



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

ELEKTRONIKOS FAKULTETAS

ELEKTRONINIŲ SISTEMŲ KATEDRA

Vita JASEVIČIŪTĖ

**ATVIROJO KODO ĮRANKIŲ INTERNETO SVETAINĖMS OPTIMIZUOTI
TYRIMAS**

**INVESTIGATION OF OPEN SOURCE TOOLS FOR WEBSITE
OPTIMIZATION**

Magistro baigiamasis darbas

Informatikos inžinerijos studijų kryptis

Informacinių elektroninių sistemų studijų programa, valstybinis kodas 621E15003

Atvirojo kodo sistemų specializacija

Vilnius, 2014

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
ELEKTRONIKOS FAKULTETAS
ELEKTRONINIŲ SISTEMŲ KATEDRA

TVIRTINU
Katedros vedėjas

(parašas)
prof. habil. dr. R. Martavičius
2014 m. _____ mėn. ____ d.

Vita JASEVIČIŪTĖ

**ATVIROJO KODO ĮRANKIŲ INTERNETO SVETAINĖMS OPTIMIZUOTI
TYRIMAS**

**INVESTIGATION OF OPEN SOURCE TOOLS FOR WEBSITE
OPTIMIZATION**

Magistro baigiamasis darbas

Informatikos inžinerijos studijų kryptis

Informacinių elektroninių sistemų studijų programa, valstybinis kodas 621E15003

Atvirojo kodo sistemų specializacija

Vadovas	dr. doc. A. Serackis (mokslinis vardas ir laipsnis, vardas, pavardė)	_____	_____
		(parašas)	(data)
Konsultantas	dr. V. Buivydienė (mokslinis vardas ir laipsnis, vardas, pavardė)	_____	_____
		(parašas)	(data)

Vilnius, 2014

Vita JASEVIČIŪTĖ

Atvirojo kodo įrankių interneto svetainėms optimizuoti tyrimas. Magistro baigiamasis darbas informatikos inžinerijos laipsniui. Vilniaus Gedimino technikos universitetas. Vilnius, 2014, 80 p., 40 iliustr., 6 lent., 26 bibl., 4 priedai.

Vilniaus Gedimino technikos universitetas
Elektronikos fakultetas
Elektroninių sistemų katedra

ISBN
Egz. sk.
Data-.....-..... ISSN

Antrosios pakopos studijų **Informacinių elektroninių sistemų** programos magistro baigiamasis darbas

Pavadinimas **Atvirojo kodo įrankių interneto svetainėms optimizuoti tyrimas**
Autorius **Vita Jasevičiūtė**
Vadovas **doc. dr. Artūras Serackis**

Kalba:
lietuvių

Anotacija

Baigiamajame darbe ištirti atvirojo kodo sistemose įgyvendinti optimizavimo metodai ir įvertintas jų tinkamumas svetainėms optimizuoti. Naudojant trijų svetainių, esančių paskolų rinkoje („Bobutės Paskola“, „Popkreditas“, „Paskoliukas“) duomenis, atrinkti ir ištirti neautomatiniam svetainių optimizavimui naudojami rodikliai: išorinių nuorodų skaičius, paspaudimų skaičius, atmetimo rodiklis, svetainės puslapių atvertimo laikas ir kiti. Surinkti ir naudojami duomenys yra pusės metų laikotarpio. Apžvelgti klasikiniai ir populiarūs metodai: tiesinės prognozės metodas (LPC), autoregresinis slenkamųjų vidurkių metodas (ARMA), netiesinis autoregresinis neuronų tinklas (NAR), netiesinis autoregresinis su papildomu įėjimu neuronų tinklas (NARX) ir laiko vėlinimo neuronų tinklas (TDNN). Naudojant atrinktus duomenis, taikant minėtus metodus, keičiant jų parametrus, skaičiuotos vidutinės paklaidos (prognozės efektyvumas). Atlikta kelių tipų prognozė: pirmiausiai prognozuojamas paspaudimų skaičius pagal to paties kintamojo (paspaudimo skaičiaus) istorinius duomenis (LPC, ARMA, NAR), ir kai apmokymas vyksta pagal du skirtingus įėjimus: atmetimo rodiklį ir vizito trukmę (NARX, TDNN).

Geriausi rezultatai (mažiausios vidutinės paklaidos), buvo gauti taikant LPC metodą, grįstą 1 eilės filtro taikymu, ir 2 sluoksnių su 6 neuronais pirmame, 5 neuronais antrame sluoksnyje – NARX neuronų tinklą. Blogesni rezultatai gauti taikant 1 sluoksnio su 7 neuronais NARX neuronų tinklą ir ARMA metodą, grįstą [1 1] modelio eile. Prasčiausi rezultatai gauti taikant 1 sluoksnio su 6 neuronais TDNN ir 1 sluoksnio su 1 neuronu NAR neuronų tinklus. Mažiausia paklaida gauta taikant LPC metodą, prognozuojant „Paskoliuko“ duomenis – 7,6 %.

Prasminiai žodžiai: Duomenų prognozavimas, prognozės metodai, neuronų tinklas, tiesinės prognozės metodas, autoregresinis slenkamųjų vidurkių metodas, netiesinis autoregresinis neuronų tinklas, netiesinis autoregresinis su papildomu įėjimu neuronų tinklas, laiko vėlinimo neuronų tinklas.

Vilnius Gediminas Technical University Faculty of Electronics Department of Electronic Systems	ISBN _____ ISSN _____ Copies No. Date-.....-.....
--	--

Master Degree Studies Information Electronics Systems study programme Master Graduation Thesis	
Title	Investigation of Open Source Tools for Website Optimization
Author	Vita Jasevičiūtė
Academic supervisor	Assoc Prof Dr Artūras Serackis

Thesis language:
 Lithuanian

Annotation In a master thesis website optimization methods are examined using open source systems and their suitability for website optimization evaluated. For the research three websites operating in quick loans market are investigated: "Bobutes Paskola", "Popkreditas", and "Paskoliukas". Data, such as the number of external links pointing to a website, click through rate, bounce rate, page load time, is extracted from various sources and analyzed. The timeframe of this research is 6 months. In the thesis the following traditional and most popular methods are used: linear predictive coding method (LPC), autoregressive moving average model (ARMA), nonlinear autoregressive neural network (NAR), nonlinear autoregressive exogenous model (NARX) and time delay neural networks (TDNN). Mean squared errors i.e. the effectiveness of prognosis, is calculated applying methods discussed above, using sorted data and changing various parameters. In the thesis there are several types of hypotheses tested. Firstly, click through rate is estimated using historic data of the same variable, i.e. click through rate (LPC, ARMA, NAR). Secondly, click through rate is estimated when the training is done using two separate entities: bounce rate and time on site (NARX, TDNN). The best results, or in other words the lowest mean squared errors, are observed when LPC method and NARX neural network are used. LPC method is based on 1 order of the prediction filter polynomial. While NARX neural network is based on 2 layers with 6 neurons in the first layer, 5 neurons in second layer. Less accurate results are observed when one layer seven neurons NARX neural network and [1 1] polynomial orders ARMA method are used. The least accurate results are achieved when one layer 6 neurons TDNN and one layer one neuron NAR neuron networks are applied. The lowest deviation is achieved using LPC method when forecasting "Paskoliukas" data.

Keywords: Linear predictive coding, data prediction, prediction, forecasting, neural network, linear predictive coding, autoregressive moving average model, nonlinear autoregressive neural network, nonlinear autoregressive exogenous model, time delay neural networks.

TURINYS

Žymenys ir santrumpos	6
1. Įvadas	7
1.1. Darbo aktualumas ir tikslas	7
1.2. Darbo uždaviniai	7
1.3. Taikyti tyrimo ir analizės metodai	7
1.4. Darbo naujumas ir praktinė nauda	8
1.5. Darbo struktūra	8
2. Neautomatinių svetainių optimizavimo sprendimų analitinė apžvalga	9
2.1. Konversijų ir srauto didinimo būdai	9
2.2. Interneto svetainių rodiklių analizavimo įrankiai	11
2.2.1. Atvirojo kodo interneto svetainių rodiklių matavimo įrankis „Piwik“	11
2.2.2. Atvirojo kodo interneto svetainių rodiklių matavimo įrankis „Open Web Analytics“	12
2.2.3. Atvirojo kodo interneto svetainių rodiklių matavimo įrankis „Grape Web Statistics“	12
2.2.4. Atvirojo kodo interneto svetainių rodiklių matavimo įrankis „Google Analytics“	13
2.3. Atvirojo kodo įrankių interneto svetainės matavimo ir vertinimo rodikliai	14
2.4. Skyriaus apibendrinimas	16
3. Atvirojo kodo sistemose įgyvendintų prognozavimo metodų analizė	18
3.1. Prognozavimo metodų taikymo raida	18
3.2. Prognozavimo laiko srityje analizė	19
3.3. Ateities prognozavimas	20
3.3.1. Autoregresinis slenkamųjų vidurkių prognozės metodas – ARMA	21
3.3.2. Tiesinės prognozės metodas – LPC	22
3.3.3. Dirbtinis neuronų tinklas – DNT	23
3.4. Skyriaus apibendrinimas	26
4. Svetainės įvertinimui naudojamų priežastinių ir pasekminių rodiklių atranka	28
4.1. „Google“ algoritmų galimybių įvertinimas	28
4.2. Pagrindinių išorinės optimizacijos matavimo/vertinimo rodiklių atranka	29
4.3. Pagrindinių vidinės optimizacijos matavimo/vertinimo rodiklių įtakos analizė	30
4.3.1. Puslapio pavadinimo įtakos analizė	31
4.3.2. Puslapių antraščių įtakos analizė	32
4.3.3. Puslapio aprašymo įtakos analizė	33
4.3.4. URL struktūros įtakos analizė	33
4.3.5. Paveikslų alternatyvaus teksto įtakos analizė	34
4.3.6. Raktinių žodžių ir jų naudojimo svetainėje įtakos analizė	34
4.3.7. Puslapių atvertimo greičio įtakos analizė	35
4.4. Pagrindinių pasekmių matavimo/vertinimo rodiklių analizė	36
4.5. Interneto svetainių matuotų ir surinktų rodiklių atranka	37
4.6. Skyriaus apibendrinimas	39
5. Prognozavimo metodų efektyvumo tyrimo rezultatai ir jų analizė	40
5.1. Tiesinės prognozės metodas (LPC)	41
5.1.1. LPC metodo efektyvumo priklausomybių nuo lango pločio tyrimas	42
5.1.2. LPC metodo efektyvumo priklausomybių nuo filtro eilės tyrimas	43
5.2. Autoregresinis paslinktųjų vidurkių prognozavimo metodas (ARMA)	46
5.2.1. ARMA metodo efektyvumo priklausomybių nuo lango pločio tyrimas	47
5.2.2. ARMA metodo efektyvumo priklausomybių nuo modelio eilės tyrimas	48
5.3. Netiesinis autoregresinis neuronų tinklas (NAR)	50

5.3.1. NAR metodo efektyvumo priklausomybių nuo pirmo sluoksnio neuronų skaičiaus tyrimas	51
5.3.2. NAR metodo efektyvumo priklausomybių nuo išėjimo vėlinimo tyrimas	52
5.3.3. NAR metodo efektyvumo priklausomybių nuo mokymų skaičiaus tyrimas	53
5.4. Netiesinis autoregresinis su papildomų įėjimu neuronų tinklas (NARX)	54
5.4.1. NARX metodo efektyvumo priklausomybių nuo pirmo sluoksnio neuronų skaičiaus tyrimas	55
5.4.2. NARX metodo efektyvumo priklausomybių nuo antro sluoksnio neuronų skaičiaus tyrimas	57
5.4.3. NARX metodo efektyvumo priklausomybių nuo išėjimo vėlinimo tyrimas	58
5.4.4. NARX metodo efektyvumo priklausomybių nuo įėjimo vėlinimo tyrimas	59
5.4.5. NARX metodo efektyvumo priklausomybių nuo mokymų skaičiaus tyrimas	61
5.5. Laiko vėlinimo neuronų tinklas (TDNN)	63
5.5.1. TDNN metodo efektyvumo priklausomybių nuo pirmo sluoksnio neuronų skaičiaus tyrimas	64
5.5.2. TDNN metodo efektyvumo priklausomybių nuo vėlinimo tyrimas	65
5.5.3. TDNN metodo efektyvumo priklausomybių nuo mokymų skaičiaus tyrimas	66
5.6. Skyriaus apibendrinimas	67
6. Apibendrinimas. Išvados	71
Literatūra	73
PRIEDAI	75
A priedas. ARMA metodo pritaikymo programinis kodas	75
B priedas. NAR metodo pritaikymo programinis kodas	75
C priedas. NARX metodo pritaikymo programinis kodas	76
D priedas. TDNN metodo pritaikymo programinis kodas	78
E priedas. LPC metodas	79

Žymenys ir santrumpos

- A/B testas – dviejų tinklalapių versijų rodymas skirtingiems lankytojams
- GA (anlg. *Google Analytics*) – svetainės duomenų matavimo įrankis
- OWA (anlg. *Open Web Analytics*) – svetainės duomenų matavimo įrankis
- SEO (anlg. *Search Engine Optimization*) – paieškos variklių optimizacija
- GWS (anlg. *Grape Web Statistics*) – svetainės duomenų matavimo įrankis
- LPC (anlg. *linear prediction code*) – tiesinės prognozės metodas
- ARMA (anlg. *autoregressive moving average model*) – autoregresinis paslinktųjų vidurkių metodas
- NARX (anlg. *nonlinear autoregressive exogenous model*) – netiesinis autoregresinis su papildomu įėjimu neuronų tinklas
- NAR (anlg. *nonlinear autoregressive neural network*) – netiesinis autoregresinis neuronų tinklas
- TDNN (anlg. *time delay neural network*) – laiko vėlinimo neuronų tinklas
- IP (anlg. *internet protocol*) – interneto protokolas
- IG – investicijų grąža
- SSA – skaitmeninių signalų apdorojimas
- IRR (anlg. *infinite impulse response*) – beribės impulsinės reakcijos
- FIR (anlg. *finite impulse response*) – ribotos impulsinės reakcijos
- AR (anlg. *autoregressive*) – autoregresinis modelis
- MA (anlg. *moving average*) – slenkamųjų vidurkių modelis

1. ĮVADAS

1.1. Darbo aktualumas ir tikslas

Pagrindinis darbo tikslas – ištirti atvirojo kodo sistemose įgyvendintus optimizavimo metodus ir įvertinti jų tinkamumą svetainėms optimizuoti.

Kiekviena interneto svetainė, ar tai būtų elektroninė parduotuvė, asmeninis tinklaraštis, forumas, turi turėti aiškius matuojamus tikslus. Dėl tos priežasties yra matuojamas indikatorius: konversijos koeficientas – trupmeninis skaičius tarp 0 ir 1 (dažnai išreiškiamas procentine išraiška), kuris parodo lankytojo naudingą elgesį svetainėje. Kuo skaičius yra didesnis, tuo lankytojo elgesys svetainėje yra naudingesnis. Konversijų didinimas – tai procesas, kurio metu tobulinamas svetainės struktūros išdėstymas, funkcionalumas, dizaino elementai, mažinamas lankytojų, nuėjusių į tinklapį, peržiūrėjusių vos vieną puslapį ir jį palikusių, procentas (angl. *bounce rate*). Lankytoją stengiamasi nuvesti prie numatyto tikslo, pavyzdžiui, pirkimo, paslaugos užsakymo [1].

1.2. Darbo uždaviniai

Pagrindiniai darbo uždaviniai:

- Atrinkti ir ištirti neautomatiniam svetainių optimizavimui naudojamus rodiklius.
- Sudaryti atvirojo kodo sistemose įgyvendintų prognozavimo metodų rinkinį, tinkamą atrinktiems svetainių optimizavimui naudojamiems rodikliams.
- Ištirti atrinktų prognozavimo metodų tinkamumą svetainių optimizavimo uždaviniams spręsti.

1.3. Taikyti tyrimo ir analizės metodai

Pirmiausiai atrinkti 3 projektai: svetainės, esančios toje pačioje paskolų rinkoje ir optimizuojamos pagal tuos pačius raktinius žodžius: „Bobutės paskola“, „Paskoliukas“ ir „Popkreditas“.

Atrinkti laike kintantys duomenys, kurie svarbūs svetainės optimizacijos procese: puslapio užkrovimo laikas, atmetimo rodiklis, vidutinė vizito trukmė, išorinės nuorodos, socialinių profilių rodikliai. Šie duomenys reikšmingi norint pagerinti svetainę bei turėti geresnes pozicijas „Google“ organinėje paieškoje. Tai lemia didesnę svetainės lankytojų srautą, didina konversijų skaičių ir žinomumą. Taip pat, atrinkti galutiniai vertinimo rodikliai: svarbių raktinių žodžių pozicijos, svetainės lankytojų srautas, konversijos.

Duomenų rinkimui buvo naudojami įrankiai: „Ahrefs“, „Google Analytics“, „Moz“ įskiepis, „AWR“, socialiniai „Facebook“ ir „Google Plus“ profiliai.

Rodiklių prognozavimui buvo naudojami signalų analizės metodai, įgyvendinti MATLAB® aplinkoje: tiesinės prognozės metodas (LPC), autoregresinis paslinktųjų vidurkių prognozavimo metodas (ARMA), nelineijinis autoregresinis neuronų tinklas (NAR), nelineijinis autoregresinis su papildomu įėjimu neuronų tinklas (NARX), laiko vėlinimo neuronų tinklas (TDNN). Naudojami MATLAB® įrankių rinkiniai: sistemos identifikavimo, signalų apdorojimo, neuronų tinklų ir ekonometrikos.

1.4. Darbo naujumas ir praktinė nauda

Visos konversijų ir srauto didinimo priemonės yra skirtos rinkti bei saugoti informaciją apie lankytojo elgesį tinklapyje. Yra net kelios priežastys, dėl kurių yra naudojamos konversijų ir srauto didinimo technologijos.

Reklamos kompanijos aktyviai renka ir kaupia informaciją apie vartotojus, jų sukurtuose profiliuose, individualiai reklamai pritaikyti. Užuo rodžius atsitiktinai parinktus vartotojams reklaminius skelbimus, apie juos sukaupia profilio informacija yra panaudojama parinkti kuo tinkamesnį reklaminių skelbimą.

Teisėsaugos ir žvalgybos agentūros gali naudoti šias technologijas šnipinėti tam tikrus asmenis ir nagrinėti nusikaltimus. Taip galima išvengti kreditinių kortelių naudojimo sukčiavimo.

Testuoti internetinių aplikacijų patogumą. Stebint, analizuojant tam tikrus lankytojo veiksmus, galima pagerinti tinklalapio praktiškumą.

Operatoriai elektroninės komercijos versle, naudoja žiniatinklio analizę norėdami padidinti pelną. Pavyzdžiui įvertinti, kurie interneto puslapiai generuoja daugiausiai pajamų, kurie vaizdiniai skelbimai pritraukia daugiausiai lankytojų srauto, ar patiria daugiausiai nuostolių.

Nesvarbu, ar tai būtų teisėsaugos agentūra, ar internetinė prekyba, visoms yra svarbu didinti konversijas, analizuoti srautą ir rinkti informaciją, norint išsilaikyti šiuolaikinėje internetinėje rinkoje, užtikrinti saugumą, stabilumą ir augimą [2].

1.5. Darbo struktūra

Darbą sudaro neautomatinių svetainių optimizavimo sprendimų analitinė apžvalga, kurioje pateikti konversijų ir srauto didinimo būdai, interneto svetainių rodiklių analizavimo įrankiai: „Piwik“, „OWA“, „GWS“, „GA“, matavimo ir vertinimo rodikliai. Darbe pateikti atvirojo kodo sistemose įgyvendintų prognozavimo metodai, jų raida, prognozavimas laiko srityje ir klasikinių prognozavimo metodų analizė: LPC, ARMA, NAR, NARX, TDNN. Svetainės įvertinimui naudojama priežastinių ir pasekminių rodiklių atranka. Pateiktas nurodytų prognozavimo metodų efektyvumo tyrimas, analizė ir rezultatai.

2. NEAUTOMATINIŲ SVETAINIŲ OPTIMIZAVIMO SPRENDIMŲ ANALITINĖ APŽVALGA

Kuriant tinklalapį ir norint pritraukti daugiau lankytojų, pardavimų bei konversijų svarbu ne tik gražus ir modernus dizainas, bet ir turinys, žinomumas, funkcionalumas bei praktiškumas.

Interneto svetainių analizė – matavimas, duomenų rinkimas, analizės ir internetinių duomenų ataskaitos, reikalingos geriau suprasti ir optimizuoti interneto tinklalapį. Tai ne tik interneto srauto matavimas, bet gali būti naudojamas kaip verslo įrankis, rinkos paieškoms ir kaip prieiga pagerinti tinklalapio efektyvumą. Galimi du analizės tipai – vidinė ir išorinė.

Išorinė analizė – tai internetinės svetainės rodiklių matavimas ir analizė nepriklausomai nuo to ar patys kuriame, ar tik turime sukurtą svetainę. Ši analizė įtraukia potencialios auditorijos matavimus, matomumo didinimą, komentarus dalinimą.

Vidinė analizė – tai vartotojų elgesys tinklalapyje. Šiai kategorijai priskiriami techniniai svetainės aspektai, konversijų ir kitų rodiklių matavimas. Atvirojo kodo įrankiai su kuriais galima pamatuoti šiuos rodiklius: „Google Analytics“, „Open Web Analytics“ ir panašiai. Yra ir kitų įrankių, dažniausiai mokamų, kurie leidžia pridėti papildomų sluoksnių, naudojant A/B testavimą (dviejų tinklalapių versijų rodymas skirtingiems lankytojams), „Heat Maps“ testavimą (tinklalapio paspaudimų intensyvumo žemėlapis), sesijos atkartojimą ir panašiai.

Šiame darbe bus kalbama apie vidinę internetinių tinklalapių analizę ir įrankius bei jų gaunamus rodiklius.

2.1. Konversijų ir srauto didinimo būdai

Norint padidinti srautą ir konversijų skaičių, svarbiausia sukurti tinkamą, patrauklų, patogų tinklalapį, jį optimizuoti, testuoti bei tobulinti.

Kuriant tinklalapį svarbu jo sudėtingumas. Interneto puslapis turi būti gana aiškiai suprantamas, kad lankytoją verstų kuo mažiau papildomai mąstyti. Pagrindiniai aspektai, kurie verčia be reikalo gaišti laiką ir mąstyti yra tinklalapyje esami mygtukai, kurie kartais nebūna išskirti kaip mygtukai, sunku suprasti, ar juos galima spausti. Taip pat neaiškūs paieškos laukeliai su daugybe ilgų dviprasmiškų pasirinkimų. Pagrindiniai aspektai turi būti išskirti, nes dažniausiai lankytojas neskaito visos puslapio informacijos, o turi savybę „skenuoti“ jį. Daug sunkios ir dviprasmiškos informacijos taip pat kelia problemų, nes žmonės dažniausiai skuba, neturi laiko viską perskaityti ar atrinkti reikiamos informacijos, o ypač laukti kol atsivers puslapis.

Taip pat labai svarbu yra aiški tinklalapio hierarchija. Svarbiausi dalykai turi būti aiškiai pažymėti pradžioje. Loginės sąsajos taip pat yra geriausiai matomos vizualiai. Atėję lankytojai, kaip ir į parduotuvę, turi suprasti, kur jiems eiti toliau, kaip grįžti, kaip rasti reikiamą informaciją ir kur

ieškoti pagalbos, jei kažko nerado. Dėl šios priežasties yra reikalingas aiškus mygtukas, skirtas grįžti į pradinį puslapį bei paieškos laukelis, jei kažko nepavyko rasti [3].

Ilgos iškalbos ir instrukcijos turi būti vengiamos, nes lankytojas, kuriam yra reikalinga konkreti informacija, tikrai negaiš savo laiko skaitant gražias kalbas ir padėkas.

Sukūrus teisingą tinklalapį, kitas žingsnis, yra duomenų rinkimas ir analizė. Analizuojant interneto puslapius reikia atlikti tinklalapio lankomumo ir naudojimo matavimus bei atlikti duomenų analizę. Ši informacija yra svarbi norint suprasti, kas veikia geriausiai, ką reiktų keisti ar testuoti toliau. Svarbūs matavimo duomenys yra:

- Apsilankymų skaičius, kuris vėliau gali būti išskiriamas į pakartotinai apsilankiusius vartotojus ir apsilankiusius pirmą kartą. Pagal šią informaciją galima spręsti, kiek unikalūs lankytojai praleidžia laiko tinklapyje ir kurie puslapiai yra labiausiai patrauklūs.
- Kaip lankytojai randa tinklalapį ar sužino apie jį. Dažniausi šaltiniai yra: tiesioginis atėjimas įvedus adresą, tiesioginės nuorodos iš kitų tinklalapių, paieškos variklis.
- Marketingo kompanijos efektyvumas, matuojant kiek srauto atveda reklamos kompanijos.
- Geografinė lankytojų padėtis, kuri dažniausiai identifikuojama pagal IP adresą.
- Kompanijų identifikavimas, pavyzdžiui, kiek konkurentų stebėjo tinklalapį. Šie duomenys dažniausiai grindžiami taip pat IP adresu.
- Įvairios techninės detalės: operacinė sistema, vaizdo skiriamoji geba ir lankytojų naudojamos interneto naršyklės.

Duomenų sekimo ir matavimo įrankiai taip pat yra svarbūs praktiškumo atžvilgiu, kurie padeda surasti sprendimą dizaino klausimais ir vystymo procesams.

- Svarbu atsekti lankytojo kursoriaus judesius, kad būtų galima įvertinti, ar jam sunku naršyti svetainėje, ar jis turi problemų su tam tikromis tinklalapio funkcijomis. Ši informacija gali būti apibendrinta į aukštesnio lygio informaciją apie tam tikrus puslapio elementus, tokius kaip mygtukai ar slinkties juosta, vietoje kursoriaus koordinatų.
- Lankytojo užklausų pildymo eiliškumas, laiko trukmė kiekviename žingsnyje, gali padėti suprasti, kuris klausimas sunkiausias, keliantis daugiausiai problemų ir kaip jie gali būti pakeičiami.
- Surinkti duomenys gali būti naudojami lankytojo žinių apie kompiuterį klasifikavimui, taip pritaikant patį puslapį. Pavyzdžiui, sudėtingesni pasirinkimai, opcijos, gali būti skiriamos tik patyrusiems vartotojams, kitiems – supaprastintos versijos su pasiūlymais bei užuominomis [3].

Taigi, tam, kad pamatuoti lankytojų elgesį internete ir pagerinti lankomumo ir konversijų rodiklius yra naudojami įvairūs įrankiai. Pagrindiniai atvirojo kodo įrankiai yra „Open Web Analytics“, „Piwik“ ir „Google Analytics“. Šios priemonės skirtos geriau suprasti lankytojų elgesį,

dėl kokių priežasčių jie ateina į tinklalapį, ko jame ieško. Tai reikalinga norint geriau pamatuoti ir įvertinti lankytojų elgesį bei siekti išsikeltų verslo tikslų.

2.2. Interneto svetainių rodiklių analizavimo įrankiai

Pagrindiniai plačiai taikomi atvirojo kodo interneto svetainių rodiklių analizavimo įrankiai: „Piwik“, „Open Web Analytics“, „Grape Web Statistics“ ir „Google Analytics“. Plačiausiai taikomas yra „Google Analytics“, nes šis įrankis turi labai patogią ir paprastą vartotojo sąsają, yra „Google“ produktas ir lengvai siejamas su kitais „Google“ produktais. Šiame poskyryje bus plačiau aprašytas kiekvienas iš išvardintų įrankių, jų trūkumai ir privalumai.

