : verslo procesų ir išteklių modelių reikalavimai. *Science – Future of Lithuania*, 7(3), 291–295.
- Ross, R. G. (2000). What Is a “Business Rule”? Retrieved January 7, 2018, from <https://www.brcommunity.com/articles.php?id=b005>
- Rouse, M. (2012). Dynamic BPM (business process management). Retrieved January 7, 2018, from <http://whatis.techtarget.com/definition/dynamic-BPM-business-process-management>
- Saidani, O. and Nurcan, S. (2007). Towards Context Aware Business Process Modelling. *8th Workshop on Business Process Modeling, Development, and Support (BPMDS'07), CAiSE*, 7, 1–9.
- Singh Gill, R. and Ashima. (2014). Neural Networks in Data Mining. *IOSR Journal of Engineering*, 4(3), 1–6.
- Singh, S. and Giri, M. (2014). Comparative Study Id3, Cart And C4.5 Decision Tree Algorithm: A

Survey. *International Journal of Advanced Information Science and Technology (IJAIST)*, 3(7), 47–52.

Soni, N. and Kumar, T. (2014). Study of Various Mutation Operators in Genetic Algorithms. *International Journal of Computer Science and Information Technologies*, 5(3), 4519–4521.

Szelagowski, M. (2014). Static and Dynamic Processes - BPM Leader. Retrieved January 3, 2018, from <http://www.bpmleader.com/2014/08/28/static-and-dynamic-processes/>

Teece, D. J. (2017). Business models and dynamic capabilities. *Long Range Planning*, 51(1), 1–10.

Van der Aalst, W. M. P. (2015). Business Process Simulation Survival Guide. *Handbook on Business Process Management 1*, 337–370.

Vasilecas, O., Kalibatiene, D. and Lavbič, D. (2016). Rule and context based dynamic business process modelling and simulation. *The Journal of Systems and Software*, 122, 1–15.

Vergidis, K., Tiwari, A. and Majeed, B. (2008). Business Process Analysis and Optimization: Beyond Reengineering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 38(1), 69–82.

Vom Brocke, J., Mathiassen, L. and Rosemann, M. (2014). Business Process Management. *Business & Information Systems Engineering*, 6(4), 189–189.

Vom Brocke, J., Zelt, S. and Schmiedel, T. (2016). On the role of context in business process management. *International Journal of Information Management*, 36(3), 486–495.