2.2.1. Atvirojo kodo interneto svetainių rodiklių matavimo įrankis „Piwik“

Vienas geriausių, populiariausių atvirojo kodo interneto svetainių analizavimo įrankių yra „Piwik“. „Piwik“ turi daugiausiai pasirinkimų bei funkcijų. Šis įrankis puikiai integruojamas su bet kuriuo tinklalapiu. Populiarūs įrankiai „WordPress“ ir „Joomla“ turi sąsają su „Piwik“ bei visi tinklapiai gali būti lengvai adaptuojami, pridedami. „Piwik“ gali būti lengvai tobulinamas ir kiekvieną kartą esant kokiam nors įrankio atnaujinimui yra įspėjama. Atnaujinti galima vieno mygtuko paspaudimu. Tai suteikia galimybę turėti naujausias funkcijas ir išvengti įvairių programos „skylių“. Viena iš geresnių „Piwik“ savybių – tai, kad jis leidžia nusistatyti tikslus ir jų siekti. Tai galima įgyvendinti su daugybę skirtingų parametrų. 2.1 paveiksle pateikta „Piwik“ naudotojo grafinė sąsaja [4].



2.1 pav. Atvirojo kodo žiniatinklio analizės įrankis „Piwik“ [4]

2.2.2. Atvirojo kodo interneto svetainių rodiklių matavimo įrankis „Open Web Analytics“

„Open Web Analytics“ yra dar kitaip vadinamas trumpiniu OWA. Šis įrankis yra kur kas galingesnis nei „Grape“. Programinės įrangos funkcijos yra kur kas geresnės, grafinėje sąsajoje yra pateikiama žymiai daugiau statistinių duomenų, grafikų. Taip pat, OWA programinė įranga pateikia daug skirtingų statistikų, kurios gali būti peržiūrimos: naršyklės tipas, geografinė lokacija, geriausi puslapiai, paieškos raktiniai žodžiai, paieškos varikliai, reklamos skelbimo rodikliai ir panašiai. OWA taip pat suteikia galimybę peržiūrėti grafikus, kuriuose reprezentuojami įvairūs statistiniai duomenys. Duomenys yra rodomi gana nauji, iki kelių dienų senumo. Didžiausias šio įrankio pliusas – visi duomenys yra pateikiami vizualiai, įvairiais grafikais, todėl informacija yra labiau prieinama ir suprantama. 2.2 paveiksle pateikiama pagrindinio lango dalis, kuriame matosi statistiniai duomenys bei grafikai [4].

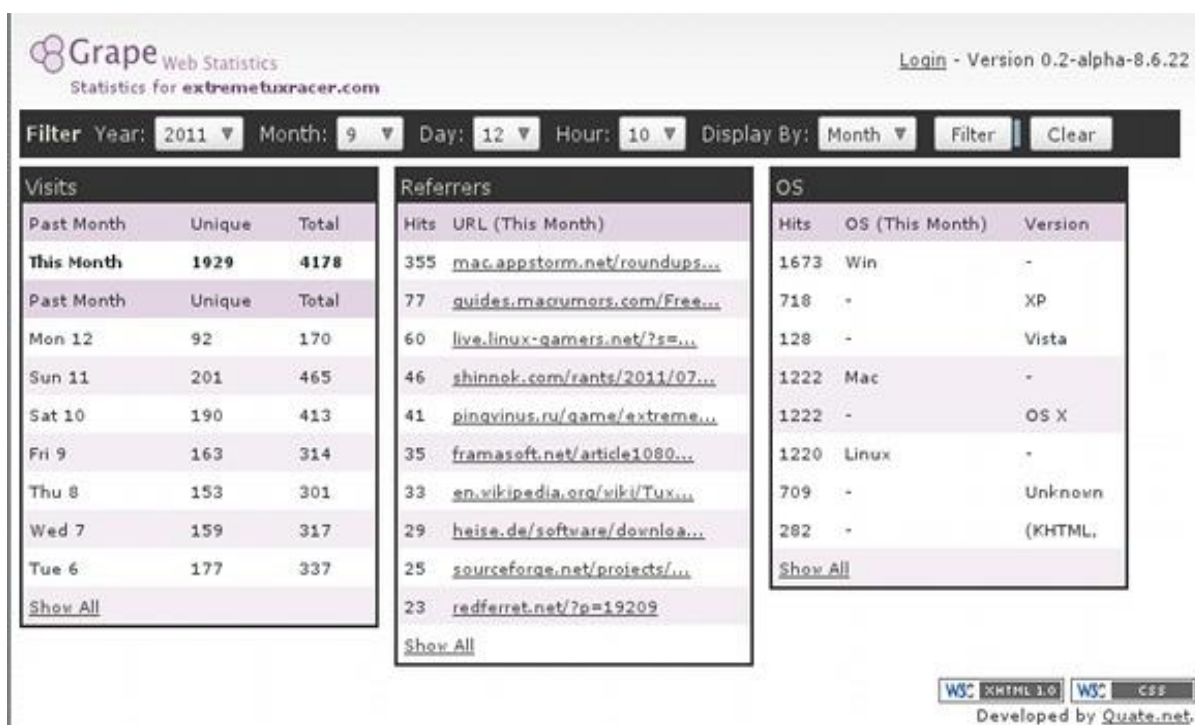


2.2 pav. Atvirojo kodo internetinių svetainių analizės įrankis „Open Web Analytics“

Taigi, OWA yra itin informatyvus bei gana paprastas ir suprantamas įrankis daugiau ar mažiau pažengusiems vartotojams [4].

2.2.3. Atvirojo kodo interneto svetainių rodiklių matavimo įrankis „Grape Web Statistics“

„Grape Web Statistics“ yra vienas paprasčiausių analizės įrankių. „Grape“ labiausiai skirtas tiems, kuriems „Google“ siūlomi įrankiai yra per daug neįveikiami ir nori, kad viskas būtų kur kas paprasčiau. Šis įrankis yra labai paprastos struktūros, bet jei vartotojas išmano HTML ir PHP, turi galimybę prasiplėsti programinę įrangą, kad atitiktų asmeninius reikalavimus. 2.3 paveiksle pavaizduotas „Grape“ pagrindinis langas su labai aiškia ir paprasta struktūra, kur yra galimybė pasirinkti metus, mėnesį, dienas, valandas ir matyti rezultatus pagal apsilankymus, nuorodas ir panašiai [4].



2.3 pav. Atvirojo kodo interneto svetainių analizės įrankis „Grape Web Statistics“ [4]

„Grape“ išskiria informaciją, statistiką, kuri yra aktuali tinklapiui. Viena iš neigiamų savybių yra ta, kad naujos versijos yra išleidžiamos dažniausiai vieną kartą per metus, bet tai nėra labai svarbu, jei vartotojo poreikiai yra minimalūs ir jiems nereikia labai daug ir naujausių galimybių bei pasiūlymų.

2.2.4. Atvirojo kodo interneto svetainių rodiklių matavimo įrankis „Google Analytics“

„Google Analytics“ (GA) yra įrankis, siūlomas „Google“ kompanijos, kuris generuoja inofmatyvias statistikas apie apsilankymus tinklalapyje. Šis įrankis yra nemokamas ir visiems prieinamas. GA gali atsekti visus apsilankymus per nuorodas, paieškos variklius, turinio tinklo reklamas, „Google AdWords“ reklamas ir panašiai. „Google Analytics“ yra susietas su „Google AdWords“.

„Google Analytics“ įdiegiamas pažymint puslapį (angl. *page tags*). Tai vadinamas „Google Analytics“ sekimo kodas, parašytas „JavaScript“, kurį tinklapio savininkas turi prisidėti prie kiekvieno tinklapio puslapio. Šis kodas veikia kliento naršyklėje, kai klientas naršo tinklalapyje (jei „JavaScript“ yra prieinama naršyklei) ir renka duomenis, siunčia juos į Google duomenų rinkimo serverį, kaip dalį užklauskos taip vadinamam „web beacon“. Tada parsisiunčiama „JavaScript“, rinkmena iš „Google“ interneto serverio ir nustato kintamuosius kartu su naudotojo paskyros numeriu. Didžiausia rinkmena (žinomas kaip *ga.js*) dažniausiai yra 18 KB. Rinkmena dažnai nėra atveriamas, dėl naršyklės spartumo. Ji būna atveriamas vieną kartą vizito pradžioje. Tada visi tinklapiai kurie turi įdiegtą GA kartu su „ga.js“ kodu naudoja tą pačią pagrindinę rinkmeną iš

„Google“. Naršyklė kurioje buvo atvertas tinklalapis su „Google Analytics“, jau turi įdiegtą rinkmeną. Vėliau, perduodant duomenis į „Google“ serverį, yra nustatomi „sausainėliai“ (angl. *cookies*) kiekvienam lankytojo kompiuteriui. Šie „sausainėliai“ saugo anoniminę informaciją, pavyzdžiui, kur lankytojas buvo prieš apsilankydamas tame tinklapyje, laiko žymę apie dabartinę vizitą ir nuorodos per kurias lankytojas atėjo į tą puslapį: paieškos varikliai, raktiniai žodžiai, reklamjuostės (angl. *banners*), elektroninis paštas ir t. t. 2.4 paveiksle pateikta „Google Analytics“ naudotojo sąsaja.



2.4 pav. Atvirojo kodo internetinių svetainių analizės įrankis „Google Analytics“ [4]

2.3. Atvirojo kodo įrankių interneto svetainės matavimo ir vertinimo rodikliai

Kuriant ir optimizuojant svetainę svarbu ne tik gražus, modernus dizainas, bet labai aktualus yra tinklapių turinys, navigacija, kaip jis yra pateikiamas ir patogus lankytojui. Optimizuojant šiuos aspektus, galima sumažinti rinkodaros išlaidas, arba tikslingiau jas paskirstyti, sulaukti lankytojų, klientų pasitenkinimą, pagerinti savo tinklalapį (tinklapių turinį, valdymą) ir sulaukti didesnės investicijų grąžos (IG). Visos šios aktualios yra pateiktos 2.5 paveiksle.



2.5 pav. Interneto svetainės pagrindiniai analizės aspektai [5]

Iš 2.5 paveikslo matyti, kad visi aspektai yra vienodai svarbūs. Pagerindami tinklalapį sulauksime lankytojų pasitenkinimo, sumažinsime rinkodaros išlaidas ir sulauksime didesnio pelno.

Esminiai rodikliai, kuriuos galima pamatuoti naudojant atvirojo kodo įrankius:

Esminiai rodikliai: vizitų skaičius, vidutinė vizito trukmė, atmetimo rodiklis (angl. *bounce rate*) nurodo, kuri dalis lankytojų paliko svetainę neatlikę daugiau jokio veiksmo.

Srauto šaltinių informacija: nauji lankytojai – kiek iš visų atėjusių lankytojų apsilankė pirmą kartą, sugrįžę lankytojai – kiek iš visų atėjusių lankytojų apsilankė daugiau nei vieną kartą, lankytojai atėję per nuorodas iš kitų tinklapių ir atėję pasinaudoję paieškos variklius, kaip „Google“, „Bing“, „Yahoo“ ir kt.

Pačių nusistatyti rodikliai (konversijos), kiek pateiktų užsakymų, užpildytų formų, kiek lankytojų užsiregistravo ir pan. [8].

Pagrindiniai svetainės aspektai, kuriuos galima įvertinti matuojant rodiklius:

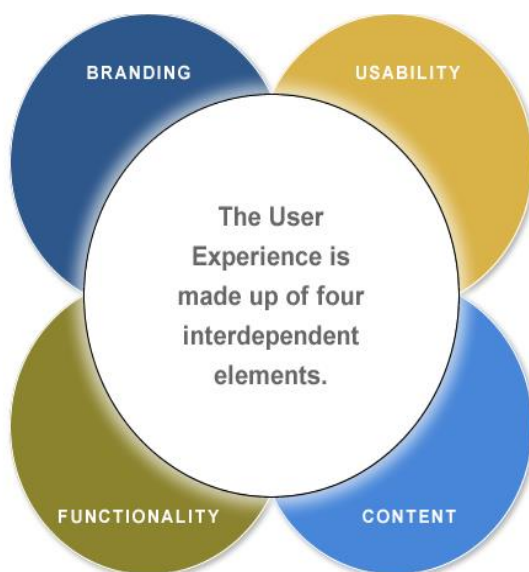
Žinomumas (angl. *branding*). Tinklalapio žinomumą labiausiai galima įvertinti matuojant, kiek lankytojų vėl sugrįžta į tinklapį, kiek laiko jame praleidžia. Jei tinklapyje gausu interaktyvių terpių, svarbu, kiek lankytojų jau kartą jas išbandę sugrįžta vėl.

Funkcionalumas (angl. *functionality*). Funkcionalumas priklauso nuo puslapių atvertimo laiko, tinklalapio atsako laiko. Taip pat svarbu įvertinti, kiek atvertimo metu būna serverio, tinklapio klaidų.

Praktiškumas (angl. *usability*). Svarbu žinoti, kiek tinklalapis yra praktiškas ir naudingas, o tai padaryti galima įvertinus, koks yra atmetimo rodiklis, kurioje vietoje lankytojas nutraukia naršymo, užsakymo ar registravimosi procesą. Norint pagerinti tinklalapio praktiškumą, svarbu išvengti

rinkmenų, serverių klaidų, o klaidoms įvykus puslapis turi būti atstatomas. Svetainė turi padėti lankytojams pasiekti jų tikslus, dėl kurių jie ir atėjo į tinklalapį.

Turinys (anlg. *content*). Svetainėje svarbu, koks ir kaip yra pateiktas turinys. Jis turi būti aiškiai suprantamas, atitikti puslapio temą. Įvertinti puslapio turinį galime pagal atmetimo rodiklį, vidutinę peržiūros trukmę, paieškų skaičių. Kuo šie rodikliai geresni, tuo tinklapio turinys labiau atitinka lankytojo tikslus ir poreikius, jis gauna tai, ko ieškojo. Kitu atveju, reiktų apgalvoti, ką galima pakeisti, ar išvis atsisakyti, kad lankytojas nebūtų klaidinamas [7]. Visi šie aspektai pavaizduoti grafiškai 2.6 paveiksle.



2.6 pav. Pagrindiniai analizės aspektai vertinant internetinę svetainę [6]

Iš 2.6 paveikslo matyti, kad visi šie aspektai yra įvardinti kaip individualūs ir taip pat priklausomi nuo naudotojų. Jie labai svarbūs tiek tinklalapio atžvilgiu, tiek pritraukiant daugiau potencialių lankytojų, klientų, tiek didinant žinomumą ir gerą tinklalapio vardą.

2.4. Skyriaus apibendrinimas

Neautomatinių svetainių optimizavimo du pagrindiniai tipai: išorinė ir vidinė. Išorinė optimizacija labiausiai orientuota į srauto didinimą, nuorodų tinklo plėtimą. Vidinė optimizacija – svetainių struktūros gerinimas tam, kad daugiau pritraukti potencialių klientų, didinti konversijų skaičių. Optimizuojant svetainę yra renkami duomenys, pagal kuriuos daromos išvados, priimami sprendimai.

Norint padidinti srautą ir konversijas svetainėje, svarbu ją supaprastinti, padaryti kuo lankstesne, patrauklesne ir aiškesne lankytojiui. Svetainės struktūra ir navigacija turi būti lengvai suprantama ir įsisavinama.

Matuojant duomenis svarbu įvertinti apsilankymų skaičių, šaltinius, reklaminių kampanijų efektyvumą, geografinę padėtį, konkurentus, lankytojų kursoriaus judesius ir pan.

Duomenų matavimą ir rinkimą supaprastina žinomi atvirojo kodo, nemokami įrankiai, kuriuos galima pasirinkti pagal kiekvieno poreikius: „Piwik“, „Open Web Analytics“, „Grape Web Statistics“ ir „Google Analytics“.

„Piwik“ – gana populiarius ir plačiai naudojamas įrankis, turintis nemažai funkcijų, lengvai pritaikomas ir susiejamas su kitais įrankiais, plačiai taikomas.

„Grape Web Statistics“ – paprastas, skirtas tiems, kuriems kiti įrankiai yra per daug sudėtingi. Šis įrankis labai paprastas, bet jei naudotojas išmano HTML ir PHP, galima lengvai prasiplėsti įrankio galimybes.

„Open Web Analytics“ (OWA) – turi daugiau funkcionalumo nei įrankis „Grape Web Statistics“. OWA turi patogią naudoto ir grafinę sąsajas, duomenys pateikiami iki kelių dienų senumo.

Populiariausias įrankis – „Google Analytics“ yra „Google“ nemokamas produktas, kuris yra lengvai prieinamas, renka kur kas daugiau duomenų, lengvai susiejamas su „Google AdWords“ ir kitais „Google“ produktais. Pateikia naujausius tos dienos duomenis.

Svarbiausi matavimo rodikliai: vizitų skaičius, vidutinė vizito trukmė, atmetimo rodiklis, naujų lankytojų skaičius, konversijų rodiklis (užsakymai, registracijos, pirkimai). Pagrindiniai svetainės vertinimo aspektai: žinomumas, funkcionalumas, praktiškumas ir turinys.

3. ATVIROJO KODO SISTEMOSE ĮGYVENDINTŲ PROGNOZAVIMO METODŲ ANALIZĖ

Šiame skyriuje bus apžvelgta prognozavimo metodų taikymo raida, taikymo sritys, svarba. Bus apžvelgtas prognozavimas laiko srityje, kuris iki šiol yra labai svarbus uždavinys moksle ir inžinerijoje. Ateities prognozavimas buvo ir yra svarbi užduotis daugelyje sričių, todėl bus pateikti pagrindiniai metodai, apžvelgti pagrindiniai parametrai, veikimo principai bei realūs jų pritaikymai moksle, buityje, inžinerijoje ir pan. Didžiausias dėmesys bus atkreiptas į tuos modelius, kurie taikomi tolimesniame darbe, atliekant tyrimą, tai: autoregresinis slenkamųjų vidurkių – ARMA, tiesinės prognozės metodas – LPC ir metodai, grįsti dirbtiniais neuronų tinklais: netiesinis autoregresinis neuronų tinklas – NAR, netiesinis autoregresinis su papildomų įėjimu neuronų tinklas – NARX bei laiko vėlinimo neuronų tinklas – TDNN.

3.1. Prognozavimo metodų taikymo raida

Šiomis dienomis nuožmi konkurencija tarp individualių žmonių ir organizacijų yra modernios visuomenės skiriamasis ženklas, kurioje pasiekti strategijos privalumai gali atnešti didžiulę sėkmę. Galimybė prognozuoti ateitį, remiantis praeities duomenimis, suteikia didelį svarumą organizacijoms judėti į priekį. Prognozavimas laiko srityje nuosekliai kintančių duomenų yra labai svarbus aspektas, kurio pagrindinis tikslas yra nuspėti elgseną pagal sudėtingos kompleksinės sistemos praeities duomenų šabloną. Didelį indelį šioje srityje įnešė veiklos tyrimai, statistika ir kompiuterių mokslas, kurie padėjo pakeisti intuiciją į rimtus prognozavimo mokslus, kaip „Holt-Winters (Winter, 1960)“ ir „Box – Jenkins“ (1976 m.) metodologijos. Šie modeliai buvo sukurti prieš dešimtmečius, kai skaičiavimo galimybės dar buvo ribotos. Šiais modeliais buvo galima tiksliai apskaičiuoti tiesiškai laike kintančius duomenis, tačiau modeliai turėjo didžiulį trūkumą skaičiuojant netiesiškas komponentes (Shoneburg, 1990 m.), kurios daugiau vyrauja realaus pasaulio situacijose (pavyzdžiui, finansinių duomenų analizėje).

Alternatyvus požiūris į prognozavimo laiko srityje metodus atsirado pradedant naudoti dirbtinių neuronų tinklus. Tokiam jungiamajam modelių kūrimui įtaką padarė centrinės nervų sistemos elgsena, kuri buvo patraukli savo intelektualia sistema, renkant bei valdant duomenis. Daugiasluoksnis perceptronas yra populiariausia neuronų tinklo architektūra, kur neuronai yra sugrupuoti į sluoksnius, o tarp jų galioja tik pirminis ryšys. Ši sistema save apsimoko, pagrindiniai privalumai: netiesinis apsimokymas ir tolerancija triukšmui. Šio daugiasluoksnio perceptrono naudojimas laike kintantiems modeliams atsirado vėlyvaisiais 1980 metais ir nuo tada ši mokslo kryptis vystėsi labai sparčiai [9].

Susidomėjimas daugiasluoksniais perceptronais skatino atgalinio mokymo (angl. *backpropagation*) algoritmas, sukurtas 1986 metais, ir nuo tada buvo siūlomi keli greiti variantai, kaip RPROP algoritmas (Riedmiller, 1994 m). Šie mokymo metodai sumažina tikslo funkciją (grįstą klaidos skaičiavimu), nustatant kintamųjų parametrus (neuronų ryšių svorius) pagal prioritetą. Neuronų veikla būna jautri tinklo topologijos pasirinkimui: mažas tinklas suteikia ribotas mokymosi galimybes, o didesnis tinklas gali perpildyti mokymosi duomenis, o tai gali padidinti nuostolius.

Neuronų tinklo topologijos kūrimas yra sudėtinga užduotis, susijusi su mokymosi ir klaidų procedūromis (pavyzdžiui, bandant skirtingą paslėptųjų sluoksnių skaičių). Buvo pasiūlyti kiek tikslesni „pruning“ ir „constructive“ algoritmai, kurių pagrindinis privalumas – automatinis neuronų tinklo kūrimas. Bet šios „lipimo į kalną“ (angl. *hill-climbing*) procedūros iššaukia du didelius nuosmukius: jie gali turėti lokalius minimumus ir vykdo paiešką naudojant mažas architektūros porcijas, anot naudojus visą paieškos erdvę.

Taip pat, buvo pasiūlyta skirtinga alternatyva – evoliuciniai skaičiavimai (angl. *evolutionary computation*), kurie nurodo skaičiavimo procedūrų grupę pagal natūralios atrankos procesą. Evoliucinis algoritmas tinka optimizavimo užduotims, atliekant paiešką iš daugelio sudedamųjų dalių, greitai nustato aukštos kokybės vietas, nors paieškos laukas yra didelis ir sudėtingas. Hibridinė evoliucinio skaičiavimo kombinacija yra neuronų tinklai, dažnai yra vadinami – evoliuciniai neuronų tinklai (angl. *evolutionary neural networks*).

Prognozavimas laiko srityje yra gana naudinga metodų grupė, naudojama testuojant evoliucinio neuronų tinklo kombinacijas, kol neuronų darbas nepriklauso vien tik nuo vidinio tinklo parametrų, bet priklauso ir nuo tinkamo įėjimų parinkimo – „įėjimo vėlavimo“ (angl. *input lags*) [9].

3.2. Prognozavimo laiko srityje analizė

Prognozavimas laiko srityje, ypač kai jis yra ilgalaikis – labai svarbus uždavinys daugelyje inžinerijos ir mokslo sričių. Egzistuoja daugybė įvairiausių metodų prognozavimui laiko srityje, bet visų šių metodų pagrindinis tikslas yra sukonstruoti modelį pagal praeities duomenis ir tam tikrus parametrus ir jį pritaikyti paskutinių reikšmių ekstrapoliavimui į ateitį. Prognozavimo procedūras galima skirstyti į ne vieną kategoriją, pavyzdžiui: slenkančio vidurkio algoritmai, metodai paremti neuronais, neraiškos logikos tinklai ir kiti.

Duomenys, kintantys laiko srityje (angl. *time series*), yra rinkinys periodiškai išsidėsčiusių įėjimų $\{x_1, x_2, \dots, x_t\}$, pasireiškiančių daugelyje sričių, tokių kaip žemės ūkis, fizika, finansai, inžinerija ir kt. Taip pat ši analizė yra svarbi ir statistikoje. Šis modelis taikomas, atsižvelgiant, kad praeities duomenų šablonas pasikartos ateityje.

Prognozavimas laiko srityje yra deterministinis, jei prognozė yra įmanoma 100 %. Praktikoje dažniausiai būna stochastinių elementų, dėl kurių dažniausiai atsiranda prognozavimo paklaida.

Tikrinant modelio tikslumą yra skaičiuojama kvadratinės paklaidos suma (angl. *sum of squared errors* – *SSE*) ir vidutinės kvadratinės paklaidos šaknis (angl. *root mean squared error* – *RMSE*), kurių matematinės išraiškos pateikta (3.1) formulėje.

$$SSE = \sum_{i=t+1}^{t+L} (e_n^2), \quad (3.1)$$

čia $e_t = x_t - \hat{x}_t$ prognozės paklaida ir L – prognozavimo skaičius.

Prognozavimui laiko srityje yra naudojamas autokoreliacijos koeficientas, intervale $[-1; 1]$, kuriuo matuojama koreliacija tarp sekos ir jos pačios, pavėlintos k periodų. Matematinė išraiška pateikta (3.2) formulėje.

$$r_k = \frac{\sum_{t=1}^{s-k} (x_t - \bar{x})(x_{t+k} - \bar{x})}{\sum_{t=1}^s (x_t - \bar{x})^2}, \quad (3.2)$$

čia s yra eilės dydis, o $\bar{x}$ – vidurkis.

Autokoreliacija gali būti naudinga nustatant tendencijas ir sezoniškumo įtaką. Tendencija parodo duomenų konstantos augimą arba kritimą (pavyzdžiui, galima pokyčio priežastis – technologijų augimo infliacija). Sezoniškumo faktorius aptinkamas periodiniame kitime ir jis yra labai dažnas mėnesio periodiškume (pavyzdžiui, ledų pardavimas). Sezoniškumo faktorius didelę įtaką turi ir tiriant paskolų rinką, pavyzdžiui prieš Kalėdas ir kitas šventes paskolų imama, kur kas daugiau nei kitais mėnesiais ir dienomis [9].

Prognozavimo laiko srityje galimi pavyzdžiai: saulės dėmių aktyvumas, elektros poreikis mieste, gimstamumas visuomenėje, oro temperatūra pastate ir kt. [10].

3.3. Ateities prognozavimas

Ateities prognozę galima klasifikuoti į kelias klases (pavyzdžiui kai norime ką nors parduoti):

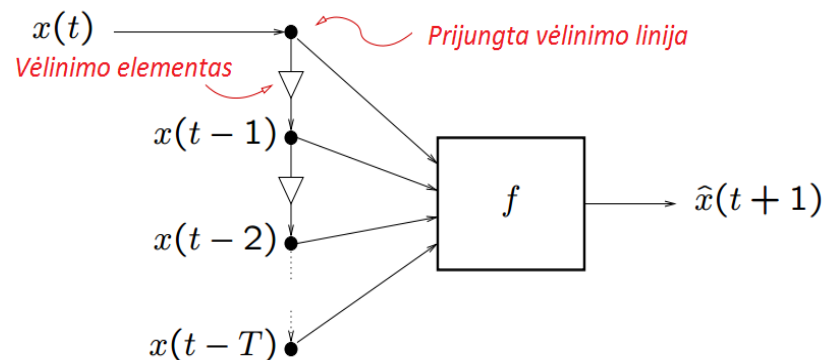
1. Kaina augs.
2. Kaina mažės – reikia parduoti dabar.
3. Nepasikeis.

Tariant, kad duomenų kitimo laike išraiška – $x(t)$, kur x – duomenys konkrečiu laiko momentu, t – konkreti laiko reikšmė, imant praeities laiko reikšmes, gauname eilutę $\{x[t], \widehat{x[t-1]}, \dots\}$. Pagal tai, išreiškiame tam tikros ateities reikšmės prognozės eilutę $\hat{x}[t+s] = f(\widehat{x[t]}, x[t-1], \dots)$, kur s yra prognozės horizontas. Tuomet iškyla funkcijos aproksimacijos problema, kuri sprendžiama:

1. Pritaikant atitinkamą modelį.

2. Kiekvienai praeities reikšmei $x[t_i]$, apmokomas modelis, kuriame nurodomos įėjimo reikšmės yra reikšmės, kurias tikimės gauti.

Bendroji prognozavimo schema, skirta naudojant praeities duomenys prognozuoti ateitį pateikta 3.1 pav.



3.1 pav. Ateities prognozavimo, naudojant praeities duomenys, bendroji funkcinė schema

Praeities duomenys apdorojami tam tikra funkcija, pritaikytu modeliu ir yra prognozuojama ateities reikšmė.

3.3.1. Autoregresinis slenkamųjų vidurkių prognozės metodas – ARMA

Tiesinės sistemos yra viena skaitmeninių signalų apdorojimo (angl. *Digital Signal Processing*) kryptų dažniausiai taikoma tiesiniams reiškiniams analizuoti, situacijose, kai tarp sistemos įėjimo ir išėjimo yra pastebima tiesinė priklausomybė. Filtravimas – viena pagrindinių SSA apdorojimų operacijų. Šioms operacijoms vykdyti naudojami filtrai – įtaisai, kurie pagal tam tikrus kriterijus atskiria tam tikras signalo dalis [12]. Operacijos atliekamos filtrais su įėjimo duomenimis $x[n]$, pateikia išėjimo $y[n]$ duomenis. Yra dvi pagrindinės filtrų architektūros: ribotos impulsinės reakcijos (angl. *finite impulse response – FIR*) ir beribės impulsinės reakcijos (angl. *infinite impulse response – IIR*).

SSA teorija siūlo 3 pagrindines tiesinės analizės metodų grupes:

- Autoregresiniai modeliai (angl. *autoregressive – AR*).
- Slenkamųjų vidurkių modeliai (angl. *moving average – MA*).
- Autoregresiniai slenkamųjų vidurkių modeliai (angl. *autoregressive moving average – ARMA*)

ARMA modelio matematinė išraiška pateikta 3.3 formulėje.

$$y_t = \alpha + \varphi_1 y_{t-1} + \dots + \varphi_p y_{t-p} + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t, \quad (3.3)$$

kur $\varphi_1 y_{t-1} + \dots + \varphi_p y_{t-p}$ yra AR procesas, o $\theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t$ – MA procesas.

Autoregresinis procesas aiškina prognozavimo laiko srityje stebėjimus ankstesniais momentais $Y_t = \varphi_1 Y_{t-1} + \varphi_2 Y_{t-2} + \dots + \varphi_p Y_{t-p} + \varepsilon_t$, kur Y_t prognozavimo laiko srityje stebėjimas, $\varphi_1 \dots \varphi_p$ –

autoregresinio proceso parametrai, ε_t – atsitiktinės paklaidos, o p – autoregresinio proceso eilė [13]. Autoregresijos $AR(p)$ modelyje, prognozavimo laiko srityje reikšmė išreiškiama kaip tiesinis darinys iš p ankstesnių ir atsitiktinio triukšmo reikšmių.

ARMA metodas buvo pritaikytas daugelyje prognozės bandymuose. Pavyzdžiui taikant ARMA metodą ir įvairias jo modifikacijas buvo prognozuojamas Austrijos infliacijos rodiklis, paklaida siekė iki 20 % [14].

Ispanijoje pritaikytas ARMA modelis prognozuoti vidutinį vėjo greitį per valandą. Dažniausiai naudojami ir populiariausi ARMA metodai yra, kurių modelių indeksai (1 1) ir (1 2). Vidutinė paklaida nėra nurodyta, tiesiog lyginama su kitais modeliais bei modifikacijomis [15].

Taikant ARMA ir ARIMA metodus buvo prognozuojamos elektros sąnaudos buityje. Mažiausia paklaida prognozuojant sąnaudas kasdien per 28 dienas – apie 29 %, 35 dienas – 32 %. Prognozuojant kas savaitę paklaida mažesnė – apie 18 %, kas mėnesį – iki 10 %. Kuo prognozuojamas ilgesnis laiko intervalas, tuo paklaida gaunama mažesnė [16].

3.3.2. Tiesinės prognozės metodas – LPC

Tiesinė prognozė (angl. *linear prediction coding* – LPC), yra vienas galingiausių, populiariausių signalų analizės metodų, matematiškai tiksliai interpretuojamas ir pakankamai lengvai realizuojamas programinėje įrangoje. Tiesinės prognozės metodo pagrindinė idėja yra prognozuoti kalbos signalo atskaitą remiantis kalbos signalo buvusiomis atskaitomis. LPC modelis remiasi prielaida, kad kalbos signalas $s(n)$ momentu n gali būti aproksimuotas prieš tai esančių p signalo reikšmių tiesine kombinacija [13].

$$s(n) \cong a_1 s(n-1) + a_2 s(n-2) + \dots + a_p s(n-p). \quad (3.4)$$

Koeficientai $a_1, \dots, a_p$ nagrinėjamame signalo intervale yra laikomi pastoviais. Formulės (3.4) dešinės pusės narys (pažymėsime jį $\tilde{s}(n)$) dar yra vadinamas tiesine prognoze:

$$\tilde{s}(n) = a_1 s(n-1) + a_2 s(n-2) + \dots + a_p s(n-p), \quad (3.5)$$

tai yra ankstesnių signalo reikšmių tiesinė kombinacija (todėl šis modelis ir yra vadinamas tiesinės prognozės kodavimo modeliu). Turėdami signalą $s(n)$ momentu n ir jo reikšmės prognozę $\tilde{s}(n)$, galime paskaičiuoti momentinę prognozės klaidą (žr. 3.6 formulę) [13].

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^p a_k s(n-k) \quad (3.6)$$

Pavyzdžiui, kai filtro eilė yra $p = 6$, signalo matematinė išraiška atrodys - $y = a_2s_1 + a_3s_2 + a_4s_3 + a_5s_4 + a_6s_5 + a_7s_6$. Koeficientai a turi būti parenkami taip, kad paklaida tarp realios ir prognozuojamos vertės būtų mažiausia.

Kaip jau buvo minėta, LPC dažniausiai naudojamas kalbos signalo atpažinimui [18]. Taip pat, gali būti pritaikytas triukšmo šalinimui iš elektros grandinių, įrenginių. Pagal atliktus tyrimus padaryta išvada, kad LPC metodas gali būti naudojamas triukšmo dažnių šalinimui, esantiems mažesniems nei 1 MHz [17].

3.3.3. Dirbtinis neuronų tinklas – DNT

Dirbtinis neuronų tinklas – tai struktūra sukurta remiantis biologinės nervų sistemos pagrindu, kuri skirta informacijos apdorojimui. Kaip ir nervų sistema, dirbtinis neuronų tinklas yra sudarytas iš daugelio tarpusavyje susijusių komponentų, iš kurių kiekvienas atlieka skaičiavimus – šie komponentai vadinami neuronais. Tam, kad neuronų tinklas išspręstų jam pateiktas užduotis, pirmiausia reikia jį apmokyti iš jau turimų analogiškų pavyzdžių. Šie turėtų būti labai tikslūs ir atitikti norimą rezultatą, kitaip tinklas gali veikti netiksliai. Tinklo apmokymas yra vykdomas mokymo metu keičiant tarp neuronų esančių jungčių svorius [19].

Neuronų tinklai pasižymi galimybėmis išvesti sudėtingų ir netikslų duomenų reikšmes, todėl jų panaudojimas tinkamas nustatant rinkos tendencijas ar sudarant sudėtingus modelius, kuriuos yra per sunku sudaryti žmogui ar su įprastinėmis kompiuterinėmis technologijoms. Apmokytas dirbtinis neuronų tinklas, gali būti laikomas kaip tos srities ekspertas. Ir juo galima pasinaudoti, kuriant ar ieškant naujų tendencijų, sprendžiant dilemą „kas, jeigu“ (angl. „*What if*“) [20].

Yra išskiriamos kelios neuronų tinklų savybės:

- Adaptyvus mokymas – gebėjimas išmokyti daryti užduotis, remiantis pateiktais duomenimis apie mokymus ar pradinę patirtį.
- Savi-organizacija – dirbtinis neuronų tinklas gali sukurti savo organizaciją ar reprezentaciją, naudojant informaciją, kurią gavo per mokymus – parinkti ryšių svorių reikšmes analizuojant tik tinklo įėjime paduodamus pavyzdžius.
- Realios laiko operacija – neuronų tinklo skaičiavimai gali būti atliekami lygiagrečiai, tai yra vienu metu, o skaičiavimų supaprastinimui yra kuriama ir gaminama specialiai suprojektuota kompiuterinė technika.
- Klaidų tolerancija – daliniai tinklo pažeidimai, tiesiogiai daro įtaką tinklo našumui ir duomenų tikslumui, tačiau kai kurios tinklo funkcijos vis dar gali būti atliekamos tiksliai.

Tiesioginio sklaidimo neuronų tinkluose, signalai sklinda tik viena kryptimi. T. y. tolimesnio sluoksnio neuronų išėjimai gauna informaciją tik iš prieš tai buvusių sluoksnių. Tokiuose tinkluose

nėra, nei grįžtamųjų ryšių iš ankstesnių sluoksnių, nei ciklą su tame pačiame sluoksnyje esančiais neuronais.

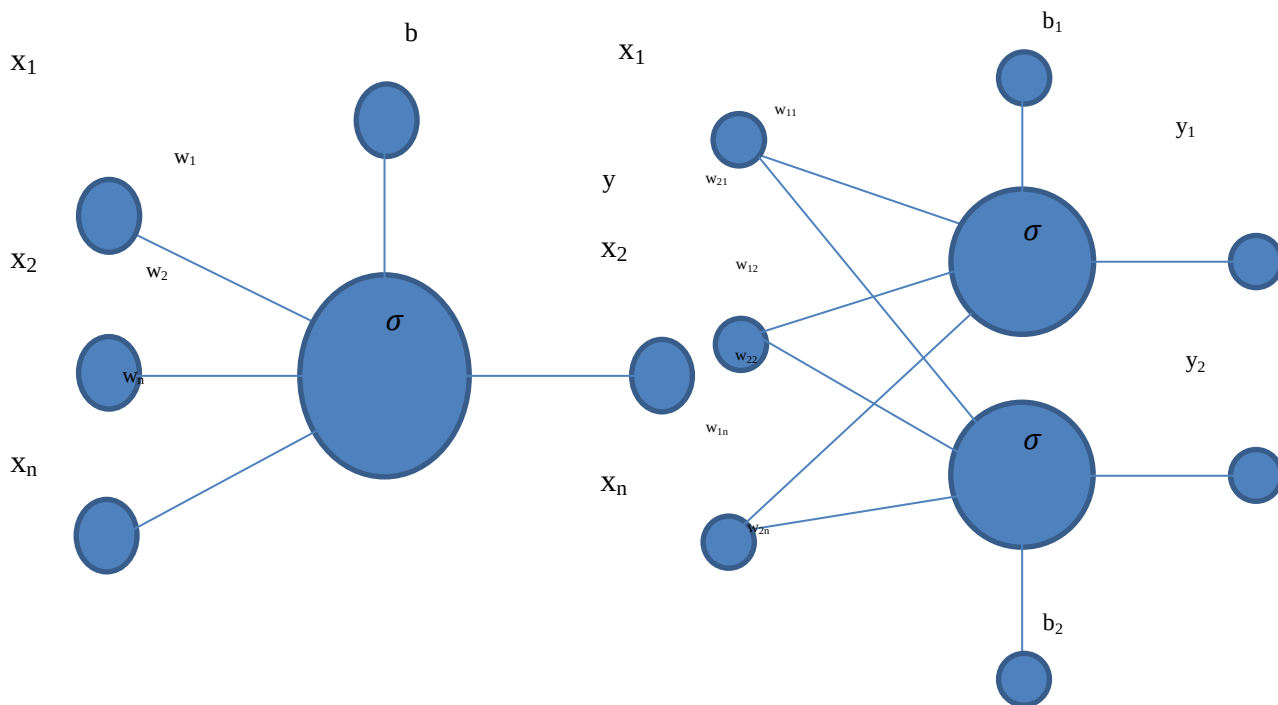
Galimi tiesioginio sklaidimo neuronų tinklai:

- Vienasluoksnis perceptrono tinklas – kai tarp įėjimo ir išėjimo sluoksnių nėra paslėptų neuronų sluoksnių.
- Daugiasluoksnis perceptrono tinklas – kai tarp įėjimo ir išėjimo sluoksnių yra vienas ar daugiau paslėptų neuronų sluoksnių.
- Spindulio tipo bazinių funkcijų tinklai – kai paslėptuose sluoksniuose, kaip aktyvavimo funkcija, yra naudojamos radialinės bazinės funkcijos.

Bendrinė vienasluoksnio perceptrono tinklo (3.7) formulė atitinkanti 3.2 paveikslą:

$$y = \sigma(x_1w_1 + x_2w_2 + \dots + x_nw_n + b), \quad (3.7)$$

čia y – išėjimo duomenys, x – įėjimo, w – atitinkami neuronų svoriai ir b – slenkstis. Bendrinės dirbtinių neuronų tinklų struktūros pateiktos 3.2 pav.



3.2 pav. Vienasluoksnis (kairėje pusėje) ir daugiasluoksnis (kairėje) perceptronų tinklas

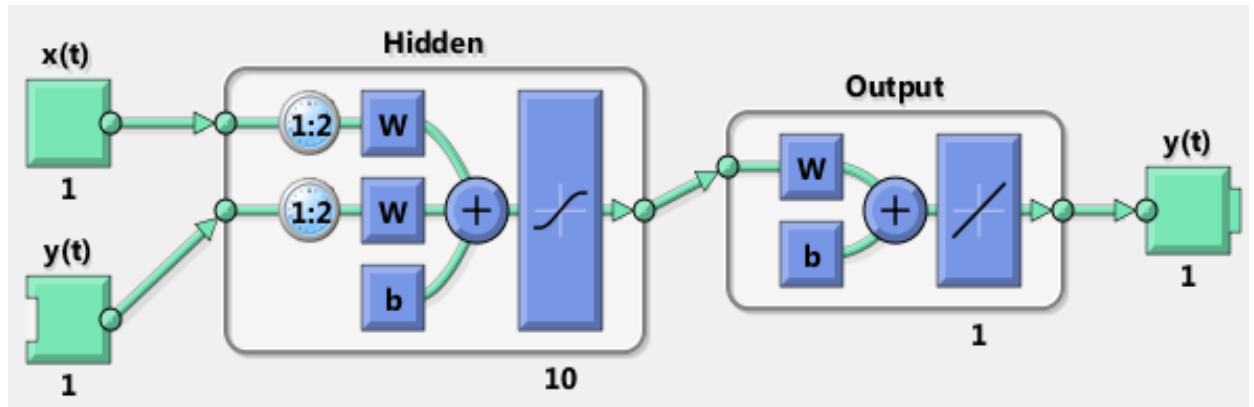
Norint prognozuoti laiko srities ateities vertes priklausomas nuo laiko srities praeities verčių y ir antrosios laiko srities praeities verčių x . Tokio tipo prognozės yra vadinamos netiesinės autoregresinės su išoriniais įėjimais arba trumpinama NARX, kuri aprašoma 3.8 formule:

$$y = f(y(t-1), \dots, y(t-n), x(t-n), \dots, (t-n)) \quad (3.8)$$

Šio tipo modelis gali būti naudojamas nuspėti ateities akcijų ar obligacijų vertes, remiantis tokiais ekonominiais rodikliais, kaip: nedarbo lygis, BVP ir t. t. Jis taip pat gali būti naudojamas

sistemos identifikavimui, kur modeliai yra išvystyti, jog atvaizduotų dinamines sistemas, tokias kaip cheminiai procesai, gamybos sistemos, robotų, kosmoso technikai ir t. t. [21].

NAR pavyzdys MATLAB® terpėje (žr. 3.3 pav.):



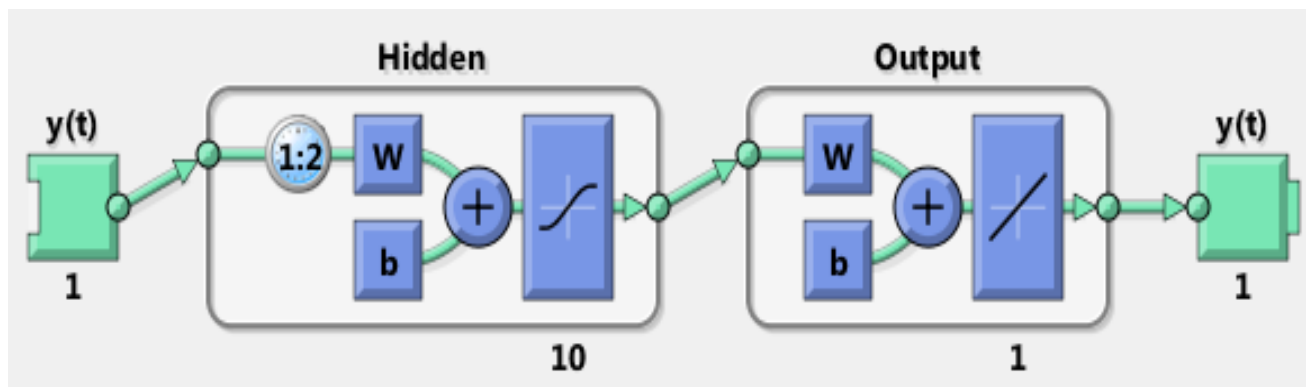
3.3 pav. Netiesinio autoregresinio su papildomu įėjimu neuronų tinklo grafinis pavyzdys [23]

3.3 paveiksle pateiktame NARX pavyzdyje neuronų skaičius – 10, įėjimo vėlinimas – 2, išėjimo vėlinimas – 2, du įėjimai – $x(t)$ ir $y(t)$, išėjimas – $y(t)$.

Kai laiko srities modelyje yra įtraukta tik viena sritis, o laiko srities y ateities vertės yra prognozuojamos naudojant tik tos srities praeities vertes, šio tipo prognozės metodas yra vadinamas netiesinis autoregresinis neuronų tinklas arba trumpinamas – NAR. NAR matematinė išraiška pateikta (3.9) formulėje [21].

$$y = f(y(t-1), \dots, y(t-n)) \quad (3.9)$$

Šis modelis gali būti naudojamas prognozuoti finansines priemones, nenaudojant papildomų įėjimo sričių. 3.4 paveiksle pateiktas MATLAB® naudojamas pavyzdys.



3.4 pav. Netiesinio autoregresinio neuronų tinklo grafinis pavyzdys [22]

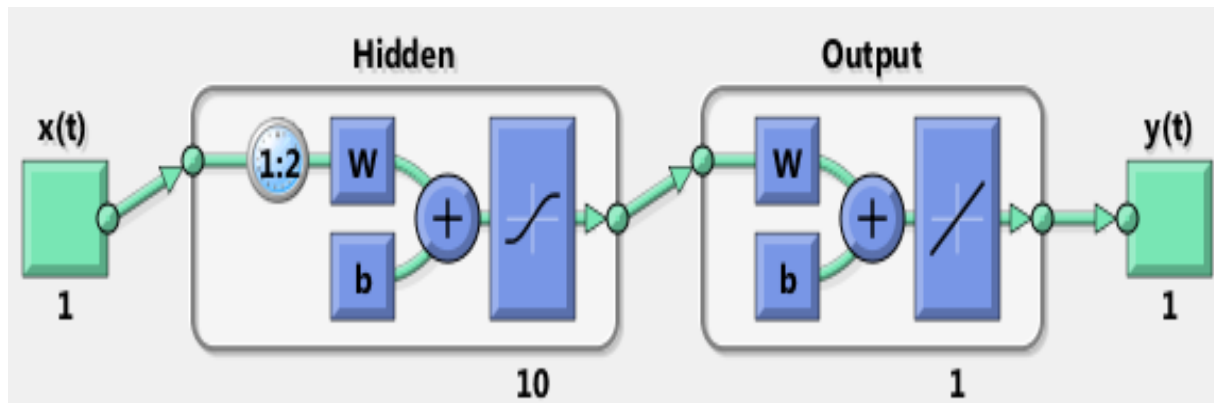
Pagal 3.4 paveiksle neuronų skaičius paslėptajame sluoksnyje – 10, o išėjimų vėlinimas – 2, įėjimas – $y(t)$, išėjimas taip pat – $y(t)$.

Dar vienas neuronų tinklo modelis – laiko vėlinimo neuronų tinklas – TDNN, yra panašus į pirmąją paminėtą, t.y. NARX, kur naudojamos dvi sritys: įėjimo sritis x ir išėjimo/tikslo sritis y . Šiame modelyje prognozės tikslas yra gauti duomenis y , kurie priklausomi nuo praeities x verčių,

tačiau be žinių apie y praeities vertes. Šis įėjimo/išėjimo modelis gali būti aprašytas 3.10 formulėje [21].

$$y(call) = f(x(t-1), \dots, x(t-n)) \quad (3.10)$$

TDNN grafinis pavyzdys pateiktas 3.5 paveiksle.



3.5 pav. Laiko vėlinimo neuronų tinklo grafinis pavyzdys [24]

Pagal 3.5 paveiksle pateiktą pavyzdį neuronų skaičius – 10, laiko vėlinimas 2, įėjimas – $x(t)$, išėjimas – $y(t)$.

Taigi, geresnis prognozės modelis galėtų būti NARX, nei šis išėjimo/įėjimo modelis, kuris naudoja papildomą informaciją, turi žinias apie y praeities vertės. Tačiau, gali būti atveju, kai negalima panaudoti praeities y reikšmių. Tai gali būti tada, kai norima panaudoti įėjimo/išėjimo modelį vietoje NARX modelio [21].

Buvo atliktas bandymas atliekant prognozę laiko srityje taikant NAR ir NARX metodus ir naudojant atraminių vektorių mašiną. Vidutinė paklaida, keičiant kintamuosius, naudojant NAR metodą buvo apie 33 %, o NARX metodą – apie 23 % [25].

Naudojant NAR, TDNN modelius ir jų hibridą buvo mėginta prognozuoti saulės radiaciją per valandą. Padarytos išvados, kad geriausiai veikiantis yra ARMA modelis yra (1 1). TDNN modelis kur kas jautresnis pokyčiams nei ARMA, bet nėra toks stabilus. Išmėgintas ARMA ir TDNN modelių hibridas. Hibridinis modelis prognozuoti tiesines komponentes naudojam ARMA, o netiesines komponentes TDNN. Hibridinis modelis veikia gerai tol kol yra stabilus ir tikslus [26].

3.4. Skyriaus apibendrinimas

Apžvelgus prognozavimų metodų raida galima daryti išvadą, kad ateities prognozavimas buvo svarbus klausimas ir užduotis jau ankstesniais laikais kai technologijos nebuvo dar tokios pažangios. Įvairūs veiklos tyrimai, statistika, kompiuterių mokslas padėjo pakeisti intuiciją į prognozavimo mokslą. Žymiausi pradininkai buvo „Holt-Winters“ ir „Box-Jenkins“, pagal kuriuos buvo sukurtos pirmosios metodologijos, su dar ribotomis skaičiavimo galimybėmis. Apžvelgta neuronų tinklų prognozavimo pradžia, taikymai bei evoliuciniai skaičiavimai.

Prognozavimas laiko srityje svarbus uždavinys. Daugelis procesų, kuriuos norima prognozuoti kinta tam tikrame laiko intervale. Egzistuoja įvairiausi metodai prognozavimui laiko srityje, visų pagrindinis tikslas pagal praeities duomenys ir tam tikrus parametrus sukonstruoti modelį ir pritaikyti ateities prognozavimui.

Ateities prognozavimui sukurtas, taikomas, tobulinamas ne vienas metodas. Pagrindiniai, populiarūs klasikiniai metodai, kurie buvo apžvelgti šiame skyriuje: autoregresinis slenkamųjų vidurkių (ARMA), tiesinės prognozės metodas (LPC) ir neuronų tinklai bei konkretūs metodai: netiesinis autoregresinis neuronų tinklas (NAR), netiesinis autoregresinis su papildomų įėjimu neuronų tinklas (NARX) ir laiko vėlinimo neuronų tinklas (TDNN). Šie metodai plačiai taikomi prognozavimui įvairiose srityse: saulės radiacijos, kalbos atpažinime, elektros sąnaudų, infliacijos rodiklių ir panašiai. Metodai skiriasi savo matematinėmis išraiškomis, struktūra, parametrais, veikimo principais.

Tolimesniame darbe bus eksperimentuojama su šiais prognozavimo metodais, tiriamos vidutinės paklaidos prognozuojant kintančius laike duomenys, metodai lyginami tarpusavyje.

4. SVETAINĖS ĮVERTINIMUI NAUDOJAMŲ PRIEŽASTINIŲ IR PASEKMINIŲ RODIKLIŲ ATRANKA

Dažniausiai, o galbūt ir visada, kai yra kalbama apie svetainės optimizaciją turima galvoje paieškos variklių optimizacija (angl. *search engine optimization*), toliau visur vadinama trumpiniu SEO. Trumpai tariant, tai yra nemokamo, organinio (angl. *organic*) srauto pritraukimas iš paieškos variklių. Populiariausias pasaulyje bei Lietuvoje paieškos variklis yra „Google“, kuris ir bus pabrėžiamas šiame darbe. Taip pat daugiau ar mažiau populiarius paieškos varikliai yra „Bing“, „Yandex“, „Yahoo“, „Seznam“ ir kiti.

Lankytojai svetainėje apsilanko keliais būdais: ateina tiesiogiai įvesdami į adresui skirtą laukelį (tokie lankytojai dažniausiai svetainėje lankosi ne pirmą kartą), ateina per išorines nuorodas iš kitų svetainių (forumų, blogų, partnerių svetainių ir pan.), o populiariausias, dažniausias ir daugiausiai srauto ir naujų lankytojų pritraukiantis būdas yra organinė/natūrali paieška. Vertinimo kriterijai, pagal kuriuos, „Google“ reitinguoja, neatitilsta nuo vartotojų lūkesčių. Pavyzdžiui, kiekvienam lankytojui atėjus į svetainę yra svarbi navigacija joje, svetainės paprastumas, aiškumas, kad atėjęs rastų tai, ko ieško, kad nereikėtų ilgai laukti, kol atsivers svetainė. Pagal šiuos aspektus „Google“ vertina vidinius svetainės aspektus, kurie dažniausiai vadinami „vidine optimizacija“: turinys, puslapio atvertimo greitis, vidinių nuorodų struktūra, metaduomenys ir kiti. Taip pat prie vidinės optimizacijos vertinami ir lankytojų reakcijos kriterijai: kaip atmetimo rodiklis (angl. *bounce rate*), kiek puslapių peržiūrėta, kokia vidutinė vizito trukmė ir pan. „Išorinė optimizacija“ ir jos kriterijai pagal kuriuos taip pat priklauso svetainės pozicija „Google“ paieškoje – išorinės nuorodos ir jos tipai. Išorinės nuorodos parodo, kiek lankytojų dalinasi informacija ir nuoroda į šią svetainę, kiek ji naudinga.

4.1. „Google“ algoritmų galimybių įvertinimas

Nesvarbu, ar svetainė nauja ar ne, dažniausias tikslas yra, kad ji būtų matoma ir žinoma, pasiekama lankytojų, kurie ieško susijusios informacijos. Interneto informacijos yra galybė, norint, kad pagal tam tikras užklausas paieškoje svetainė būtų kuo aukštesnė pozicijose, ji turi būtų gerai vertinama „Google“ algoritmų pagal tam tikrus kriterijus. „Google“ algoritmai turi labai daug vertinimo kriterijų, kurių svoriai yra skirtingi. Šie dalykai viešai nėra skelbiami, nes tai komercinis dalykas, kuriais niekas nenori, kad būtų per daug manipuluojama. Kokie kriterijai yra svarbūs, akcentuoja ir patys „Google“ atstovai, o daugumos svorį ir svarbą vertiname pasitelkę patirtį. Vienas iš svarbių aspektų yra tai, kad šie algoritmai keičiasi, tobulėja, atsinaujina labai dažnai. Kaskart yra išleidžiami ir nauji algoritmai kovoti su tais kurie platina brukalą internete, tobulinti paiešką ir pan. Pastaruoju metu populiariausi papildomi išleisti algoritmai yra „Google Penguin“

(Lietuvoje vadinamas tiesiog „Pingvinas“), „Google Panda“ – panda, ir naujausias „Google Hummingbird“ – kolibris. Paieškos algoritmas yra tam tikras komandų rinkinys, pagal kurį „Google“ robotai renka informaciją apie puslapius, ją sistemina ir pateikia rezultatus.

4.2. Pagrindinių išorinės optimizacijos matavimo/vertinimo rodiklių atranka

Prieš pradėdant išorinę (angl. *off-site*) optimizaciją svarbu įsivertinti raktinių žodžių naudojimą išorinėse nuorodose (angl. *anchor text*), jų populiarumą (kiekį/kokybę), nuorodų šaltinių įvairovę (kiek nuorodų iš skirtingų domenu), nuorodas iš aukštos reputacijos tinklapių ir iš susijusių tinklapių/puslapių (teminės nuorodos).

Konkurencingose rinkose išorinė optimizacija sudaro 70–80 % viso SEO darbo proceso (vidinis optimizavimas užima 20–30 %).

Svetainės *mazylioratai.lt* atveju išorinių nuorodų tinklelio analizė pateikta 4.1 lentelėje.

4.1 lentelė. Svetainės *mazylioratai.lt* išorinių nuorodų tinklelio analizė

Unikalūs domenai (kuriuose yra nuorodos į tinklapį)	93
Unikalūs IP	86
Viso nuorodų į tinklapį	3936
„No follow“ nuorodos	2983
„Site wide“ nuorodos	942

Šie duomenys buvo tikrinami svetainėje ahrefs.com

Nuorodos iš geros reputacijos tinklapių – autoriteto ir pasitikėjimo kėlimas. Reikalingos nuorodos iš labai autoritetingų ir didelį pasitikėjimą turinčių tinklapių. Idealu būtų, jei tai būtų susiję tinklapiai (su kategorijomis/prodūktais susijusios svetainės/puslapiai ir t. t.). Tai padėtų pakelti pradinio ir tuo pačiu vidinių puslapių autoritetą – paieškos sistemos svetaine pasitikėtų labiau – svetainės puslapiai būtų reitinguojami aukščiau.

Nuorodų į tinklalapį kiekis/kokybė. Geriau mažiau, bet susijusių nuorodų (nuorodos iš puslapių, straipsnių, tinklaraščių įrašų, forumų, kuriuose kalbama apie siūlomus produktus), nei daug nuorodų iš nesusijusių – šlamšto direktorių („nuorodų fermų“).

Nuoroda iš mažesnį autoritetą turinčio, bet susijusio turinio svetainės, yra vertingesnė nei nuoroda iš aukštą autoritetą turinčio, bet iš visai nesusijusio puslapio/tinklapio.

Svarbiausi išorinės optimizacijos faktoriai domeno lygiu:

Domeno lygiu:

- Unikalių domenų su nuorodomis į svetainę skaičius.
- Domenų su nuorodomis į svetainę autoritetas.
- Domenų su nuorodomis į svetainę patikimumas.
- Unikalių puslapių su nuorodomis į svetainę skaičius.
- Nuorodos iš susijusio turinio domenų.
- Nuorodų atsiradimo dažnumas.
- Nuorodų naudojančių prekinį ženklą kaip „anchor“ tekstą procentas.
- Nuorodos iš skirtingų svetainių tipų: tinklaraščių, forumų, straipsnių direktorių ir pan.
- Nuorodų į svetainę pozicija puslapyje.

Svarbiausi išorinės optimizacijos faktoriai puslapio lygiu:

- Unikalių domenų su nuorodomis į puslapį su susijusiu ar konkretaus atitikmens nuorodos tekstu, skaičius.
- Unikalių domenų su nuorodomis į puslapį skaičius.
- Unikalių nuorodų į puslapį su susijusiu ar konkretaus atitikmens nuorodos tekstu, skaičius.
- Puslapių su nuorodomis į puslapį patikimumas.
- Nuorodos iš susijusio turinio puslapių.
- Puslapių su nuorodomis į puslapį autoritetas.
- Unikalių puslapių su nuorodomis į svetainę skaičius.
- Nuorodos iš skirtingų svetainių tipų: tinklaraščių, forumų, straipsnių direktorių ir pan.
- Nuorodų pozicijos puslapyje .

Nėra kriterijaus ar ribos, kurio vertinimo rodiklio vertė aukšta, kurio ne. Viskas priklauso nuo vietinės rinkos, nuo paklausos ir pasiūlos, t. y. nuo panašaus tipo svetainių konkurencijos ir paieškų skaičiaus. Visi šie duomenys yra labai kintantys. Todėl geriausiai tirti skirtingas rinkas individualiai, arba vertinti tam tikrų duomenų koreliaciją, jų pokyčius ir įtaką.

Šiame darbe bus vertinami ir matuojami keli išorinės optimizacijos kriterijai: visos išorinės nuorodos, „sitewide“ išorinės nuorodos, „do follow“ ir „no follow“ išorinės nuorodos.

Taip pat, kadangi vis didesnę vertę turi socialiniai profiliai (taip akcentuoja „Google“ ir bendroji praktika), bus matuojamas ir vertinamas populiariausio socialinio tinklo „Facebook“ kriterijus – kiek lankytojų paspaudė mygtuką „patinka“ ir taip pat sekėjų skaičius „Google Plus“ profilyje. Kadangi šis socialinis profilis yra „Google“ produktas, todėl jis yra labai svarbus.

4.3. Pagrindinių vidinės optimizacijos matavimo/vertinimo rodiklių įtakos analizė

Svarbiausi svetainės SEO optimizacijos aspektai, norint geresnio „Google“ robotų vertinimo ir reitingavimo:

- Raktinių žodžių naudojimas puslapio pavadinime (angl. *meta title*).
- Raktinių žodžių naudojimas turinyje.
- Unikalus turinys, turinio atnaujinimas.
- Tinklapio architektūra (aiški hierarchija, navigacija, vidinės nuorodos).
- Svetainės puslapių užkrovimo greitis.
- Raktinių žodžių naudojimas antraščių pavadinimuose.
- Raktinių žodžių naudojimas paveikslėlių alternatyviame tekste.
- Raktinių žodžių naudojimas URL (domene, subdomene, paslaugų puslapiuose).

4.3.1. Puslapio pavadinimo įtakos analizė

Raktinius žodžius puslapio pavadinimuose (angl. *meta title*) reikia parinkti pagal tai, ko ieško žmonės, panaudojant įvairius linksnius. Kadangi pavadinimas rodomas paieškoje, reikia jį padaryti „draugiškesniu“ ir aiškesniu naudotojui. Paieškos sistemose pozicijoms įtaką turi ir vartotojų paspaudimas ant nuorodos tarp paieškos rezultatų (t. y. jei jus rodo tarp rezultatų, bet vartotojai ant jūsų įrašo nespaudžia, „Google“ supranta tai kaip nesusijusį turinį).

Puslapio pavadinime turi būti optimizuojami raktiniai žodžiai, kiekvienas tinklalapio puslapis turi būti skirtas konkrečios frazės ar kelių frazių (iki 5) optimizavimui. Svarbiausi raktiniai žodžiai turi būti pavadinimo pradžioje, po to antriniai raktiniai žodžiai ir/ar prekinis ženklas.

Puslapio pavadinimas turi būti iki 70 ženklų (įskaičiuojant tarpus), kad tekstas nebūtų trumpinamas paieškos rezultatuose ir vartotojas matytų pilną puslapio pavadinimą. Galimos išimtys: jeigu pilnas prekės ar paslaugos pavadinimas turi daugiau kaip 70 ženklų, jį galima palikti.

Puslapio pavadinimus galima rodyti ir didžiosiomis raidėmis, kas leistų padidinti paspaudimų rodiklį. Tačiau tai daryti nepatartina, nes atrodo kaip brukalas. Kiekvienas svetainės puslapis turi turėti unikalų pavadinimą.

Mazylioratai.lt puslapio pavadinimas atrodo - `<title>Mažylio ratai - prekės vaikams internetu, vaikiški vežimėliai, vežimėlių remontas, žaislai vaikams</title>`.

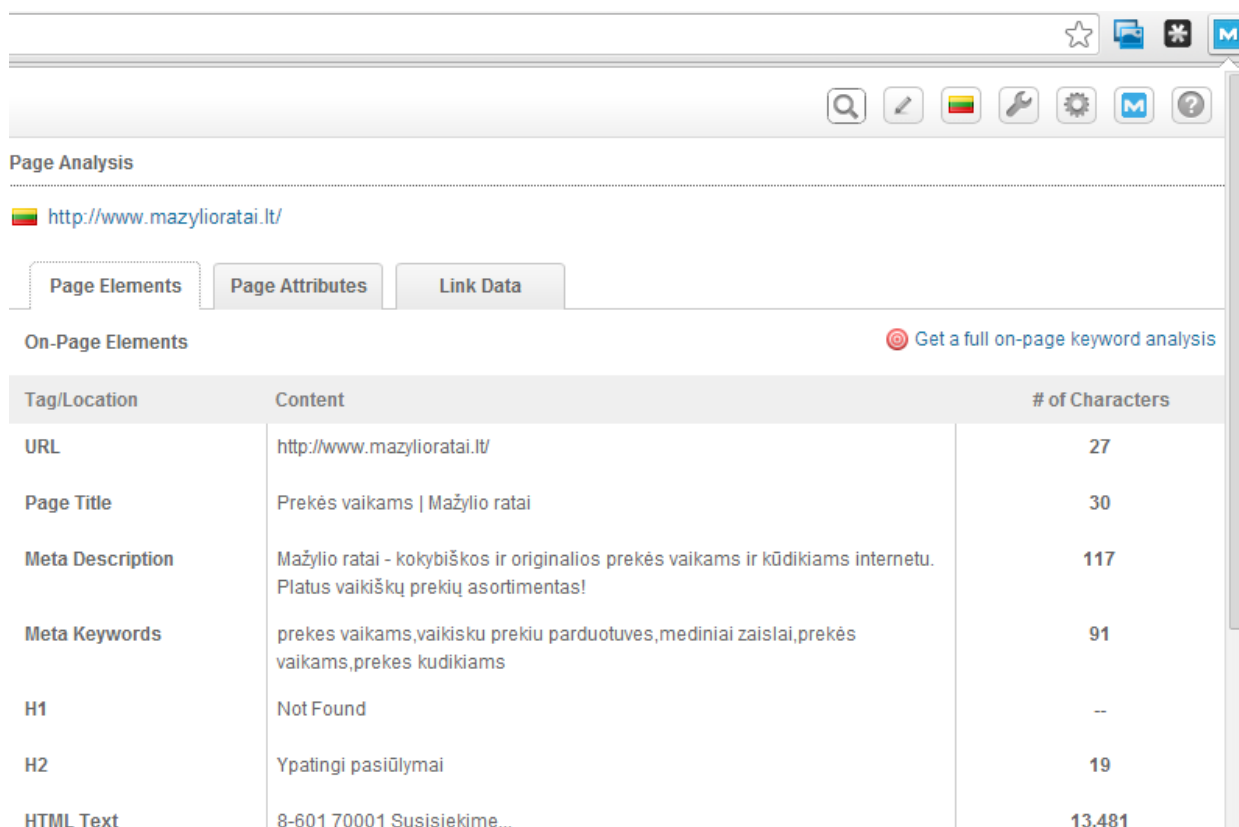
Pasiūlymas susitvarkyti tokiu principu: „Puslapio pavadinimas | Mažylio Ratai“. Pradinio svetainės puslapio pavadinimas turėtų atrodyti – Prekės vaikams | Mažylio ratai. Tokiu pat principu sudaryti ir visus kitus svetainės puslapius. Tarkim kategorijos vežimėliai, pavadinimas turėtų atrodyti – „Vežimėliai | Mažylio Ratai“, o subkategorijos „universalūs vežimėliai“, turėtų būti – „Universalūs vežimėliai | Vežimėliai | Mažylio ratai“. 4.1 paveiksle matosi, kaip puslapio pavadinimas ieškant vežimėlių rodomas dabar.

vaikiski vezimeliai - Vežimėliai | vezimeliu remontas, prekės vaikams ...
www.mazylioratai.lt/vezimeliai Translate this page
 10+ items – 8-601 70001 Susisieikime. Registruotis. Prisijungimas ...
 20 % Vežimėlis X-Lander XS 2013. 295
 35 % Vežimėlis Navington Cadet 3in1 su autokėdute Navington ... 1199

4.1 pav. Mazylioratai.lt puslapio pavadinimas ieškant vežimėlių

Dabartinis puslapio pavadinimas yra per ilgas ir netelpa paieškoje. Daug frazių sudėtą į vieną, todėl paieškos robotai gali traktuoti kaip ne visai susijusį turinį.

Puslapio pavadinimus galima pasižiūrėti tiesiogiai užvedus pelytę ant svetainės puslapio skirtuko. Taip vienas iš patogesnių būdų įsirašyti nemokamą „moz“ įskiepi į naršyklę. Šis įskiepis yra naudojamas šiame darbe patikrinti visus vidinės optimizacijos rodiklius. 4.2 paveiksle parodyta kaip atrodo „moz“ įrankis ir kokius duomenis rodo konkrečioje mazylioratai.lt svetainėje.



The screenshot shows the MozBar extension interface. At the top, it displays the URL <http://www.mazylioratai.lt/>. Below this, there are tabs for 'Page Elements', 'Page Attributes', and 'Link Data'. The 'Page Elements' tab is active, showing a table of on-page elements. The table has three columns: 'Tag/Location', 'Content', and '# of Characters'. The data is as follows:

Tag/Location	Content	# of Characters
URL	http://www.mazylioratai.lt/	27
Page Title	Prekės vaikams Mažylio ratai	30
Meta Description	Mažylio ratai - kokybiškos ir originalios prekės vaikams ir kūdikiams internetu. Platus vaikiškų prekių asortimentas!	117
Meta Keywords	prekes vaikams, vaikisku prekiu parduotuves, mediniai zaislai, prekės vaikams, prekes kudikiams	91
H1	Not Found	--
H2	Ypatingi pasiūlymai	19
HTML Text	8-601 70001 Susisieikime...	13,481

4.2 pav. „Moz“ įskiepio duomenys mazylioratai.lt svetainei

Šis įrankis/įskiepis yra nemokamas ir labai patogus. Galima matyti visus metaduomenis vienoje vietoje, nereikia žiūrėti į puslapio išeities kodą.

4.3.2. Puslapių antraščių įtakos analizė

Puslapių antraštės (angl. *headings*) turi daugiau reikšmės nei paprastas tekstas. Parinkus tinkamą antraštę didėja ir raktinių žodžių tankumas puslapyje.

Antraštės turi būti derinamos prie puslapio pavadinimo. Optimaliausia naudoti šiek tiek platesnę frazę ar kitus susijusius raktinius žodžius. Platesnės, susijusios frazės parenkamos pagal

tai, ko ieško naudotojai. Galima naudoti ir konkrečią optimizuojamą frazę, tačiau svarbu, kad antraštė ir puslapio pavadinimas nebūtų visiškai identiški.

Kuriant puslapio pavadinimą principu „Raktinis žodis - Prekinis ženklas“ antraščių ir pavadinimo dubliavimosi yra išvengiama.

Kiekviename puslapyje negali būti daugiau kaip vieno *h1*, daugiau kaip dviejų *h2*, daugiau kaip trijų *h3* ir t.t.

„Mažylio Ratai“ projekte puslapių antraštės parinktos gan optimaliai, bet tam tikrose vietose galėtų būti koreguojamos. Tarkim vežimėlių kategorijoje, *h1* atrodo `<h1>Vaikiški vežimėliai</h1>`. Principas teisingas, ir atsižvelgus į tai, kad *h1* yra matomas puslapio lankytojams jis turėtų būti lietuviškas. Bet pagal paieškų skaičių šiuo atveju geriau būtų frazė – vežimėliai, nes sulaukia kur kas daugiau paieškų.

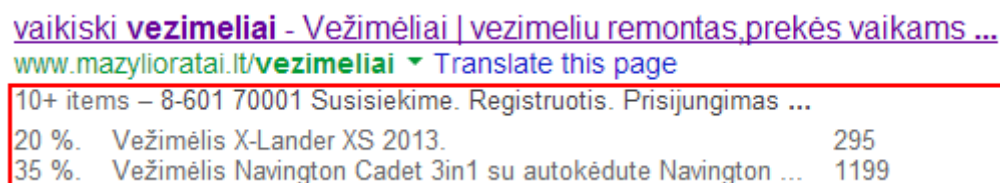
Duomenys žiūrėti naudojant „moz“ įskiepi.

4.3.3. Puslapio aprašymo įtakos analizė

Nors paieškos sistemos neatsižvelgia į puslapio aprašymą (angl. *meta description*) reitinguodamos puslapius, aprašymas turi didelę įtaką vartotojų pritraukimui. Svarbu pagrindiniams puslapiams sukurti gerus šūkius/ragimus (angl. *call to action*) – „Išigykite“, „Užsisakykite“ ir pan. Tai padės padidinti paspaudimų rodiklį iš nemokamos paieškos rezultatų.

Optimaliausias puslapio aprašymas turėtų būti 25 – 150 ženklų (įskaičiuojant tarpus), kad puslapio aprašymas nebūtų trumpinamas paieškos rezultatuose. Nesukūrus aprašymo, jis generuojamas automatiškai (paieškos sistemos parenka tekstą iš puslapio).

4.3 paveiksle pateikta vežimėlių paieškos rezultatas ir pažymėta koks yra puslapio aprašymas.



4.3 pav. Puslapio aprašymas ieškant vežimėlių

Aprašymas yra sugeneruotas automatiškai, per ilgas ir nieko nepasakantis. Aprašymas imamas iš puslapio turinio pačios pradžios `<meta name="description" content="" />`. Šis aspektas turi būti keičiamas, norint pritraukti didesnio srauto lankytojų, sudominti juos. Tiesioginės vertės SEO optimizacijai tai neturi, bet didesnę srautą „Google“ interpretuoja kaip susijusį ir patikimą turinį bei svetainę.

Aprašymą galima matyti naudojant minėtą „moz“ įrankį.

4.3.4. URL struktūros įtakos analizė

URL struktūra turi būti aiški tiek vartotojams, tiek paieškos sistemų robotams.

Optimaliausia URL struktūra:

<http://www.puslapis.lt/kategorija/sub-kategorija/produktas/>

Galimi ir kiti variantai:

- <http://www.puslapis.lt/kategorija/produktas/>
- <http://www.puslapis.lt/sub-kategorija/produktas/>
- <http://www.puslapis.lt/produktas/>

Kuo optimizuojamas raktinis žodis yra arčiau pagrindinio svetainės domeno URL, tuo jis turi didesnę reikšmę. Nors paieškos rezultatams tai daro minimalią įtaką, puslapiai bus reikšmingesni.

URL būtina naudoti pagrindinius optimizuojamus žodžius, pirmumo teisę suteikiant labiausiai ieškomoms frazėms.

Visuose svetainės puslapiuose reiktų naudoti tokį pat URL formavimo formatą.

„Mažylio Ratai“ URL struktūra daugeliu atveju sudaryta teisingai, bet galėtų būti minimaliai pakoreguota. Tarkime, nuėjus į vežimėlių kategoriją ir į universalių vežimėlių subkategoriją, URL, kuris atrodo taip: <http://www.mazylioratai.lt/vezimeliai/universalus>, turėtų būti pakoreguotas naudojant pilnus pavadinimus – <http://www.mazylioratai.lt/vezimeliai/universalus-vezimeliai>.

Šis kriterijus nėra tiksliai ir lengvai įvertinamas, tam „Google“ naudoja sudėtingus algoritmus. Todėl tai nebus įtrauktą į duomenų analizę.

4.3.5. **Paveikslų alternatyvaus teksto įtakos analizė**

Paveikslėlių alternatyvus tekstas (trumpinys – alt) reikalingas tam, kad paieškų sistemų robotai žinotų, kas paveikslėlyje pavaizduota, nes kol kas paveikslėlių turinio jie nemato ir interpretuoti negali. Rekomenduojama, kad alt tekstas būtų išvedamas panaudojant pagrindinius ir susijusius raktinius žodžius bei/ar platesnes/siauresnes frazes.

„Mažylio Ratai“ pagrindinio logotipo alt tekstas dabar atrodo – "Prekės vaikams - Mažylio Ratai". Kadangi alt tekstas nėra matomas išoriškai lankytojams, jis gali būti nelietuviškomis raidėmis ir parinktas pagal tai, kas turi daugiausiai paieškų ir atitinka visos svetainės turinį. Taigi šiuo konkrečiu atveju alt tekstas turėtų būti „prekes vaikams“. Panašiu principu turėtų būti parenkami visų kitų prekių paslaugų paveikslų alt tekstai.

Šis rodiklis nėra vienas iš svarbiausių todėl į duomenų analizę taip pat nebus įtrauktas.

4.3.6. **Raktinių žodžių ir jų naudojimo svetainėje įtakos analizė**

Didžiausias dėmesys yra mažos konkurencijos (angl. *long tail*) raktiniams žodžiams.

Atlikus organinio (angl. *organic*) nemokamo srauto tyrimus (duomenys iš įmonės „SEM.LT“), nustatyta, kad koncentruotis vien į didelio konkurencingumo frazes neverta – jos sudaro vos 20-30 % nemokamo srauto iš paieškos sistemų.



4.4 pav. Raktinių žodžių pasiskirstymo kreivė

Didžiausias srautas į svetainę ateina per mažos konkurencijos, nedidelį paieškos skaičių turinčias frazes – 70 %. Norint padidinti „*longtail*“ srautą reikėtų sutelkti dėmesį į: puslapio pavadinimą, alternatyvų tekstą ir turinį.

Kuo puslapyje naudojama daugiau raktinių žodžių, kuo jame daugiau įvairių sinonimų, tuo jis turi didesnę vertę ir gali pritraukti lankytojus pagal daug įvairių raktinių frazių.

Optimaliausias raktinių žodžių tankumas puslapyje yra 4 % viso teksto. Gali būti ir mažiau, tačiau geriausiai neviršyti 4 %, nes paieškos sistemos tai gali palaikyti brukalu (angl. *spam*). Taip pat būtina naudoti susijusius raktinius žodžius – sinonimus.

Raktinių žodžių tankiui tikrinti rekomenduojamas <http://tools.seobook.com/general/keyword-density/> įrankis.

4.3.7. Puslapių atvertimo greičio įtakos analizė

Svetainės atvertimo greitis yra įvertintas 100 balų sistemoje <https://developers.google.com/speed/>. Yra keli nesudėtingi pakeitimai, kuriuos atlikus svetainės puslapių atvertimo greičiai dar padidėtų.

Kuo greičiau atverčiami puslapiai, tuo palankiau paieškos sistemos vertina tokias svetaines. Kaip pavyzdžiui, jei yra dvi svetainės, siūlančios labai panašias ar kartais net identiškąs prekes, kurių išorinių nuorodų tinklėlis panašus, svetainės turinys irgi mažai skiriasi, tai svetainę, kurios puslapių atvertimo greičio duomenys yra geresni, paieškos sistemos reitinguos aukštesnėje pozicijoje. Papildomas resursas, kur galima įsivertinti svetainės puslapių užkrovimo greičius bei kaip juos padidinti: <https://developers.google.com/speed/>, <http://www.webpagetest.org/>. Taip pat, vienas iš patogių įrankių matuoti puslapių atvertimo greitį, kuriuo ir bus matuojamas šiame darbe,

yra „Google Analytics“. Užkrovimo greitis matuojamas sekundėmis. Pagal bendrą praktiką yra rekomenduojama, kad atvertimo greitis neturėtų viršyti 4 s.

4.4. Pagrindinių pasekmių matavimo/vertinimo rodiklių analizė

Svetainė yra optimizuojama, atliekama išorinė, vidinė optimizacijos iš esmės tam, kad būtų pasiekti tam tikri užsibrėžti tikslai. Pirmiausiai apsibrėžiami tikslai ir poreikiai pagal, kuriuos vėliau yra priimami sprendimai, atliekami veiksmai, optimizacija, matavimai. Tikslas gali būti padidinti srautą į svetainę, pavyzdžiui – kelis kartus. Dažniausiai labai svarbios yra konversijos, kurias galima pamatuoti ir įvertinti jų pokytį. Konversijos priklauso nuo svetainės pobūdžio, kliento poreikių ir galimybių. Tai gali būti nuėjimas iki kontaktų, jei elektroninė parduotuvė – prekių įsidėjimas į krepšelį, apsipirkimas. Taip pat svarbūs matuojami dalykai gali būti: praleistas laikas svetainėje, peržiūrėtų puslapių skaičius. Šiuos rodiklius galima lengvai surasti naudojant „Google Analytics“ įrankį. Matavimas gali būti vykdomas patalpinus specialų sekimo kodą svetainėje, sugeneruota įrankio „Google Analytics“, arba nustatant tikslus pačiame įrankyje.

Pavyzdžiui svetainės „mazylioratai.lt“ savininkui svarbu yra konversijos – apsipirkimai, nes tai yra vaikų prekių elektroninė parduotuvė. Svarbios visos prekės, kategorijos ir kuo daugiau pirkimų. Didžiausias prioritetas buvo išskirtas vežimėlių kategorijai, kas reiškia, kad svarbiausi pirkimai, ateinantis srautas bei pozicijos „Google“ natūralioje paieškoje yra pagal šią kategoriją. 4.5 paveiksle pateikta „mazylioratai.lt“ svetainės tikslų nustatymų pavyzdys „Google Analytics“ paskyroje.

Profile: www.mazylioratai.lt/indexpage (Web)

Property ID: UA-13088508-1

Assets

Users

Goals

Filters

Profile Settings

CREATE A GOAL

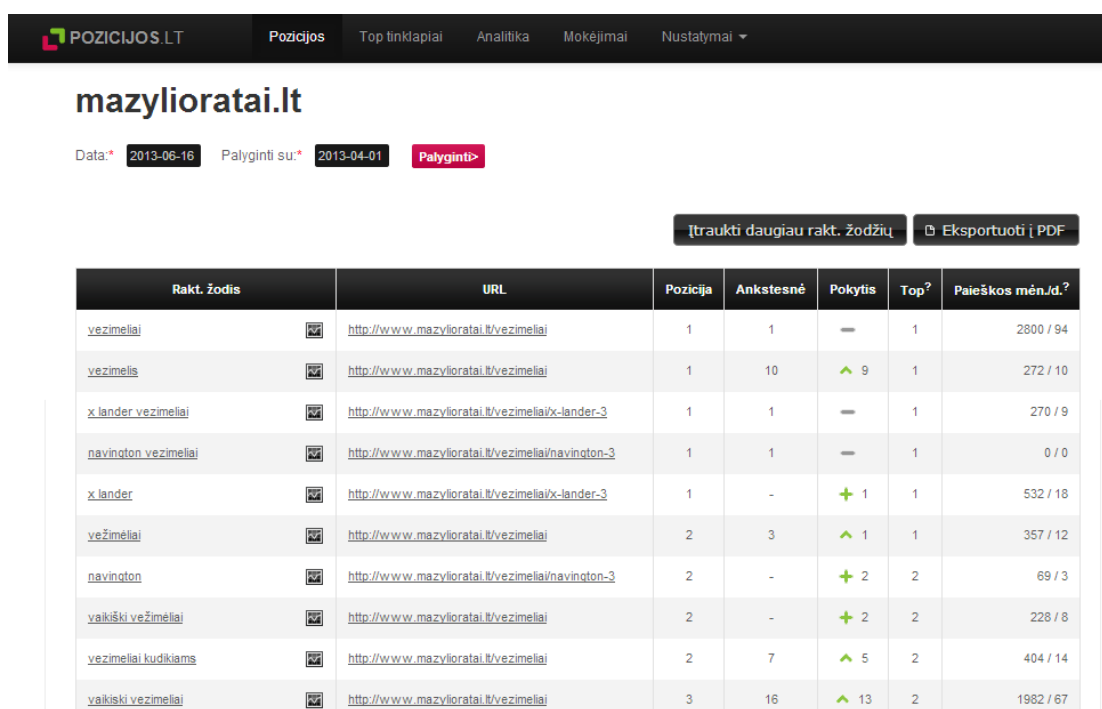
Goal	Past 7 day conversions	Recording
Kontaktai Goal ID 1 / Goal Set 1	133	On
Pirkimas Goal ID 6 / Goal Set 2	11	On

4.5 pav. Mažylio ratai nustatyti tikslai „Google Analytics“ paskyroje

Pagal 4.5 paveiksle matoma, kad yra nustatyti keli tikslai. Pirmasis – tai nuėjimai iki kontaktų, antrasis – įvykdytų pirkimų skaičius. Nuėjimai iki kontaktų yra svarbūs, nes pagal juos galima matyti, kiek yra tikslinių lankytojų, kurie raktiniai žodžiai – prekės yra paklausios, perkamos.

Dažniausiai visiems yra svarbios pozicijos „Google“ paieškoje, pagal tam tikras užklaudas, raktinius žodžius. Kuo aukštesnės pozicijos, tuo labiau svetainė yra pastebima, dažniau spaudžiama ant nuorodos į svetainę, o tai reiškia, kad sugeneruojama daugiau srauto į svetainę. Jei raktinis žodis ar frazė yra tikslingi, tai daro įtaką ir konversijų augimui. Kitu atveju, jei srautas didelis, bet dauguma lankytojų atėję nerastų to, ko jiems reikia, iškart užvertų svetainės langą, „Google“ fiksuotų kaip svetainės atmetimą, todėl išaugtų atmetimo rodiklis (angl.bounce rate). Pastarasis

rodiklis, duoda signalą „Google“, kad svetainė nėra tinkama pagal šią užklausą, dėl šios priežasties kristų ir pozicijos. Atmetimo rodiklis matuojamas naudojant „Google Analytics“ įrankį. Pozicijos pagal pasirinktus raktinius žodžius gali būti matuojamos įvairiai būdais. Paprastas ir visiems prieinamas būdas yra įvesti šį raktinį žodį į „Google“ paiešką. Esant daugiau raktinių žodžių, patogiau turėti visus raktinius žodžius ir jų kitimą vienoje vietoje. Sekti pozicijas galima pasinaudojus mokama sistema „pozicijos.lt“. 4.6 paveiksle pavaizduota, kaip galima sekti pačių pasirinktų raktinių žodžių pozicijas, matuoti pokytį nusistatytomis dienomis. Raktinių žodžių pozicijas galima lengvai sekti naudojant patogų, bet mokamą įrankį – „AWR“. Raktinių žodžių pozicijų sekimo įrankių paieškos sistemose yra kur kas daugiau..



The screenshot shows the 'pozicijos.lt' website interface. At the top, there's a navigation bar with 'Pozicijos', 'Top tinklapiai', 'Analitika', 'Mokėjimai', and 'Nustatymai'. Below the navigation bar, the domain 'mazylioratai.lt' is displayed. There are date selectors for 'Data' (2013-06-16) and 'Palyginti su' (2013-04-01), along with a 'Palyginti' button. On the right, there are buttons for 'Įtraukti daugiau rakt. žodžių' and 'Eksportuoti į PDF'. The main content is a table with the following columns: Rakt. žodis, URL, Pozicija, Ankstesnė, Pokytis, Top?, and Paieškos mėn./d.?

Rakt. žodis	URL	Pozicija	Ankstesnė	Pokytis	Top?	Paieškos mėn./d.?
vezimeliai	http://www.mazylioratai.lt/vezimeliai	1	1	—	1	2800 / 94
vezimelis	http://www.mazylioratai.lt/vezimelis	1	10	▲ 9	1	272 / 10
x lander vezimeliai	http://www.mazylioratai.lt/vezimeliai/x-lander-3	1	1	—	1	270 / 9
navinton vezimeliai	http://www.mazylioratai.lt/vezimeliai/navinton-3	1	1	—	1	0 / 0
x lander	http://www.mazylioratai.lt/vezimeliai/x-lander-3	1	-	▲ 1	1	532 / 18
vežimėliai	http://www.mazylioratai.lt/vezimeliai	2	3	▲ 1	1	357 / 12
navinton	http://www.mazylioratai.lt/vezimeliai/navinton-3	2	-	▲ 2	2	69 / 3
vaikiški vežimėliai	http://www.mazylioratai.lt/vezimeliai	2	-	▲ 2	2	228 / 8
vezimeliai kudikiams	http://www.mazylioratai.lt/vezimeliai	2	7	▲ 5	2	404 / 14
vaikiški vezimeliai	http://www.mazylioratai.lt/vezimeliai	3	16	▲ 13	2	1982 / 67

4.6 pav. Svetainės „mazylioratai.lt“ raktiniai žodžiai ir jų pozicijų kitimas matuojamas „pozicijos.lt“ įrankiu

Pozicijos yra lyginamos birželio 14 dieną su balandžio 1 diena, kai dar nebuvo vykdoma optimizacija. Pateikti yra ne visi raktiniai žodžiai. Raktinis žodis „vezimeliai“, jau buvo pirmoje pozicijoje, o tai parodo, kad jam nereikia papildomos optimizacijos, o tik palaikymo. Kai kurie raktiniai žodžiai pakilo iš žemesnių pozicijų, o kiti nebuvo net randami Google paieškoje prieš optimizaciją, pavyzdžiui raktinis žodis – „vaikiški vežimėliai“, kuris dabar yra antroje pozicijoje. Vienas iš variantų sekti pozicijas ir kitus svetainei svarbius aspektus naudojant „Google Webmaster Tool“ (trumpinys – GWT) įrankį. Su šiuo įrankiu galima stebėti pozicijų ir srauto pokyčius, domeno ir kitas technines klaidas.

4.5. Interneto svetainių matuotų ir surinktų rodiklių atranka

Tyrimo metu buvo renkami statistiniai duomenys. Duomenys buvo renkami kartą per dieną, dėl naudojamų įrankių duomenų pateikimo ribotumo. Duomenų istorija priklauso nuo to, kada

svetainės duomenys yra sekami. Visi rodikliai ir įrankiai buvo aprašomi plačiau ankstesniame skyriuje.

Duomenys buvo renkami iš trijų svetainių, projektų bei vienos Lietuvos paskolų rinkos. Svetainės: „bobutespaskola.lt“, „paskoliukas.lt“, „popkreditas.lt“. 4.7 paveiksle pateikiama renkamų duomenų lentelės dalis.

BOBUTES PASKOLA					
Data	Atmetimo rodiklis (bounce rate), % GA	Vidutinė vizito trukmė, s GA	Avg page load time, s GA	Facebook likes	G+ likes
2013-07-25	32,16	334	6,17	N/A	-
2013-07-26	38,11	454	4,50	N/A	-
2013-07-27	30,37	532	3,36	N/A	-
2013-07-28	32,08	417	3,10	N/A	-
2013-07-29	31,98	445	20,84	N/A	-
2013-07-30	36,49	329	4,07	N/A	-
2013-07-31	33,52	417	6,01	N/A	-
2013-08-01	34,00	393	3,80	N/A	-
2013-08-02	30,16	390	5,52	N/A	-
2013-08-03	30,77	472	6,61	N/A	-
2013-08-04	37,50	358	4,71	N/A	-
2013-08-05	29,41	421	4,78	N/A	-
2013-08-06	33,68	412	9,04	N/A	-
2013-08-07	31,67	446	3,44	N/A	-

4.7 pav. „Bobutės paskola“ svetainės statistinių duomenų dalis

Renkami duomenys buvo suskirstyti į dvi platesnes grupes – priežastinius ir pasekminius.. Priežastiniai rodikliai: atmetimo rodiklis, vidutinė vizito trukmė, vidutinis puslapio atvertimo laikas. Šie paminėti rodikliai gauti naudojant „Google Analytics“ įrankį. Socialinio profilio „Facebook“ „like“ skaičius, socialinio profilio „Google Plus“ „follow“ skaičius – tiesiogiai žiūrėti socialinių profilių istorijoje. Išorinių nuorodų skaičius: bendras, „do follow“ nuorodos (nuorodos, kurios parodo „Google“ robotams, kad jas reikia sekti, tokios nuorodos yra vertinamos geriau nei „no follow“), „no follow“ („Google“ robotams duoda signalą, kad jas nereikėtų sekti, toks sprendimas priimamas reklamjuostėms, kitu atveju tai būtų labai pasikartojančios nuorodos, kurios atrodytų kaip brukalas), „sitewide“ nuorodos (nuorodos, kurios yra svetainėse per visus joje esančius puslapius, o dažniausiai teigiamia, kad šios nuorodos reikalingos tik tam, kad nuorodų tinklėlis atrodytų natūralesnis, bet jų turėtų būti kuo mažiau). Šie duomenys paimti naudojant įrankį „ahrefs.com“. Vidinės nuorodos (svetainėje esančios nuorodos į vidinius puslapius), „moz rank“ rodiklis (populiarus įrankio „moz“ svetainės vertinimas), DA (domeno autoritetas) ir PA (puslapio autoritetas). Pastarieji duomenis paimti naudojant „moz“ įskiepi. Raktinių žodžių tankis paimtas iš „seobook.com“ svetainės.

Pasekminiai rodikliai: bendras paspaudimų skaičius, lankytojų, kurie atėjo per natūralią paiešką į svetainę, skaičius ir konversijos, kurios kiekvienai svetainei yra unikalios. Šiuo atveju bendros konversijos skaičiuojamos – lankytojai, kurie gavo banko patvirtinimą + lankytojai, kurie pasirinko

banką apmokėjimui + lankytojai, kurie nuėjo iki registracijos puslapio. Tai nereiškia, kad visi šie lankytojai būtinai užbaigs procesą, ar taps klientais, bet manoma, kad bent 60 % iš jų pasiteisino.

4.6. Skyriaus apibendrinimas

Dažniausiai svetainės optimizuojamos tam, kad pritraukti daugiau srauto ir konversijų. Pagrindinis srauto ir konversijų šaltinis yra „Google“ organinė paieška. Svetainės optimizuojamos, kad būtų geriau reitinguojamos „Google“ natūralioje paieškoje. Tokio pobūdžio optimizacija dažnai vadinama SEO. Svetainės optimizuojamos taip, atitiktų „Google“ algoritmų reikalavimus, vertinimų kriterijus, kurie daugiau ar mažiau atitinka lankytojo lūkesčius svetainėje. Populiariausi „Google“ algoritmai: „Google Penguin“, „Google Panda“, Google Hummingbird“.

Pagrindiniai išorinės optimizacijos matavimo ir vertinimo rodikliai, norint, kad svetainė geriau reitinguotųsi: unikalūs IP, visos nuorodos į tinklalapį, „no-follow“ nuorodos, „site-wide“ nuorodos. Vidinės optimizacijos matavimo ir vertinimo rodikliai: puslapio pavadinimas, puslapio antraštė, puslapio aprašymas, svetainės adresas, vidinių nuorodų struktūra, alternatyvus tekstas, raktinių žodžių naudojimas svetainėje, puslapių užkrovimo greitis ir kiti.

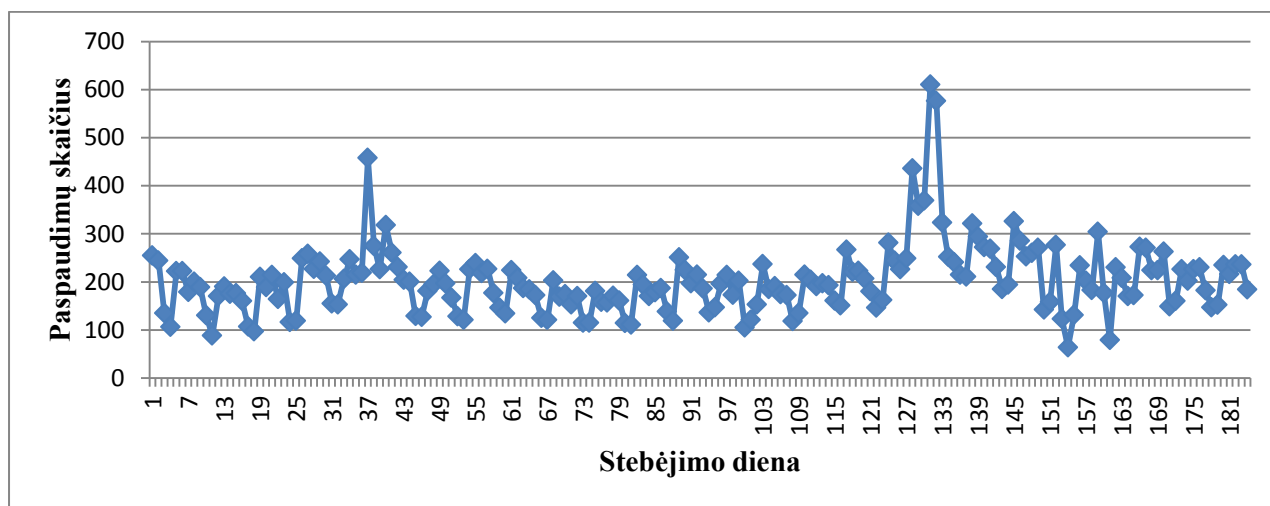
Duomenys surinkti iš trijų toje pačioje paskolų rinkoje esančių projektų: „Bobutės paskola“, „Paskoliukas“, „Popkreditas“. Duomenys atnaujinami kartą per dieną.

Pagrindiniai pasekmių matavimo, vertinimo rodikliai: raktinių žodžių pozicijos, konversijos ir srautas. Priežastiniai rodikliai: atmetimo rodiklis, vidutinė vizito trukmė, vidutinis puslapio krovimo laikas, išorinių ir vidinių nuorodų skaičius.

5. PROGNOZAVIMO METODŲ EFEKTYVUMO TYRIMO REZULTATAI IR JŲ ANALIZĖ

Darbo metu buvo pasirinkta analizuoti ir tirti kelis prognozavimo metodus: tiesinį prognozavimo metodą (LPC), autoregresinį paslinktųjų vidurkių prognozavimo metodą (ARMA), nelinijinį autoregresinį neuronų tinklą (NAR), nelinijinį autoregresinį su papildomu įėjimų neuronų tinklą (NAR) ir laiko vėlinimo neuronų tinklą (TDNN). Šie klasikiniai metodai buvo pasirinkti dėl patogaus pritaikymo MATLAB® aplinkoje, dėl plataus pritaikymo pajamų ir kitų aspektų prognozavimui.

Analizei buvo pasirinkti projektai: „Bobutės paskola“, „Popkreditas“ ir „Paskoliukas“. Šie projektai buvo pasirinkti, todėl visi trys yra iš vienos paskolų rinkos, optimizuojami tiems patiems raktiniams žodžiams ir turimas priėjimas prie visų šių projektų duomenų. Taip pat, ši rinka pastaruoju metu yra labai paklausė ir konkurencinga, todėl svetainės yra gana populiarios, sulaukia nemažai srauto bei konversijų. Kokie duomenys buvo renkami ir kokie įrankiai naudojami, plačiau aprašyta 4 skyriuje. Projekto „Bobutės paskola“ atliekami dvigubi skaičiavimai: su visais duomenimis, ir su mažesniu intervalu, išvengiant pikų, kurie akivaizdžiai matosi 5.1 paveiksle grafike. Šie staigūs pakilimai priklauso nuo metų sezoniškumo, tendencijų. Kadangi metų laikotarpio duomenų nėra, norint geriau įvertinti prognozavimo metodus, analizuojama „Bobutės paskola“ ir be staigių šuolių, kas turėtų daryti įtaką mažesnei paklaidai.



5.1 pav. „Bobutės paskola“ paspaudimų skaičiaus kitimas

Prognozavimo metodų vertinimui naudojami duomenys: paspaudimų skaičius, atmetimo rodiklis (angl. *bounce rate*) ir vizito trukmė sekundėmis. Pasekminis rodiklis – paspaudimo skaičius yra svarbus svetaine, nuo paspaudimo skaičiaus priklauso konversijų skaičius. Taip pat, šis rodiklis labai priklauso nuo sezoniškumo bei priežasties rodiklių, kaip pozicija „Google“ organinėje paieškoje. Taikant tokius prognozės metodus kaip LPC ar ARMA įėjime ir išėjime naudojami tie patys duomenys – paspaudimo skaičius. Taigi, pagal paspaudimų skaičių praeityje galime vertinti

kokie jie bus ateityje, pagal tai planuoti biudžetus, reklamą ir panašiai. Taikant neuronų tinklus paspaudimų skaičius prognozuojamas pagal du įėjimus: atmetimo rodiklį ir vizito trukmę. Abu šie įėjimo rodikliai yra svarbus, nes pagal juos „Google“ algoritmai vertina svetainės atitikimą, pagal tai ko žmonės ieško, ką randa, jos naudingumą, pagal tai kiek žmonės praleidžia svetainėje laiko. Visa tai daro įtaką pozicijoms „Google“ organinėje paieškoje, kaip jau buvo minėta nuo pozicijos priklauso ir vizitų skaičius, o galiausiai ir konversijų.

Tyrimo ir analizės metu norima įvertinti prognozavimo metodus, gaunamas paklaidas kiekvienu konkrečiu atveju. Kaip vidutinė paklaida priklauso nuo prognozavimo metodo, lango pločio, mokymų skaičiaus ir atitinkamo prognozavimo metodo kintamųjų – su kokiais kintamaisiais ir prognozės metodais paklaida yra mažiausia.

„Bobutės paskola“ paspaudimų skaičius varijuoja nuo 64 iki 610, vidutinis paspaudimų skaičius 203, „Bobutės paskola“ be sezoniškų pikų paspaudimų skaičius varijuoja nuo 105 iki 281, vidutinis paspaudimų skaičius - 181. „Paskoliukas“ – nuo 30 iki 213, vidutinis paspaudimų skaičius – 119, „Popkreditas“ – nuo 38 iki 231, vidutinis paspaudimų skaičius – 121.

Visų pirma, skaičiuojama vidutinė kvadratinė paklaida, kad išvengtų minuso ženklo bei modulio taikymo. Iš kiekvienos reikšmės esančios realioje duomenų sekoje atimama prognozuojama reikšmė ir pakeliama kvadratu. Tuomet paskaičiuojamas visos paklaidos vidurkis, o norint gauti vidutinę paklaidą yra ištraukiama šaknis iš vidutinės kvadratinės sumos.

5.1. Tiesinės prognozės metodas (LPC)

Tiesinės prognozės metodui (LPC) naudojamas rodiklis – paspaudimų skaičius. Duomenys paimti nuo 2013 liepos 25 iki 2014 sausio 22 dienos imtinai. Kintamieji: lango plotis, filtro eilė. Lango plotis tyrimo metu buvo keičiamas nuo 8 iki 170 (antra kodo fragmento eilutė), o filtro eilė nuo 1 iki 8 keičiama rankiniu būdu perrašant ciklą priklausanti nuo p – filtro eilės. Ciklas keičiamas pagal funkciją $y(n)=a(2)*y(n-1)+a(3)*y(n-2)+...$ šeštoje kodo eilutėje ir skaičiuojant naują signalo dalį 9 kodo fragmento eilutėje. Tolimesnėse kodo eilutėse yra skaičiuojama ir kaupiama paklaida. Grįžtamojo ryšio koeficientai a atnaujinami imant naujus paskutinius lango pločio elementus (žr. 5.1 programos kodą, 13 kodo eilutę).

5.1 programos kodas

```
1. x=xlsread('BobutesPaskola.xls','Lapas1', 'V44:V130');
2. for wl=7:170
3. s=x(wl:-1:1);
4. p=8;% pasirenkama filtro eilė
5. a=lpc(s,p);
6. y=-
    (a(2)*s(1)+a(3)*s(2)+a(4)*s(3)+a(5)*s(4)+a(6)*s(5)+a(7)*s(6)+a(8)*s(7)
    +a(9)*s(8)); % veiks tik kai p=8,
7. e=x(idxl)-y;
```

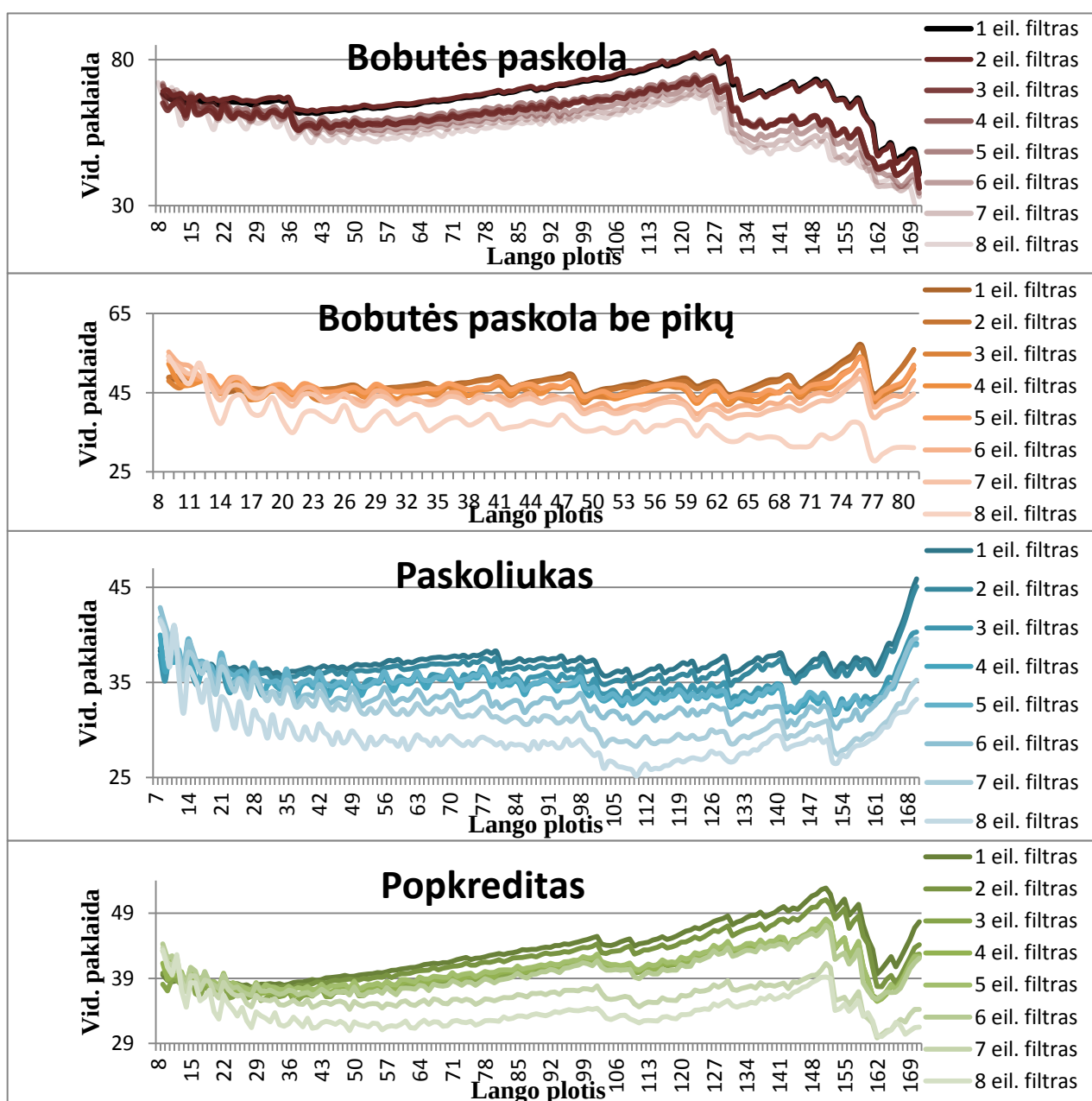
```

8. s2 = x(idx1:-1:(idx1-p)); % imama viena iteracija naujesnė signalo
   dalis, pakanka paskutiniųjų "p" reikšmių
9. y2=-
   (a(2)*s2(1)+a(3)*s2(2)+a(4)*s2(3)+a(5)*s2(4)+a(6)*s2(5)+a(7)*s2(6)+a(8)
   )*s2(7)+a(9)*s2(8)); % tie patys koeficientai, kitas signalas
10. e2=x(idx1+1)-y2;
11. kaupiu_e2 = [kaupiu_e2 e2];
12. kaupiu=[kaupiu y2];
13. s=x(idx1:-1:idx1-w1);
14. a=lpc(s,p);

```

5.1.1. LPC metodo efektyvumo priklausomybių nuo lango pločio tyrimas

Lango plotis buvo keičiamas nuo 8 iki 170, kiekvienai filtro reikšmei nuo 1 iki 8. Tiriant „Bobutės paskola“ be sezoniškumo, lango plotis buvo keičiamas nuo 8 iki 80. 5.2 paveiksle grafikuose pateikta vidutinės paklaidos kitimas priklausomai nuo lango pločio ir nuo filtro eilės.



5.2 pav. Vidutinės paklaidos priklausomybė nuo LPC modelio filtro eilės ir nuo lango pločio

Pagal 5.2 paveikslą matoma, kad didėjant lango pločiui iki tam tikros ribos (riba skirtinga kiekvieno projekto atveju – vidutiniškai iki 30 – 35 reikšmės), paklaida mažėja. Kai langas yra maždaug apie 30 – 35 paklaida gaunama viena mažiausiu ir toliau ji pradeda didėti. Paklaida didėja, nes duomenys yra vertinami iš apmokytų mėnesio laiko duomenų, bet kitus mėnesius vertės pasikeičia labiau dėl sezoniškumo, į apmokymą papuola tam tikri vienkartiniai duomenų pikai, kurie vėliau pagal tendenciją nesikartoja, ir tai didina paklaidą. Kai duomenų apmokymui yra skiriama labai daug apie 160 reikšmių, paklaida mažėja, nes prognozuojamų reikšmių ir paklaidų verčių lieka nebe daug, todėl vidutinė paklaida tampa labai maža visais atvejais. Kuo filtro eilė didesnė, tuo reikšmei prognozuoti imama daugiau praeities verčių ir vidutinė paklaida tampa mažesnė.

„Bobutės paskola“ mažiausia vidutinė paklaida (kai yra prognozuojama pakankamai daug ateities reikšmių) – 51,6 gaunama kai lango plotis 37 (filtro eilė – 8), „Bobutės paskola“ be sezoniškumo mažiausia paklaida tendencingai mažėja visą laiką – 28,1 (filtro eilė – 8), kai lango plotis 76, „Paskoliukas“ 25,2 – 110 (filtro eilė – 8), „Popkreditas“ 31 – 48.

5.1.2. LPC metodo efektyvumo priklausomybių nuo filtro eilės tyrimas

Kaip jau buvo minėta, vienas iš kintamųjų LPC metode yra filtro eilė, kuris tyrimo metu buvo keičiamas nuo 1 iki 8.

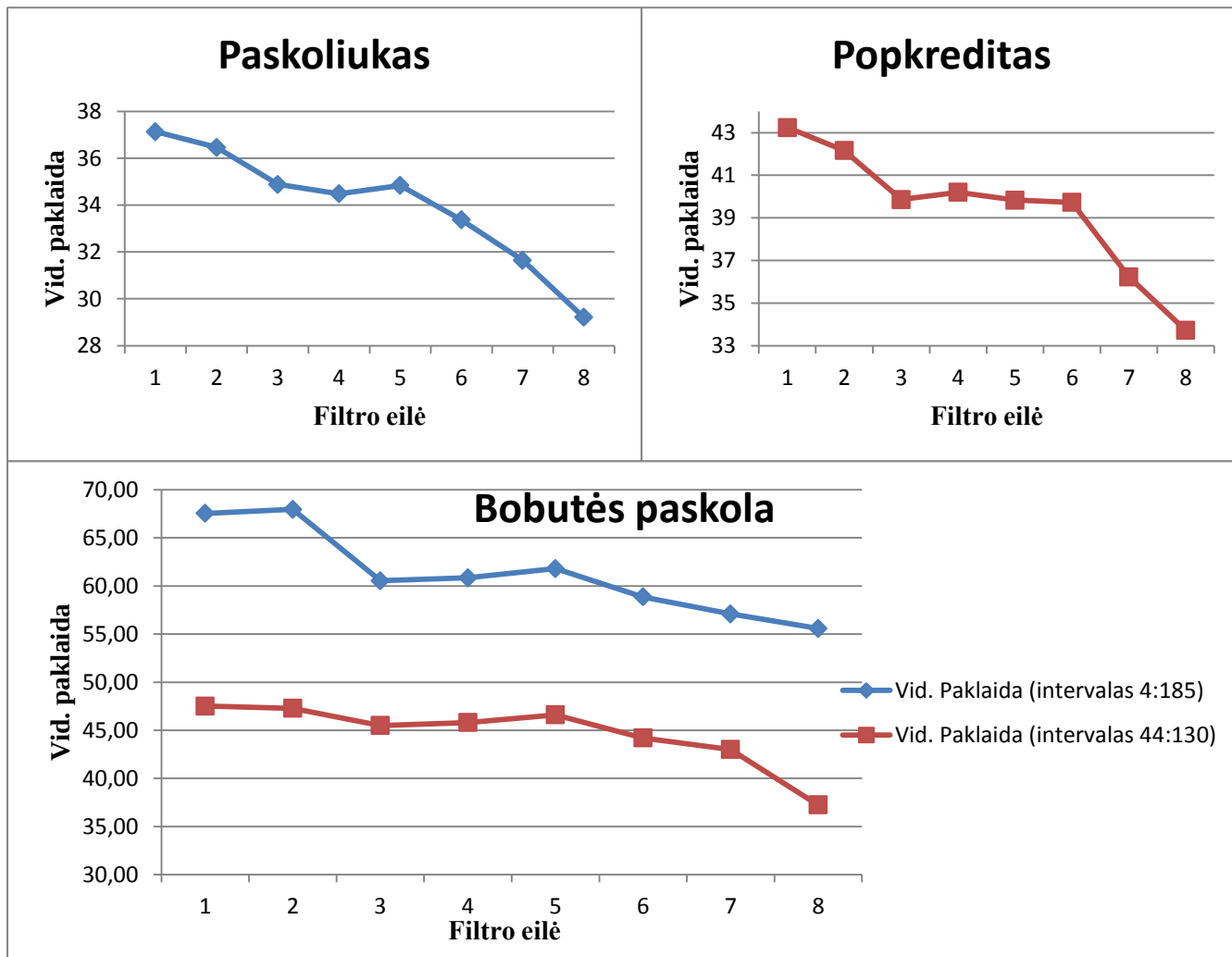
5.1 lentelėje pateikta vidutinė paklaida, kai naudojami skirtingos eilės filtrai.

5.1 lentelė. Vidutinės paklaidos priklausomybė nuo filtro eilės, taikant LPC metodą

Filtro eilė	1	2	3	4	5	6	7	8
Vid. Paklaida (paskoliukas)	37,1	36,5	34,9	34,5	34,8	33,4	31,7	29,2
Vid. Paklaida (popkreditas)	43,2	42,2	39,9	40,2	39,8	39,7	36,2	33,7
Vid. Paklaida (bobutės paskola)	67,6	68,0	60,6	60,9	61,8	58,9	57,1	55,6
Vid. Paklaida (bobutės paskola – be sezoniškų pikų)	47,5	47,3	45,5	45,8	46,6	44,2	43,0	37,2

Mažiausia vidutinė paklaida – 29,2 yra gaunama, kai filtro eilė yra 8 projekto – „Paskoliukas“ atveju, tai yra 24,3 % paklaida skaičiuojant nuo vidutinio paspaudimo skaičiaus (119). „Paskoliukas“ ir „Popkreditas“ atvejais mažiausia vidutinė paklaida gaunama kai filtro eilė yra 8, „Popkreditas“ vidutinė paklaida – 33,7, tai 27,9 % nuo vidutinio paspaudimo skaičiaus (121).

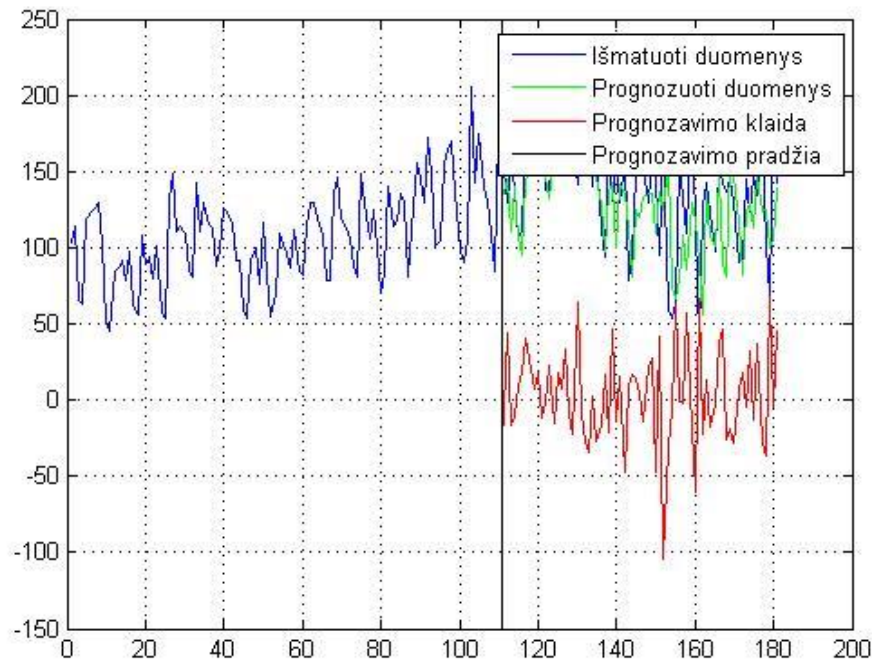
„Bobutės paskola“ vidutinė paklaida mažiausia, kai filtro eilė 8, bendruoju atveju paklaida – 55,6, tai 27,4 % nuo vidutinio paspaudimo skaičiaus (203). 5.3 paveiksle pateikta vidutinės paklaidos priklausomybė nuo filtro eilės.



5.3 pav. Vidutinės paklaidos priklausomybė nuo filtro eilės, taikant LPC modelį

Grafikuose vizualiai matosi lentelės duomenys, kaip filtro eilė daro įtaką vidutinei paklaidai. Didėjant filtro eilei, ne tiesiškai mažėja vidutinė paklaida.

5.4 paveiksle pateiktas pavyzdys kaip kinta paklaida konkrečiu „Paskoliukas“ atveju, kai filtro eilė 8, o lango plotis yra 110.



5.4 pav. „Paskoliukas“ realiųjų duomenų, išmatuotų ir paklaidos kitimas, taikant LPC modelį

5.8 paveiksle mėlyna linija atvaizduoti išmatuoti duomenys, žalia – prognozuoti duomenys, raudona – prognozavimo klaida, o juoda linija – prognozavimo pradžia.

Daroma prielaida, kad skirtingose signalo prognozės vietose klaida kinta nevienodai, todėl atliktas papildomas palyginimas prognozuojant tik paskutiniąsias 30 signalo verčių, esant skirtingiems lango pločiams. Lango pločio vertės imamos nuo 18 iki 150, kas keturias reikšmes, o filtro eilė keičiama nuo 1 iki 16. Didėjant filtro eilei paklaida mažėja. Kai filtro eilė 1 ir 2 paklaida kinta labai mažai, didint lango pločiui (beveik pastovi). „Bobutės paskola“ atveju mažiausia paklaida gauta 49,7 (24,5 %), kai filtro eilė 10, o lango plotis – 146. „Paskoliukas“ – 28,3 (23,9 %), kai filtro eilė yra 11, o lango plotis – 122. „Popkreditas“ – 38,5(31,9 %), kai filtro eilė yra 10, o lango plotis – 146.

5.2. Autoregresinis paslinktųjų vidurkių prognozavimo metodas (ARMA)

Taikant autoregresinį paslinktųjų vidurkių prognozavimo metodą (ARMA) buvo naudojamas rodiklis – paspaudimų skaičius. Duomenys paimti nuo 2013 liepos 25 iki 2014 sausio 22 dienos imtinai. Kintamieji: lango plotis, modelio eilė. Lango plotis tyrimo metu buvo keičiamas nuo 6 iki 170 imant kas antrą reikšmę, o modelio eilė buvo keičiama: „Bobutės paskola“ atveju nuo [1 1] iki [5 5], kitiems projektams nuo [1 1] iki [3 3] rankiniu būdu. 5.2 kodo fragmente pateikta ARMA metodo programinė dalis.

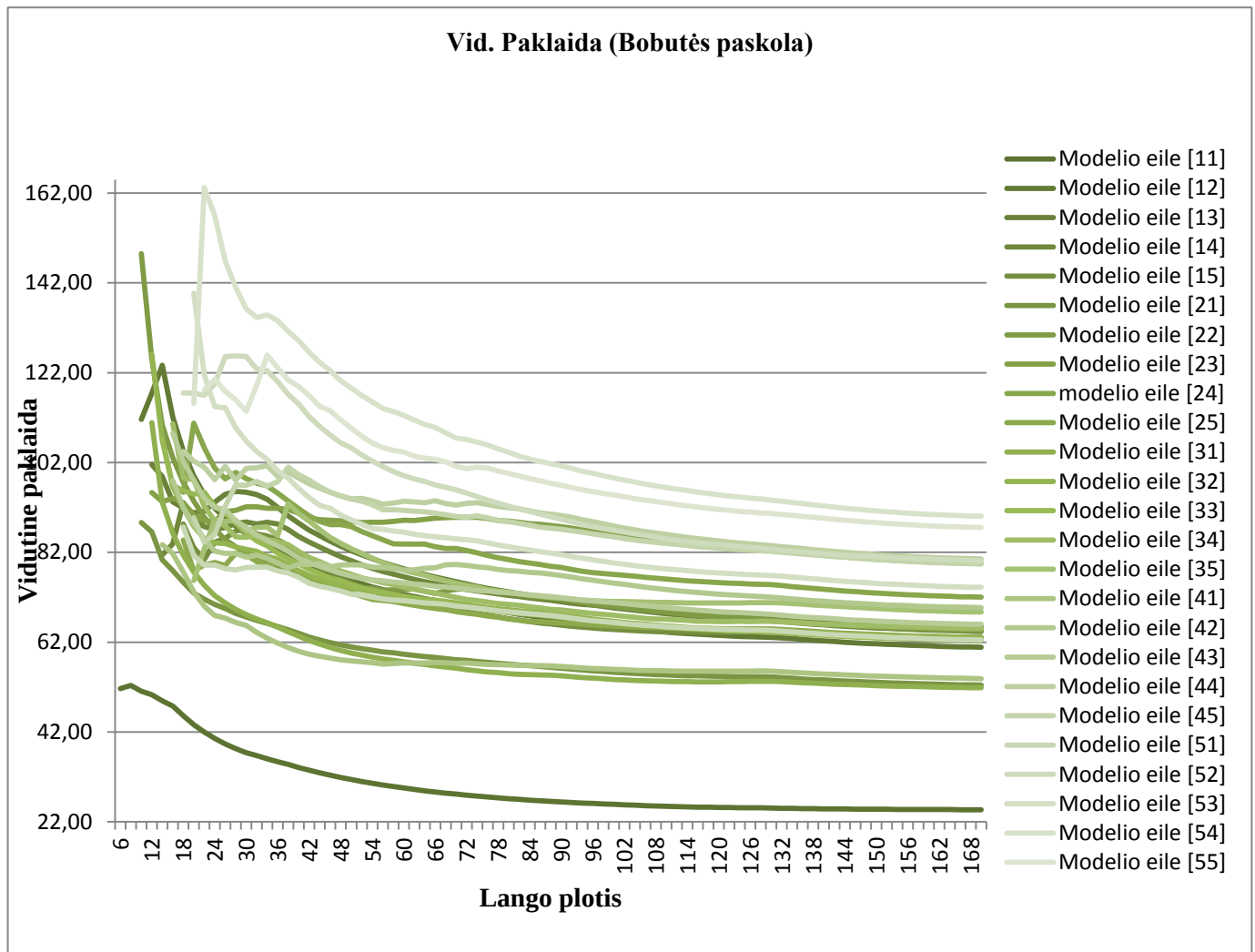
5.2 kodo fragmente pirmojoje eilutėje nuskaitomi duomenys, antroje – kuriamas ciklas lango pločiui nuo 16 iki 170, esant mažesnei modelio eilei pradinis langas gali būti mažesnis. 4 ir 5 eilutėse nustatoma daugianario eilė, kuri tyrimo metu buvo keičiama rankiniu būdu. Vėliau sukuriamas signalo fragmentas modeliui apskaičiuoti ir 7 kodo eilutėje pritaikoma MATLAB® funkcija `armax`, kuri apskaičiuoja ARMA modelio parametrus, naudoja laike kintančius nurodytus duomenis, įėjime nurodoma modelio eilė ir duomenys pradiniam lango plote, pagal kurį atliekamas apmokymas ir prognozavimas. 8 eilutėje nurodoma, kiek žingsnių bus prognozuojama į priekį, šiuo atveju – vienas. Toliau atliekamas duomenų prognozavimas su funkcija `forecast`, kuri naudoja `armax` funkcijos gautos mokymo rezultatus. Kitose eilutėse skaičiuojama ir kaupiama vidutinė paklaida.

5.2 programos kodas

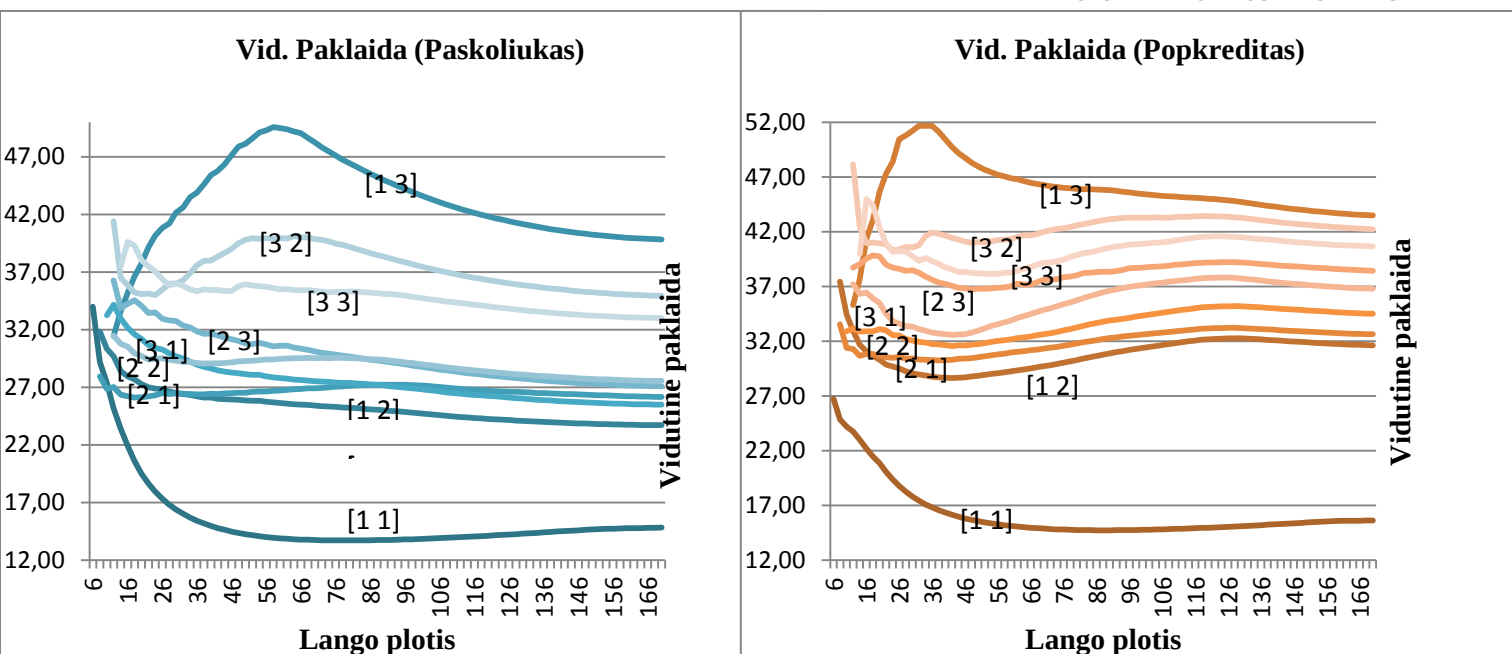
```
1. x=xlsread('BobutesPaskola.xls','Lapas1','V4:V185');
2. for wl=16:2:170
3. % modelio sudarymas
4. na = 2; % A daugianario eilė
5. nb = 4; % B daugianario eilė
6. s=x(1:wl); % signalo fragmentas modeliui apskaičiuoti
7. sys = armax(s,[na nb]);
8. K=1; % Nurodo kiek žingsnių prognozuos į priekį
9. for idx1=(wl+1):(numel(x)-1)
10. y = forecast(sys,x((idx1-wl):idx1-1),K);
11. e=x(idx1-1)-y;
12. kaupiu_e2 = [kaupiu_e2 e];
13. kaupiu=[kaupiu y];
```

5.2.1. ARMA metodo efektyvumo priklausomybių nuo lango pločio tyrimas

Lango plotis buvo keičiamas nuo 6, 8, 10, 12 iki 170, priklausomai nuo modelio eilės. Lango plotis buvo imamas kas antras, nes vidutinė paklaida kinta ne taip staigiai, kad reikėtų visų duomenų, ir taip pat dėl ribotų resursų, kadangi programa veikė ganėtinai ilgai. 5.5 ir 5.6 grafikuose vidutinės paklaidos kitimas priklausomai nuo lango pločio ir nuo modelio eilės. Kadangi programa veikia labai ilgai, nebuvo atliekamas tyrimas „Bobutės paskola“ be sezoniškų pikų, bet tendencija išlieka ta pati, tik vidutinė paklaida numatoma mažesnė.



5.5 pav. Vidutinės paklaidos priklausomybė nuo ARMA modelio eilės ir nuo lango pločio („Bobutės paskola“)

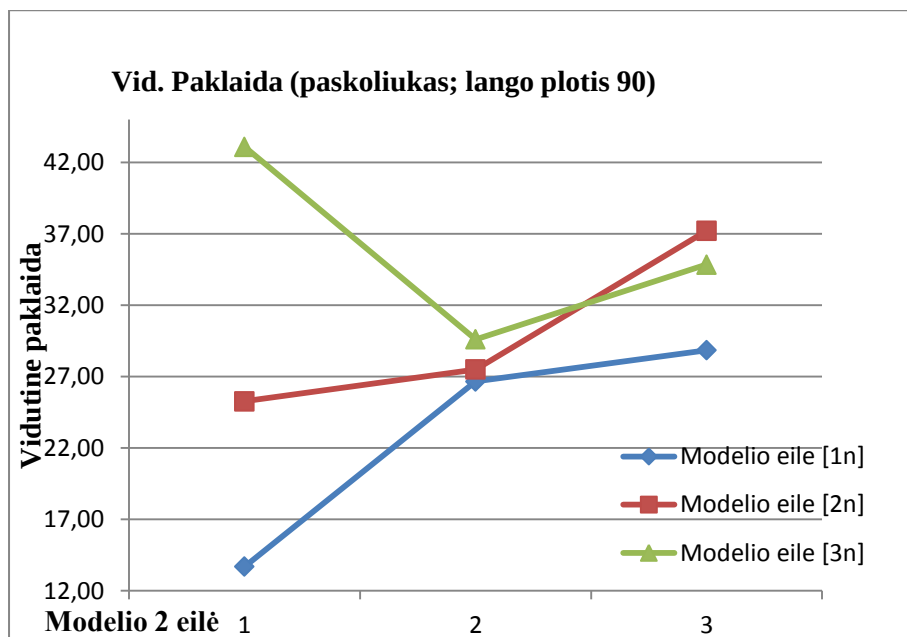


5.6 pav. Vidutinės paklaidos priklausomybė nuo ARMA modelio eilės ir nuo lango pločio („paskoliukas“ ir „popkreditas“)

Kuo modelio eilė mažesnė, tuo reikalingas mažesnis lango plotis, prie kurio vidutinė paklaida mažesnė. Esant didesniai modelio eilės numeriui, reikia daugiau istorinių duomenų, kad gauti mažesnę paklaidą. Mažiausia „Paskoliukas“ paklaida gaunama, kai lango plotis yra 76 – 13,70, tai sudaro apie 11,51 % nuo bendro paspaudimo skaičiaus vidurkio. „Bobutės paskola“ mažiausia paklaida 24,68 gaunama, kai lango plotis 170, o tai yra 12,16 %. Šiuo atveju toks platus langas reikalingas, nes apmokymą apsunkina minėtieji duomenų šuoliai. „Popkreditas“ mažiausia vidutinė paklaida 14,72 (12,17 %) gaunama kai lango plotis yra 84. Mažiausios paklaidos skiriasi nedaug, tik kai duomenų kitimas yra stabilesnis, pakanka kur kas mažiau duomenų apmokymui, norint gauti mažiausią paklaidą – šiuo atveju apie trijų mėnesių laikotarpį. „Bobutės paskola“ atveju kuo istorinių duomenų daugiau, tuo paklaida mažesnė, ir priimtina prielaida, kad turint bent metų senumo istorinius duomenis, kurių dėka būtų galima įvertinti tendenciją, sezoniskumą – paklaida būtų kur kas mažesnė.

5.2.2. ARMA metodo efektyvumo priklausomybių nuo modelio eilės tyrimas

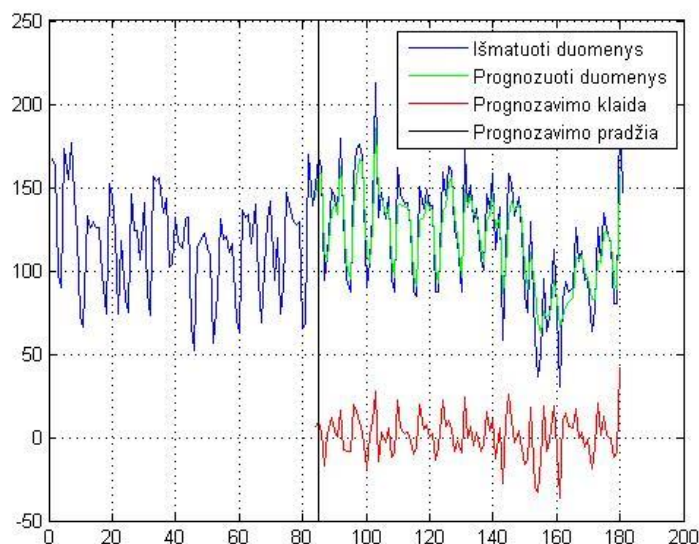
Iš 5.5 ir 5.6 paveiksle akivaizdu, kad didėjant modelio eilei vidutinė paklaida taip pat didėja. Taip pat matoma tendencija kai pirmas daugianaris – 1, o antras daugianaris yra didinamas, paklaida didėja gana dideliais žingsniais. Pavyzdžiui „Paskoliukas“ atveju didėja nuo 13,70 iki 49,57 kai modelio eilė kinta: [1 1], [1 2] ir [1 3]. Kai modelio eilė [2 n], čia $n = 1, 2, 3$ (antrojo daugianario eilė), paklaida kinta mažesniame intervale nuo 26,11 iki 27,22. Esant modelio eilei – [3 n], vidutinė paklaida taip pat kinta siaurame diapazone, bet šiek tiek didesniame: nuo 27,54 iki 41,42. Ši tendencija gana aiškiai matoma 5.7 pav., kai lango plotis 90, projektas „Paskoliukas“.



5.7 pav. Vidutinės paklaidos priklausomybė nuo ARMA modelio daugianarių (lango plotis 90), „Paskoliukas“

Iš 5.7 paveikslo matome, kad mažiausia paklaida gaunama kai modelio eilė mažiausia [1 1]. Kai daugianariai [1 n], didinant n paklaida labai ženkliai didėja, kai daugianaris [2 n], mažiausia paklaida kur kas didesnė, bet didžiausia paklaida sumažėja, analogiškai ir su [3 n].

5.8 paveiksle pateiktas pavyzdys kaip kinta paklaida konkrečiu „Paskoliukas“ atveju, kai modelio eilė [1 1], o lango plotis yra 90.



5.8 pav. „Paskoliukas“ realių duomenų, išmatuotų ir paklaidos kitimas taikant ARMA modelį

5.8 paveiksle mėlyna linija atvaizduoti išmatuoti duomenys, žalia – prognozuoti duomenys, raudona – prognozavimo klaida, o juoda linija – prognozavimo pradžia.

5.3. Netiesinis autoregresinis neuronų tinklas (NAR)

Taikant NAR metodą, kintamieji yra: pirmo sluoksnio neuronai, išėjimų vėlinimas bei mokymų skaičius. Prognozuojamas yra paspaudimų skaičius, įėjimui naudojant taip pat paspaudimų istorinius duomenis. Analizuojant „Bobutės paskolą“, pirmo sluoksnio neuronų skaičius buvo keičiamas nuo 1 iki 8, išėjimo vėlinimas nuo 6 iki 18, mokymai buvo kartojami po 20 kartų. Visiems atvejams buvo skaičiuojama vidutinė paklaida, rezultatų gauta 2080. Dėl ilgo skaičiavimo proceso, resursų trūkumo, kitiems projektams kintamųjų skaičių intervalai buvo sumažinti: pirmo sluoksnio neuronų skaičius buvo keičiamas nuo 1 iki 6, išėjimo vėlinimas – nuo 6 iki 10, mokymai kartojami 10 kartų, rezultatų gautą – 540.

Pirmoje eilutėje nuskaitomi duomenys, iš kurių paruošiamas teisingų atsakymų vektorius (2 kodo eilutė). Trečioje eilutėje sukuriamas tuščia matrica, į kurią bus kaupiami duomenys. Tuomet yra kuriami `for` ciklai pirmo sluoksnio neuronų skaičiams ir išėjimų vėlinimui (5.3 programos kodo 4 ir 5 eilutės). Pritaikomas prognozavimo modelis, nustatomas validacijos skaičius ir su funkcija `preparets` paruošiami įėjimo duomenys – atrenkami įėjimo pavyzdžiai pagal tai, kiek jie turi būti suvėlininti išėjimo atžvilgiu (žr. 5.3 programos kode 6 – 8 eilutės). Mokymas vykdomas 20 kartų naudojant funkciją `train`, rezultatai pritaikomi sukurtam neuronų tinklui ir pabaigoje skaičiuojama bei kaupiama vidutinė paklaida (5.3 programos kodo 9 – 13 eilutės).

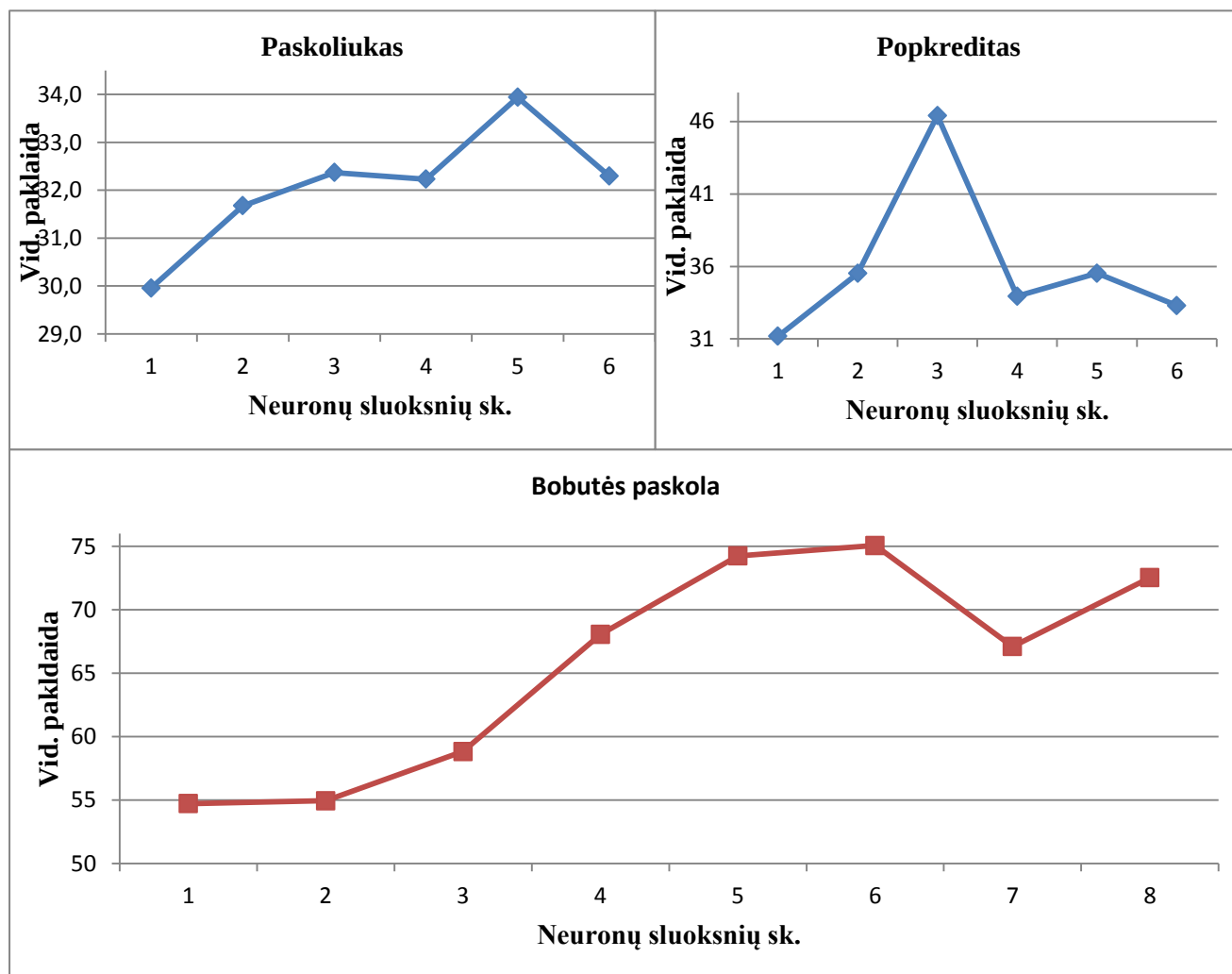
5.3 programos kodas

```
1. y=xlsread('BobutesPaskola.xls','Lapas1','V4:V185');
2. T=mat2cell(y',1,ones(1,numel(y))');
3. kaupiu_perf=[];
4. for sluoksnio_neuronai=1:8
5. for isejimu_vėlinimas=6:18
6. net = narxnet(1:isejimu_vėlinimas,[sluoksnio_neuronai]);
7. net.trainParam.max_fail=40;
8. [Xs,Xi,Ai,Ts] = preparets(net,{}, {},T);
9. for mokymas=1:20
10. net = train(net,Xs,Ts,Xi,Ai);
11. Y = net(Xs,Xi);
12. perf = perform(net,Ts,Y);
13. kaupiu_perf=[kaupiu_perf,perf];
14. end
15. end
16. end
```

Pagal gautus rezultatus buvo analizuojama jų priklausomybė nuo sluoksnio neuronų skaičiaus, išėjimo vėlinimo ir mokymų skaičiaus.

5.3.1. NAR metodo efektyvumo priklausomybių nuo pirmo sluoksnio neuronų skaičiaus tyrimas

Tiriant NAR metodo efektyvumo priklausomybių nuo pirmo sluoksnio neuronų skaičiaus buvo matuojama vidutinė paklaida, skaičiuojama visų mokymų ir visų vėlinimo variacijų vidutinė paklaida tam tikram neuronų skaičiui. „Bobutės paskola“ atveju neuronų skaičius buvo keičiamas nuo 1 iki 8, kitų projektų sumažinta iki 6 dėl ilgos programos veikimo trukmės. 5.9 paveiksle atvaizduota vidutinės paklaidos priklausomybė nuo neuronų skaičiaus.



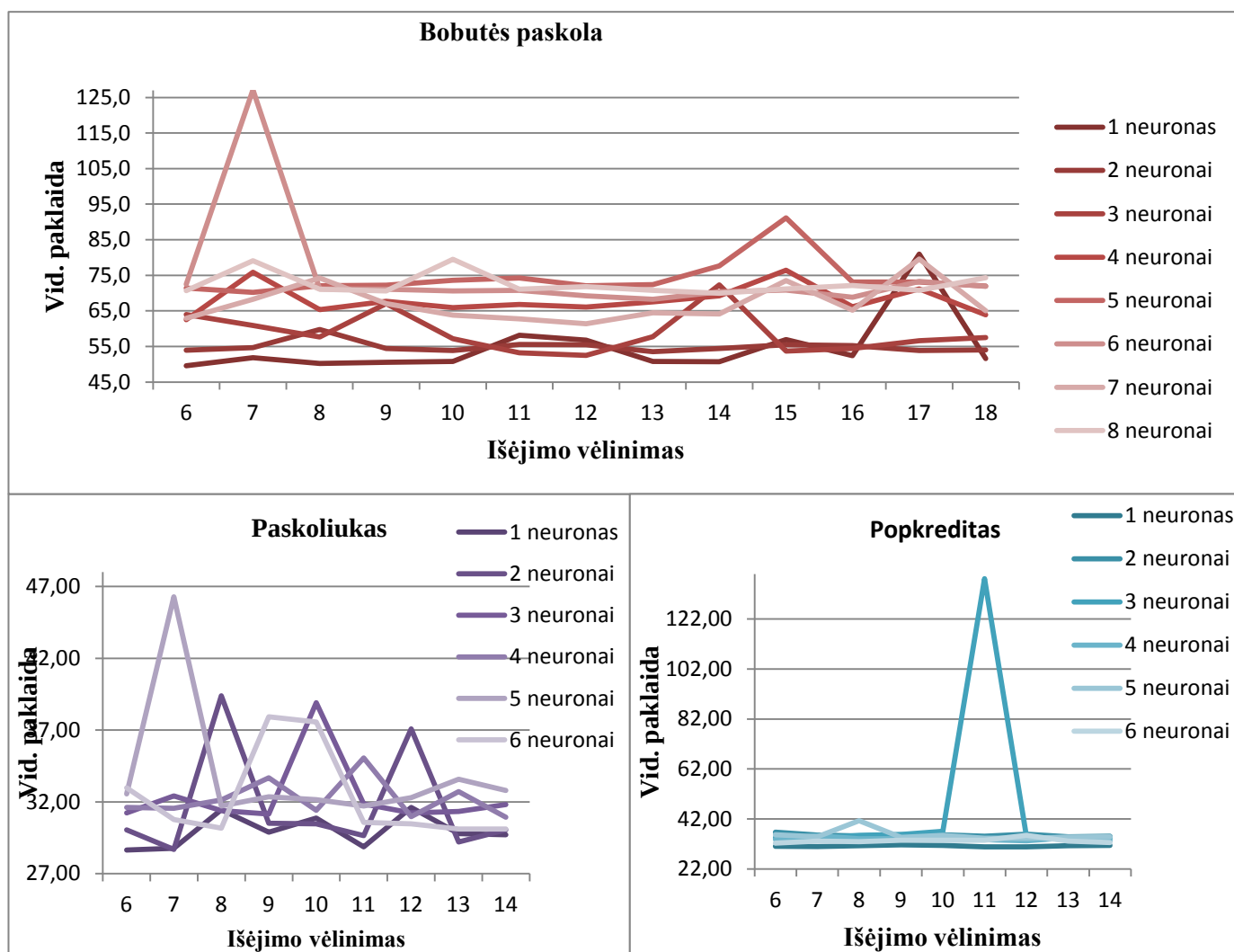
5.9 pav. Vidutinės paklaidos priklausomybė nuo pirmo sluoksnio neuronų skaičiaus, taikant NAR modelį

„Paskoliukas“ ir „Bobutės paskola“ atveju didinant neuronų skaičių vidutinė paklaida didėja, tik, kai neuronų skaičius atitinkamai 6 ir 7, vidutinė paklaida šiek tiek sumažėja, dėl apmokymų klaidų. „Popkreditas“ atveju vidutinė paklaida svyruoja keičiant neuronų skaičių, kai neuronų yra 3, matomas didžiulis šuolis, didžiausia tikimybė, kad tai tik apmokymo klaidos ir nestabilumas. Bendra tendencija, kad mažiausia vidutinė paklaida yra, kai naudojamas tik vienas neuronas.

„Bobutės paskola“ atveju mažiausia vidutinė paklaida, kai neuronų skaičius 1 – 54,7, tai yra 27 %. Didžiulė paklaida gali būti dėl to, kad šis mokymo prognozavimo metodas labiausiai pritaikytas, kai duomenų sekoje nėra didžiulių pokyčių, šuolių, ar kritimų. „Paskoliukas“ atveju mažiausia vidutinė paklaida – 30, kas sudaro apie 25,2 % vidutinio paspaudimo skaičiaus. „Popkreditas“ – 31,18 (25,8 %).

5.3.2. NAR metodo efektyvumo priklausomybių nuo išėjimo vėlinimo tyrimas

„Bobutės paskola“ bandymų metu išėjimo vėlinimas buvo keičiamas nuo 6 iki 18, analizuojant kitus projektus – nuo 6 iki 14 (žr. 5.10 pav.).



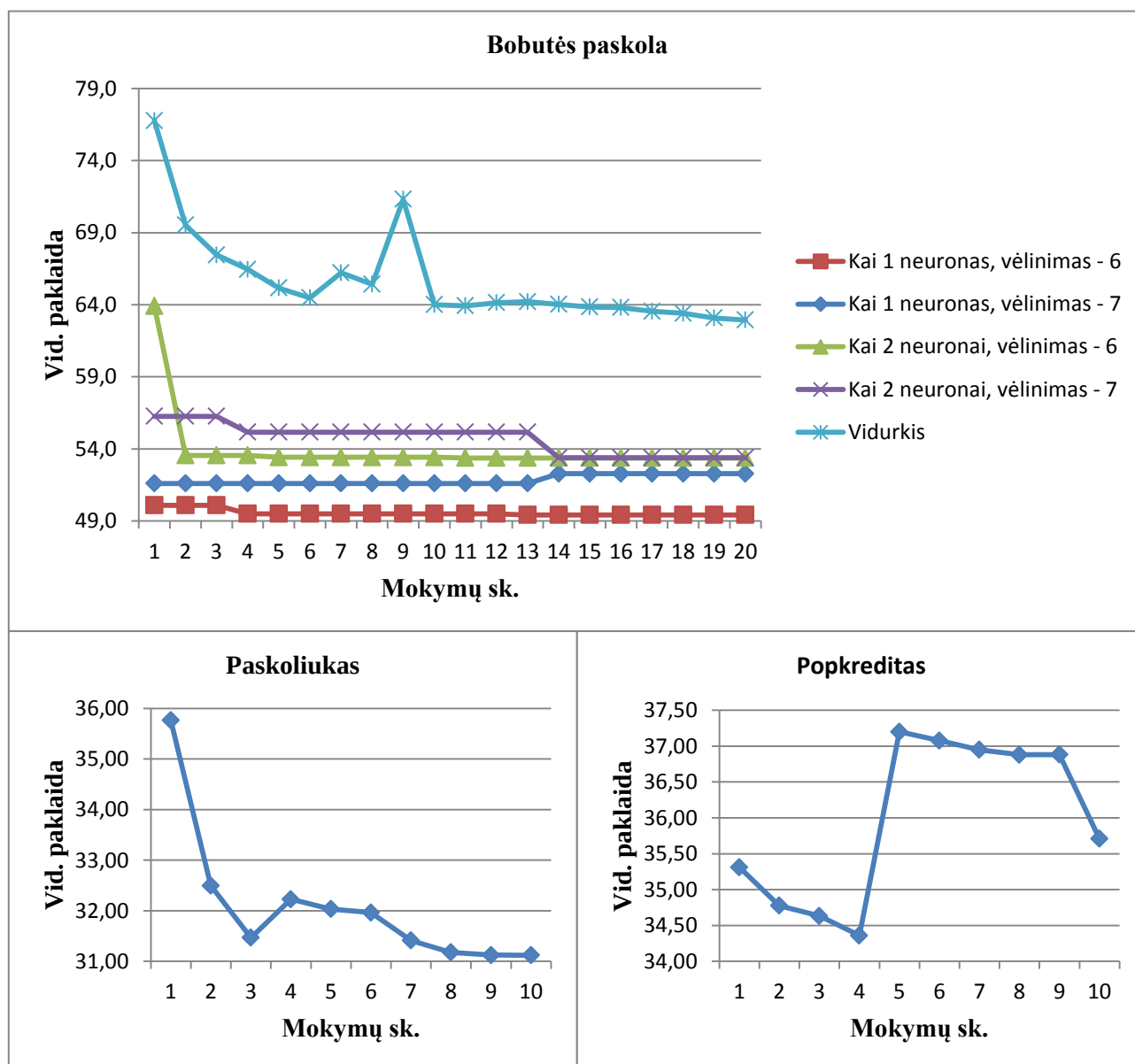
5.10 pav. Vidutinės paklaidos priklausomybė nuo išėjimo vėlinimo, taikant NAR modelį

5.10 paveiksle pateiktas vidutinės paklaidos kitimas, kai yra keičiamas išėjimo vėlinimas ir yra skirtingas neuronų skaičius. Pagal bendras tendencijas galima daryti prielaidą, kad didelės įtakos išėjimo vėlinimas šiuo atveju neturi, vidutinės paklaidos svyruoja panašiam diapazone. Taip pat, esant didesniui neuronų skaičiui ir didesnei išėjimo vėlinimo vertei paklaidos svyravimas yra šiek tiek mažesnis. Kai neuronų skaičius ir išėjimo vėlinimo vertė mažesnė, vidutinės paklaidos gaunamos mažiausios. Kai neuronų skaičius yra 1, išėjimo vėlinimas – 1, analizuojant „Bobutės

paskolą“ vidutinė paklaida gaunama 49,6 (24,4 %); „Paskoliuką“ – 28,65 (24,1 %); „Popkreditą“ – 31,14 (25,7 %).

5.3.3. NAR metodo efektyvumo priklausomybių nuo mokymų skaičiaus tyrimas

„Bobutės paskola“ bandymų metu, mokymų skaičius buvo keičiamas nuo 1 iki 20, analizuojant kitus projektus mokoma buvo po 10 kartų. 5.10 paveiksle grafiškai atvaizduota vidutinės paklaidos priklausomybė nuo mokymų skaičiaus. „Paskoliuko“ ir „Popkredito“ dalyse atvaizduota tik vidutinė paklaida, o „Bobutės paskolos“ atvaizduota vidutinės paklaidos vidurkis bei keli konkretūs variantai vidutinės paklaidos, kai atitinkamai neuronų skaičius yra 1 ir 2 bei išėjimo vėlinimas 6 ir 7.



5.11 pav. Vidutinės paklaidos priklausomybė nuo išėjimo vėlinimo, taikant NAR modelį

Bendra tendencija imant individualius atvejus, ar vidurkį, vidutinė paklaida mažėja kai mokymų skaičius didėja (žr. 5.11 pav.). Apmokymo metu naudojant „Popkredito“ duomenis,

vykdant penktą mokymą, matomas didžiulis paklaidos šuolis, kuris ir padarė įtaką bei iškreipė visus rezultatus, vidurkius

Vidutinės paklaidos yra gana apylygės ir didelės, tai galima daryti prielaidą, kad šis metodas nėra pats tinkamiausias prognozuoti tokio tipo duomenims, arba yra reikalinga ilgesnė duomenų istorija bei kur kas daugiau bandymų.

5.4. Netiesinis autoregresinis su papildomų įėjimu neuronų tinklas (NARX)

Taikant NARX metodą buvo atliekami du bandymai, naudojant tik vieną neuronų sluoksnį ir naudojant du sluoksnius. Taikant NARX metodą kintamieji yra: pirmo sluoksnio neuronai, antro sluoksnio neuronai išėjimų vėlinimas bei mokymų skaičius.

Prognozuojamas yra paspaudimų skaičius išėjime, o įėjimai naudojami du: atmetimo rodiklis ir vidutinė vizito trukmė svetainėje. Pagal įėjimo duomenis, jų tendenciją, prognozuojamas paspaudimų skaičius.

Analizuojant „Bobutės paskolą“, pirmo sluoksnio neuronų skaičius buvo keičiamas nuo 3 iki 8, įėjimo vėlinimas – nuo 1 iki 17, išėjimo vėlinimai – nuo 2 iki 8, mokymai buvo kartojami po 20 kartų. Identiški mokymai buvo atliekami imant mažesnę duomenų diapazoną, kur nėra įtraukiami sezoniški pikai. Visiems atvejams buvo skaičiuojama vidutinė paklaida, rezultatų gauta 14280, taikant vieną neuronų sluoksnį, ir 114240, taikant du neuronų sluoksnius.. Dėl ilgo skaičiavimo proceso, resursų trūkumo, kitiems projektams kintamųjų skaičių intervalai buvo sumažinti: pirmo sluoksnio neuronų skaičius buvo keičiamas nuo 3 iki 8, įėjimo vėlinimas – nuo 1 iki 10, išėjimo vėlinimas nuo 2 iki 8, mokymai kartojami 10 kartų, rezultatų gauta 4200, taikant vieną neuronų sluoksnį ir 33600, taikant du neuronų sluoksnius.

5.4 programos kodo 1 – 3 eilutės nuskaitomi duomenys, y – bendras paspaudimų skaičius, x_1 – atmetimo rodiklis, x_2 – vidutinė vizito trukmė. x – paruoštas 2 įėjimų pavyzdžių matrica iš dviejų rodiklių, T – paruošiamas teisingų atsakymų vektorius, $kaupiu_perf$ – paruošiamas kintamasis saugoti tinklo vidutinės kvadratinės paklaidos reikšmės (4 – 6 eilutės). Tuomet yra kuriami keturi `for` ciklai, pirmasis – eksperimentuojama, kai pirmajame sluoksnyje esančių neuronų skaičius yra nuo 3 iki 8, antrasis – eksperimentuojama, kai antrajame sluoksnyje esančių neuronų skaičius yra nuo 1 iki 8, trečiasis – eksperimentuojama, kai keičiamas įėjimo vėlinimas (kai įėjimo vėlinimas yra 1, imamos dabartinės rodiklių reikšmės), ketvirtasis – eksperimentuojama, kai imamos praeitos išėjimų y reikšmės (žr. 5.4 programos kodą 7 – 10 eilutes). Su nustatytais parametrais yra sukuriamas `net` tinklas, kuriam priskiriamas validacijos skaičius 40. Taikant funkciją `preparets` paruošiami įėjimo duomenys – atrenkami įėjimo pavyzdžiai pagal tai, kiek jie turi būti suvėlinti išėjimo atžvilgiu (žr. 5.4 programos kode 11 – 13 eilutės). Mokymas vykdomas 20 kartų naudojant

funkciją `train`, rezultatai pritaikomi sukurtam neuronų tinklui ir pabaigoje skaičiuojama bei kaupiama vidutinė paklaida (5.4 programos kodo 14 – 23 eilutės).

5.4 programos kodas

```

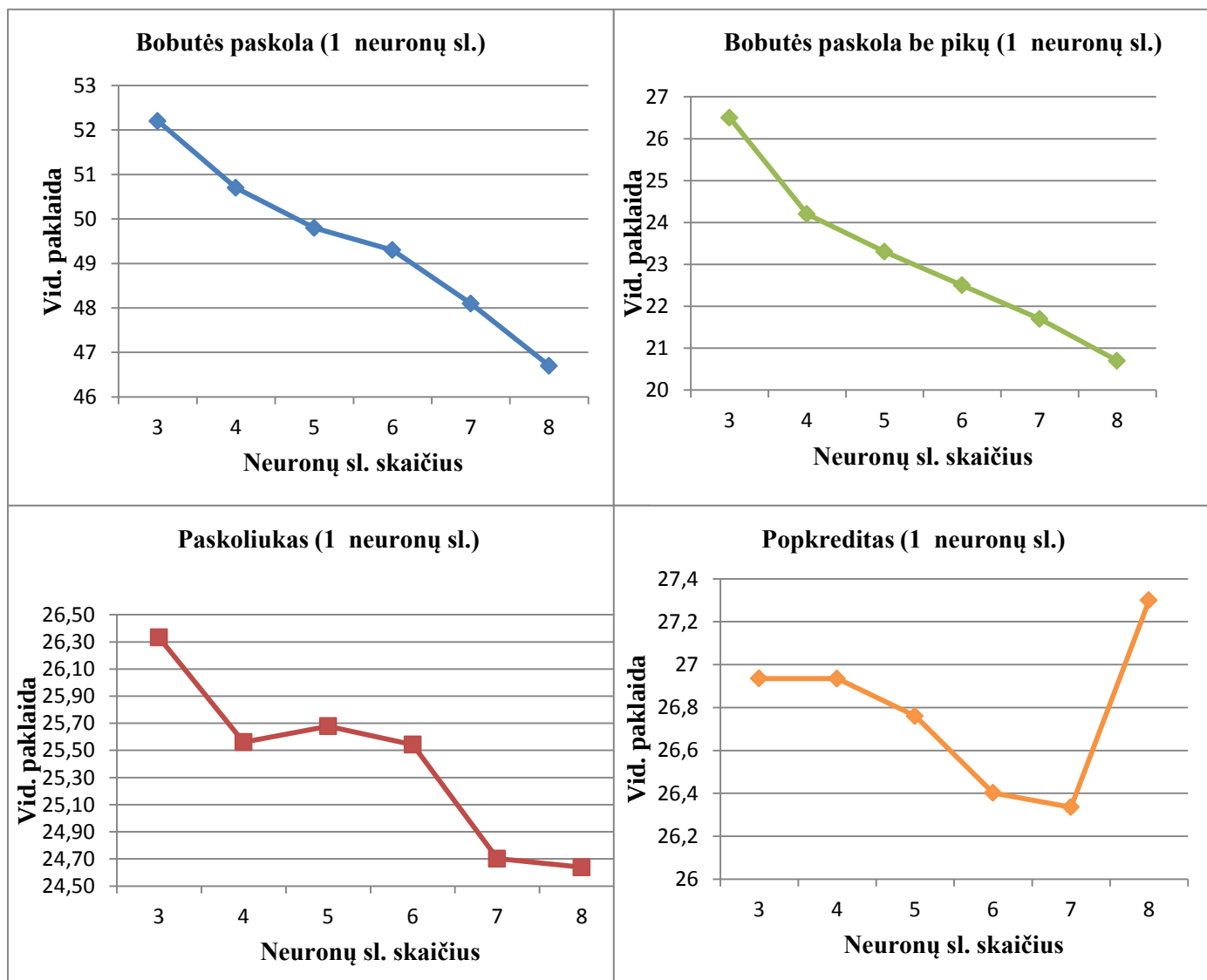
1. y=xlsread('BobutesPaskola.xls','Lapas1', 'V44:V130');
2. x1=xlsread('BobutesPaskola.xls','Lapas1', 'B44:B130');
3. x2=xlsread('BobutesPaskola.xls','Lapas1', 'C44:C130');
4. X=mat2cell([x1';x2'],2,ones(1,numel(x1)));
5. T=mat2cell(y',1,ones(1,numel(y)));
6. kaupiu_perf=[];
7. for sluoksnio_neuronai=3:8
8. for antro_sluosnio_neuronai=1:8
9. for iejimu_velinimas=1:17
10. for isejimu_velinimas=2:8
11. net=narxnet(1:iejimu_velinimas,1:isejimu_velinimas,[sluoksnio_neuronai
    antro_sluosnio_neuronai]);
12. net.trainParam.max_fail=40;
13. [Xs,Xi,Ai,Ts] = preparets(net,X,{},T);
14. for mokymas=1:20
15. net = train(net,Xs,Ts,Xi,Ai)
16. Y = net(Xs,Xi,Ai)
17. perf = perform(net,Ts,Y); % vidutinė klaida?
18. kaupiu_perf=[kaupiu_perf,perf];
19. end
20. end
21. end
22. end
23. end

```

Pagal gautus vidutinės paklaidos rezultatus, analizuojama jų priklausomybė nuo sluoksnio neuronų skaičiaus, išėjimo vėlinimo, įėjimo vėlinimo ir mokymų skaičiaus.

5.4.1. NARX metodo efektyvumo priklausomybių nuo pirmo sluoksnio neuronų skaičiaus tyrimas

Tiriant NARX metodo efektyvumo priklausomybių nuo pirmo sluoksnio neuronų skaičiaus buvo matuojama vidutinė paklaida, skaičiuojama visų mokymų ir visų vėlinimo variacijų vidutinė paklaida tam tikram neuronų skaičiui. Neuronų skaičius buvo keičiamas nuo 3 iki 8. 5.12 paveiksle atvaizduota vidutinės paklaidos priklausomybė nuo neuronų skaičiaus.



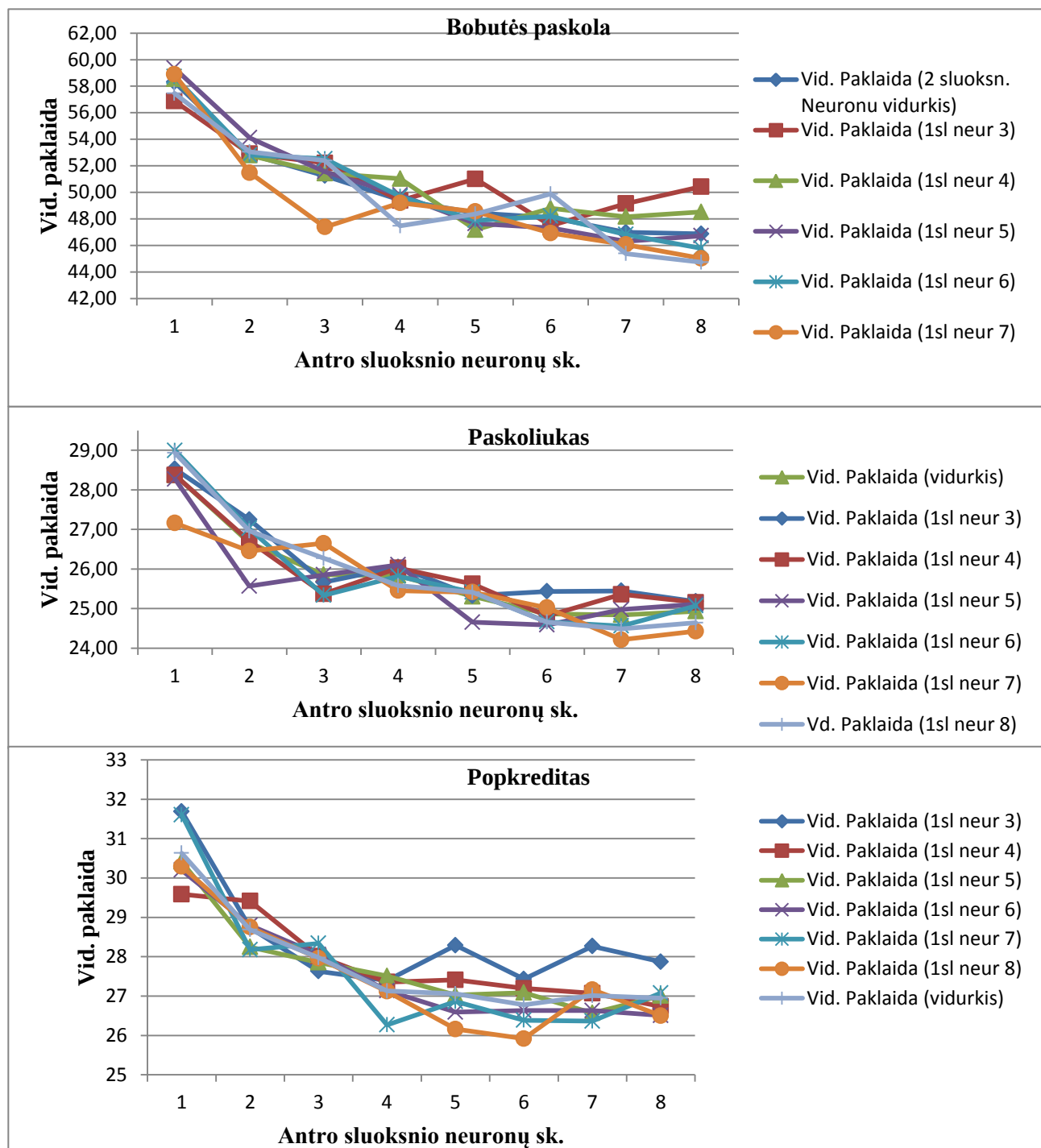
5.12 pav. Vidutinės paklaidos priklausomybė nuo 1 sluoksnio neuronų skaičiaus, taikant NARX modelį

Iš 5.12 paveiksle galime matyti, kad didėjant pirmo sluoksnio neuronų skaičiui, mažėja ir paklaida. Tik „Paskoliukas“ atveju paklaida ties aštuntuoju neuronu labai išauga. Bet tendencija yra akivaizdi, kad sumažinti paklaidą reikia didinti pirmo sluoksnio neuronų skaičių.

Mažiausia „Bobutės paskola“ bandymo vidutinė paklaida kai pirmo sluoksnio neuronų skaičius 8 yra 46,7, „Bobutės paskolos“ bandymo be sezoniškumo pikų – 20,7, „Paskoliuko“ – 24,6 (pirmo sluoksnio neuronų skaičius – 8), o „Popkredito“ – 26,3 (pirmo sluoksnio neuronų skaičius – 7).

5.4.2. NARX metodo efektyvumo priklausomybių nuo antro sluoksnio neuronų skaičiaus tyrimas

Tiriant NARX metodo efektyvumo priklausomybę nuo antro sluoksnio neuronų skaičiaus buvo matuojama vidutinė paklaida, skaičiuojama visų mokymų ir visų vėlinimo variacijų vidutinė paklaida tam tikram pirmo ir antro sluoksnio neuronų skaičiui. Neuronų skaičius buvo keičiamas nuo 1 iki 8. 5.13 paveiksle atvaizduota vidutinės paklaidos priklausomybė nuo neuronų skaičiaus.



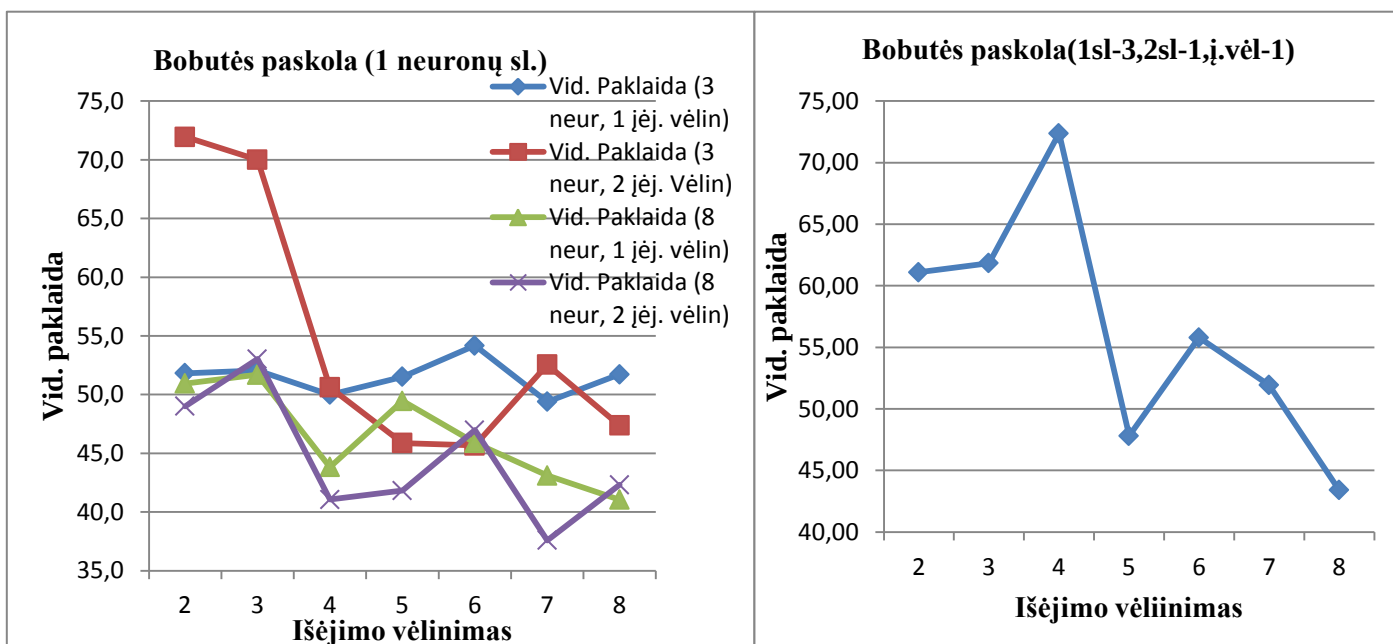
5.13 pav. Vidutinės paklaidos priklausomybė nuo antro sluoksnio neuronų skaičiaus, taikant NARX modelį

Pagal 5.13 paveikslą grafikus matome, kad vidutinės paklaidos priklausomybė nuo antro sluoksnio nėra tiesinė, bet labai akivaizdžiai vidutinė paklaida mažėja didinant antro sluoksnio neuronų skaičių visų projektų atvejais.

„Bobutės paskolos“ atveju mažiausia vidutinė paklaida 44,7 gaunama, kai pirmo sluoksnio neuronai – 8, antro – 8. „Paskoliuko“ atveju mažiausia vidutinė paklaida yra 24,2 kai pirmo ir antro sluoksnio neuronų skaičiai yra 7. „Popkredito“ atveju mažiausia vidutinė paklaida gaunama 25,9, kai pirmos sluoksnio neuronų yra 6, o antro – 8. Galima daryti prielaidą, kad dar padidinus neuronų skaičių turėtų mažėti ir paklaida, tačiau per didelis neuronų skaičius paklaidą gali ir padidinti ir per daug apkrauti programos veikimo procesą.

5.4.3. NARX metodo efektyvumo priklausomybių nuo išėjimo vėlinimo tyrimas

Mokymo metu, eksperimentuojama keičiant išėjimo vėlinimą, imant skirtingą skaičių išėjimų reikšmių. 5.14 paveiksle pateikti „Bobutės paskolos“ bandymo vidutinės paklaidos kitimo grafikai, kurie priklauso nuo išėjimo vėlinimo.

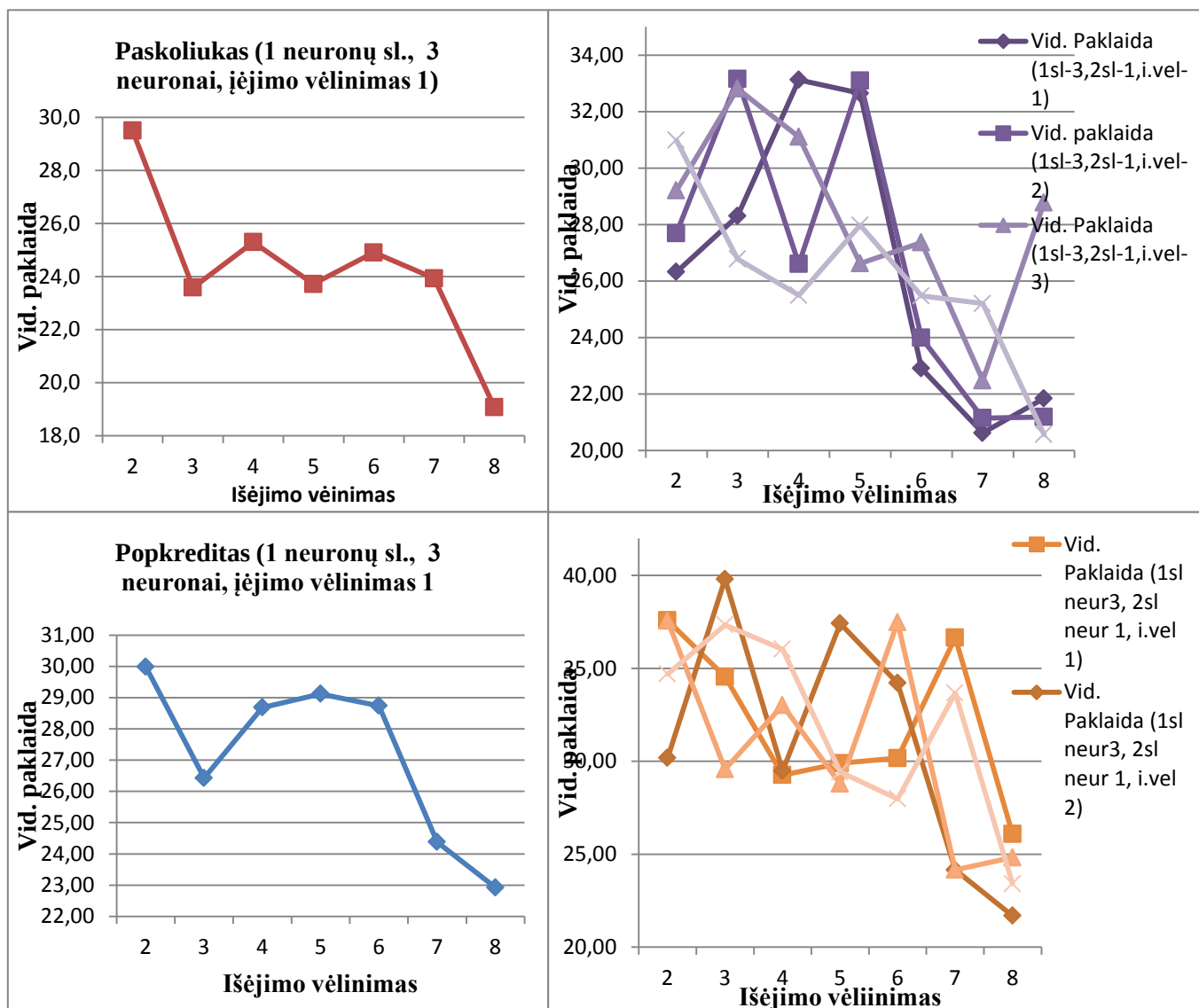


5.14 pav. „Bobutės paskolos“ bandymo vidutinės paklaidos priklausomybė nuo išėjimo vėlinimo taikant NARX modelį

Norint tiksliau įvertinti išėjimo vėlinimo įtaką vidutinei paklaidai, buvo paimti keli atvejai (žr. 5.14 pav.). Pirmame grafike kairėje pusėje pateiktos vidutinės paklaidos priklausomybės nuo išėjimo vėlinimo kreivės, kai naudojamas vieno sluoksnio neuronų tinklas, imami 3 ir 8 neuronai, įėjimo vėlinimas – atitinkamai 1 ir 2. Dešinėje pusėje pateiktas konkretus vidutinės paklaidos priklausomybės nuo išėjimo vėlinimo kreivės pavyzdys, kai taikomas 2 sluoksnių neuronų tinklas, pirmame sluoksnyje yra 3 neuronai, antrame – 1, o įėjimo vėlinimas – 1.

5.14 paveiksle vidutinė paklaida svyruoja priklausomai nuo išėjimo vėlinimo, bet matosi nedidelė tendencija, kad didinant išėjimo vėlinimą paklaidą šiek tiek vis mažėja.

5.15 paveiksle pateikti viršutiniai du grafikai yra „Paskoliuko“, apatiniai – „Popkredito“. Kairėje pusėje pateikti grafikai, kai apmokymui naudojamas tik vienas neuronų sluoksnis, dešinėje – naudojami du neuronų sluoksniai.

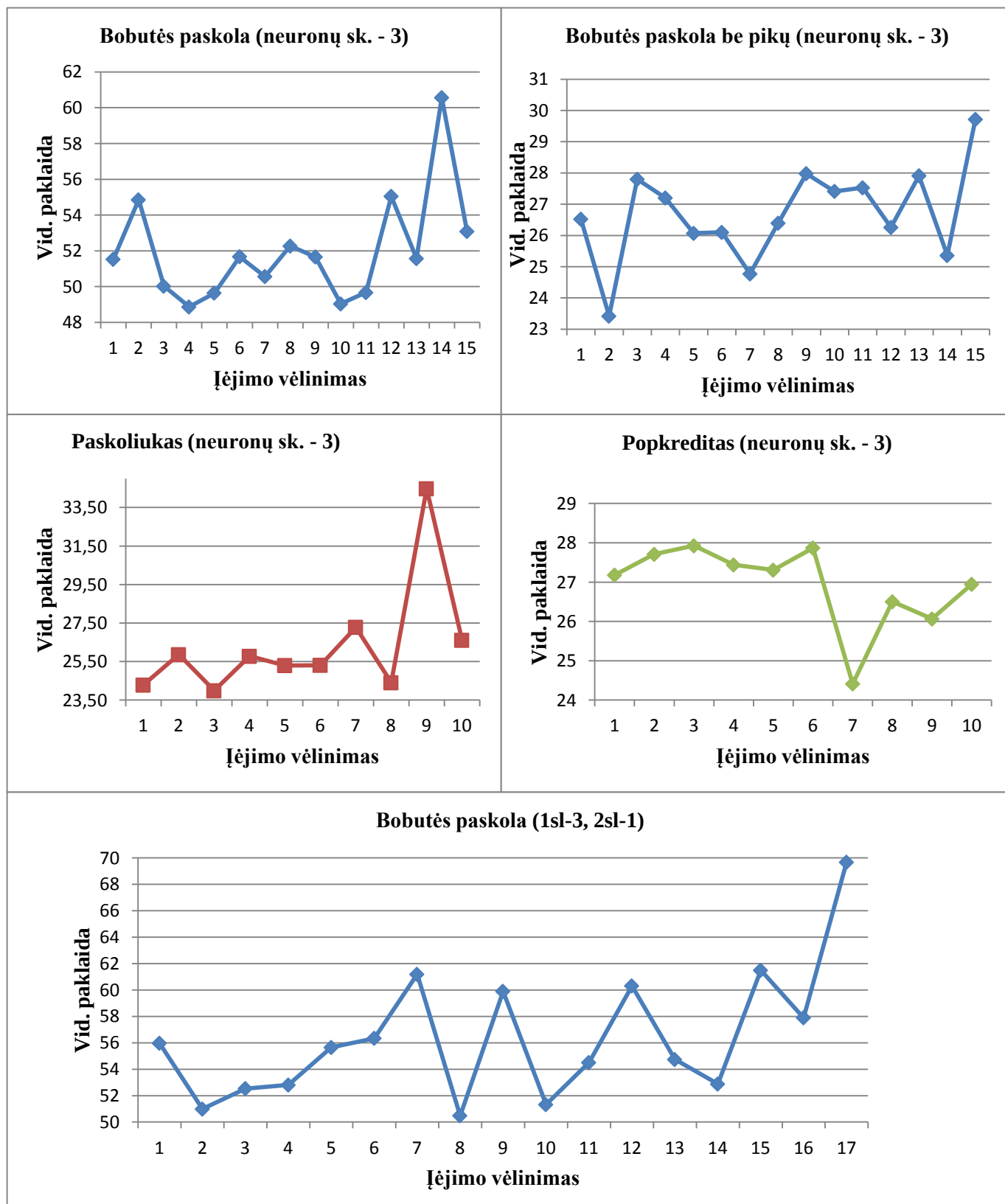


5.15 pav. Vidutinės paklaidos priklausomybė nuo išėjimo vėlinimo taikant NARX modelį

Kaip matoma 5.15 paveiksle, priklausomybė yra labai kintanti, bet tendencija – didėjant išėjimo vėlinimui, mažėja vidutinė paklaida. Mažiausios vidutinės paklaidos gaunamos, kai išėjimo vėlinimas yra 7 – 8 visais atvejais.

5.4.4. NARX metodo efektyvumo priklausomybių nuo įėjimo vėlinimo tyrimas

Eksperimento metu buvo tiriama vidutinės paklaidos priklausomybė nuo įėjimo vėlinimo. 5.16 paveiksle pirmi keturi pavyzdžiai – kai taikomas 1 neuronų sluoksnis, paskutiniame – 2 neuronų sluoksniai.



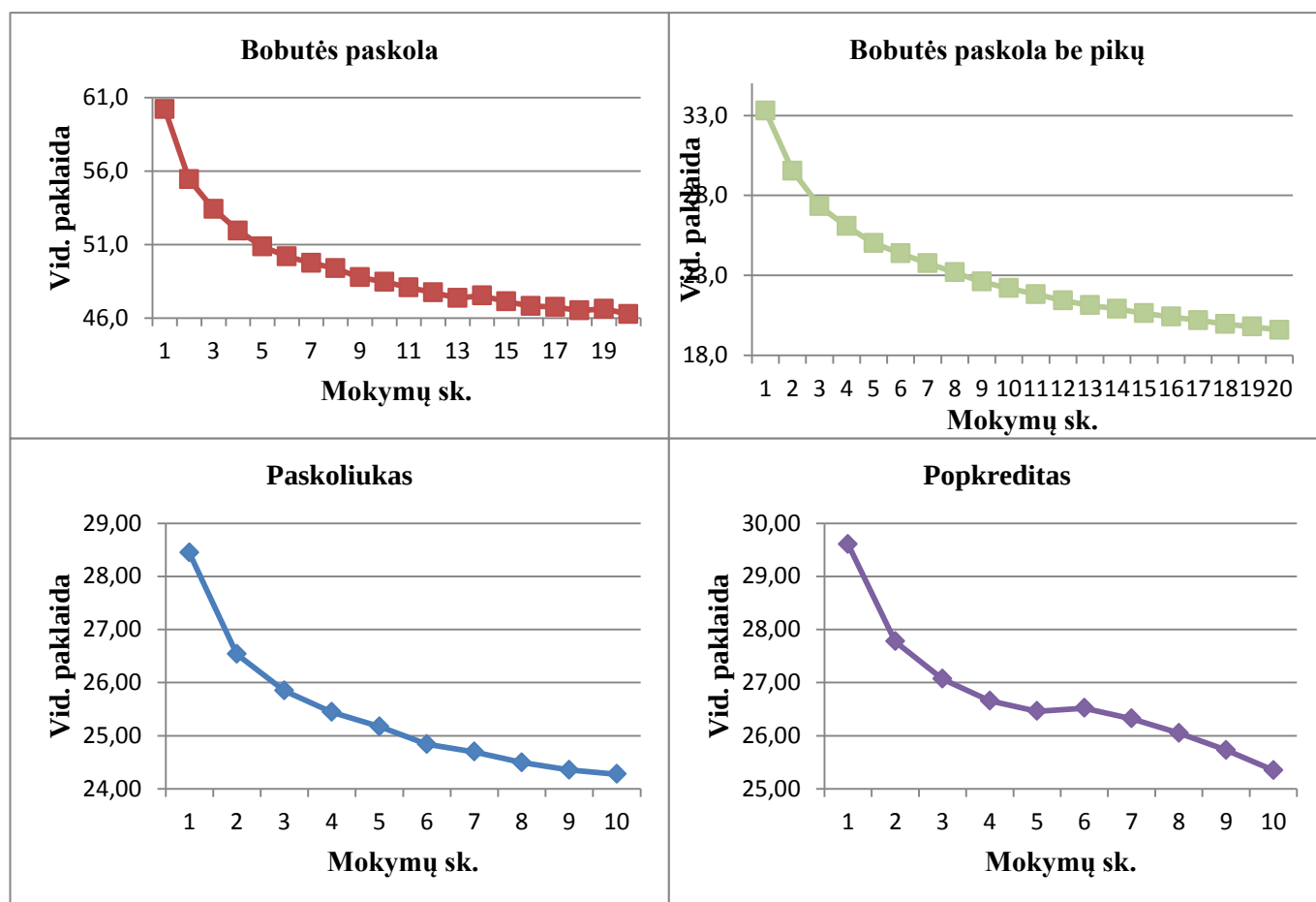
5.16 pav. Vidutinės paklaidos priklausomybė nuo įėjimo vėlinimo, taikant NARX modelį

Kai įėjimo vėlinimo vertė yra keičiama, vidutinė paklaida labai svyruoja. „Bobutės paskola“ atveju, kadangi įėjimo vėlinimo reikšmių imama daugiau, matome, kad prie didesnių vėlinimo reikšmių (14 – 17) paklaida didėja, nors ir labai svyruodama. Pagal gautus rezultatus sunku įvertinti, koki įėjimo vėlinimą reiktų imti, bet pagal 5.16 paveiksle galima padaryti prielaidą, kad

reiktų vengti didesnių įėjimo vėlinimo reikšmių, o parinkti tinkamą skaičių galima tik konkretaus eksperimento metu. „Bobutės paskola“ atveju mažiausia vidutinė paklaida gaunama kai įėjimo vėlinimas – 4 (48,9), „Bobutės paskola“ be pikų – 2 (23,4), „Paskoliukas“ – 3 (24) ir „Popkreditas“ – 7 (24,4), „Bobutės paskola“ kai naudojami du neuronų sluoksniai – 8 (50,48).

5.4.5. NARX metodo efektyvumo priklausomybių nuo mokymų skaičiaus tyrimas

Taikant 1 neuronų sluoksnį, buvo atliekama 10 – 20 mokymų, kiekvieną kartą vidutinė paklaida skirdavosi (žr. 5.17 pav.).

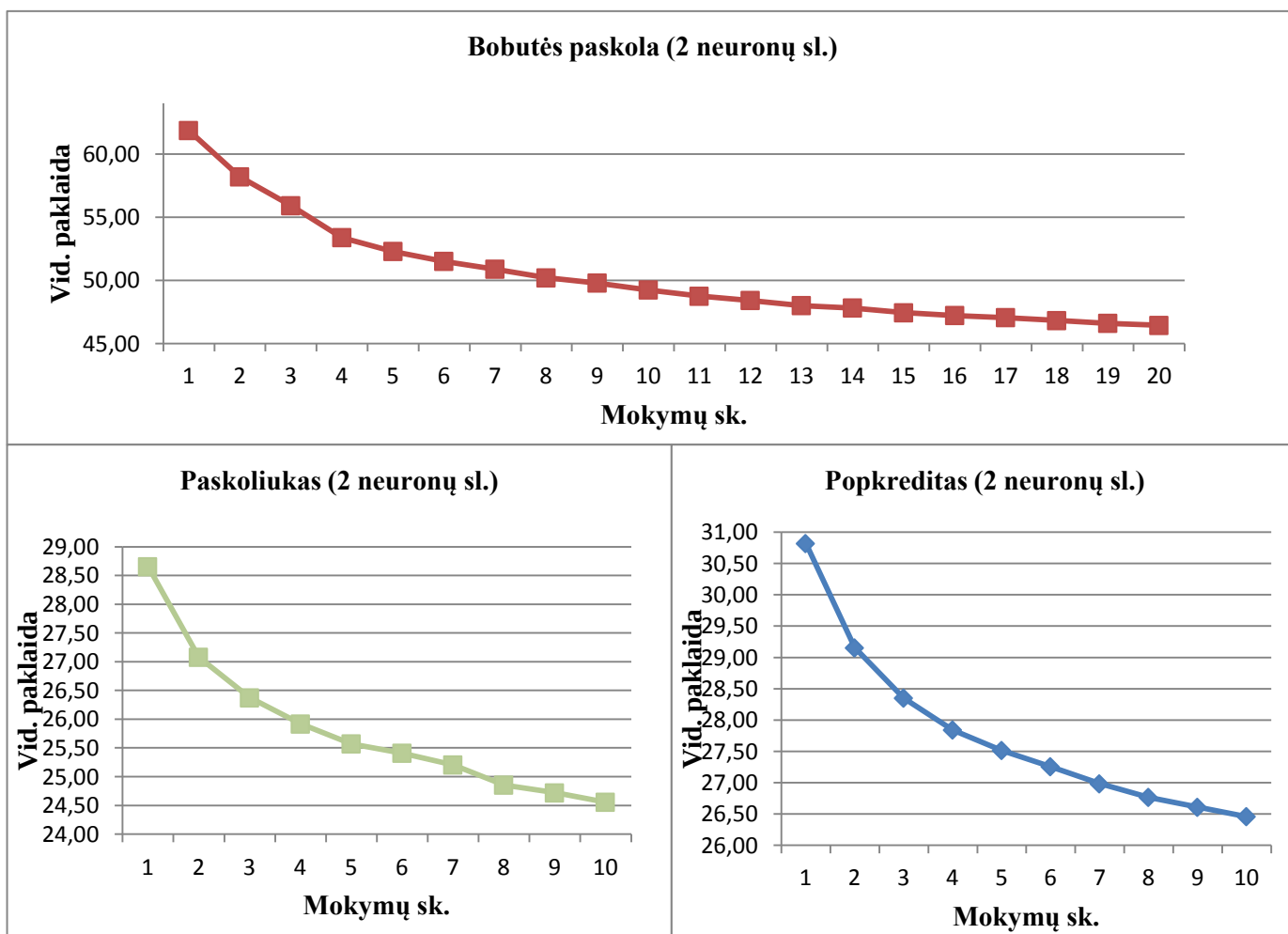


5.17 pav. Vidutinės paklaidos priklausomybė nuo mokymų skaičiaus, taikant 1 neuronų sl. ir NARX modelį

Taikant vieno sluoksnio neuronų tinklą visais atvejais vidutinės paklaidos priklausomybė nuo mokymų skaičiaus kito hiperboliškai, paklaida, didėjant mokymų skaičiui, mažėjo (žr. 5.17 pav.).

Galima daryti prielaidą, priklausomai nuo laiko ir kompiuterio resursų, norint gauti mažesnę paklaidą reikia didinti mokymų skaičių.

Taikant 2 sluoksnių neuronus, buvo atliekama 10 – 20 mokymų, kiekvieną kartą vidutinė paklaida taip pat skirdavosi (žr. 5.18 pav.).



5.18 pav. Vidutinės paklaidos priklausomybė nuo mokymų skaičiaus, taikant 2 neuronų sl. ir NARX modelį

Pagal 5.18 ir 5.17 paveikslus matoma tapati tendencija, kad didinant mokymų skaičių, vidutinė paklaida hiperboliškai mažėja. 5.2 lentelėje pateiktos mažiausios vidutinės paklaidos, kai atliekamas paskutinis bandymas (10 arba 20), taikant skirtingus projektus ir neuronų sluoksnius.

5.2 lentelė. Mažiausios vidutinės paklaidos, esant skirtingam neuronų sluoksnių skaičiui, taikant NARX modelį, atliekant paskutinį mokymą (10 arba 20)

Mažiausia vidutinė paklaida	1 neuronų sluoksnis	2 neuronų sluoksniai
„Bobutės paskola“	46,3	46,5
„Paskoliukas“	24,3	24,6
„Popkreditas“	25,4	26,5

Pagal 5.2 lentelę matome, kad naudojant 1 sluoksnių neuronų tinklą paklaida gaunama mažesnė, bet nežymiai. Naudojant mažiau sluoksnių sumažėja skaičiavimo procesas ir naudojama mažiau resursų.

5.5. Laiko vėlinimo neuronų tinklas (TDNN)

Taikant TDNN metodą kintamieji yra: pirmo sluoksnio neuronai, vėlinimas bei mokymų skaičius. Išėjime prognozuojamas paspaudimų skaičius, įėjimui naudojant du rodiklius: atmetimo rodiklį ir vidutinę vizito trukmę. Analizuojant „Bobutės paskolą“, pirmo sluoksnio neuronų skaičius buvo keičiamas nuo 1 iki 8, vėlinimas nuo 1 iki 18, mokymai buvo kartojami po 20 kartų. Visiems atvejams buvo skaičiuojama vidutinė paklaida, rezultatų gautą 2880. Dėl ilgo skaičiavimo proceso, resursų trūkumo, kitiems projektams kintamųjų skaičių intervalai buvo sumažinti: pirmo sluoksnio neuronų skaičius buvo keičiamas nuo 3 iki 8, vėlinimas – nuo 1 iki 10, mokymai kartojami 10 kartų, rezultatų gautą 600.

5.5 programos kodo 1 – 3 eilutėse yra nuskaitomi duomenys, y – bendras paspaudimų skaičius, x_1 – atmetimo rodiklis, x_2 – vidutinė vizito trukmė. x – paruoštas 2 įėjimų pavyzdžių matrica iš dviejų rodiklių, T – paruošiamas teisingų atsakymų vektorius, $kaupiu_perf$ – paruošiamas kintamasis saugoti tinklo vidutinės kvadratinės paklaidos reikšmes (4 – 6 eilutės).

Kaip ir ankstesnėse programose, kodo pradžia analogiška, vėliau yra kuriami du `for` ciklai, pirmasis – eksperimentuojama, kai pirmajame sluoksnyje esančių neuronų skaičius yra nuo 1 iki 8, antrasis – eksperimentuojama, kai vėlinimas yra keičiamas nuo 1 iki 18. Pavyzdžiui, kai vėlinimas yra 1, imamos dabartinės rodiklių reikšmės, kai vėlinimas yra 2, imamos ankščiau gautos reikšmės ir t.t. (žr. 5.5 programos kodą 8 – 9 eilutes). Su nustatytais parametrais yra sukuriamas `net` tinklas, kuriam priskiriamas validacijos skaičius 40. Taikant funkciją `preparets` paruošiami įėjimo duomenys – atrenkami įėjimo pavyzdžiai pagal tai, kiek jie turi būti suvėlinti išėjimo atžvilgiu (žr. 5.4 programos kode 10 – 12 eilutės). Mokymas vykdomas 20 kartų naudojant funkciją `train`, rezultatai pritaikomi sukurtam neuronų tinklui ir pabaigoje skaičiuojama bei kaupiama vidutinė paklaida (5.5 programos kodo 13 – 20 eilutės).

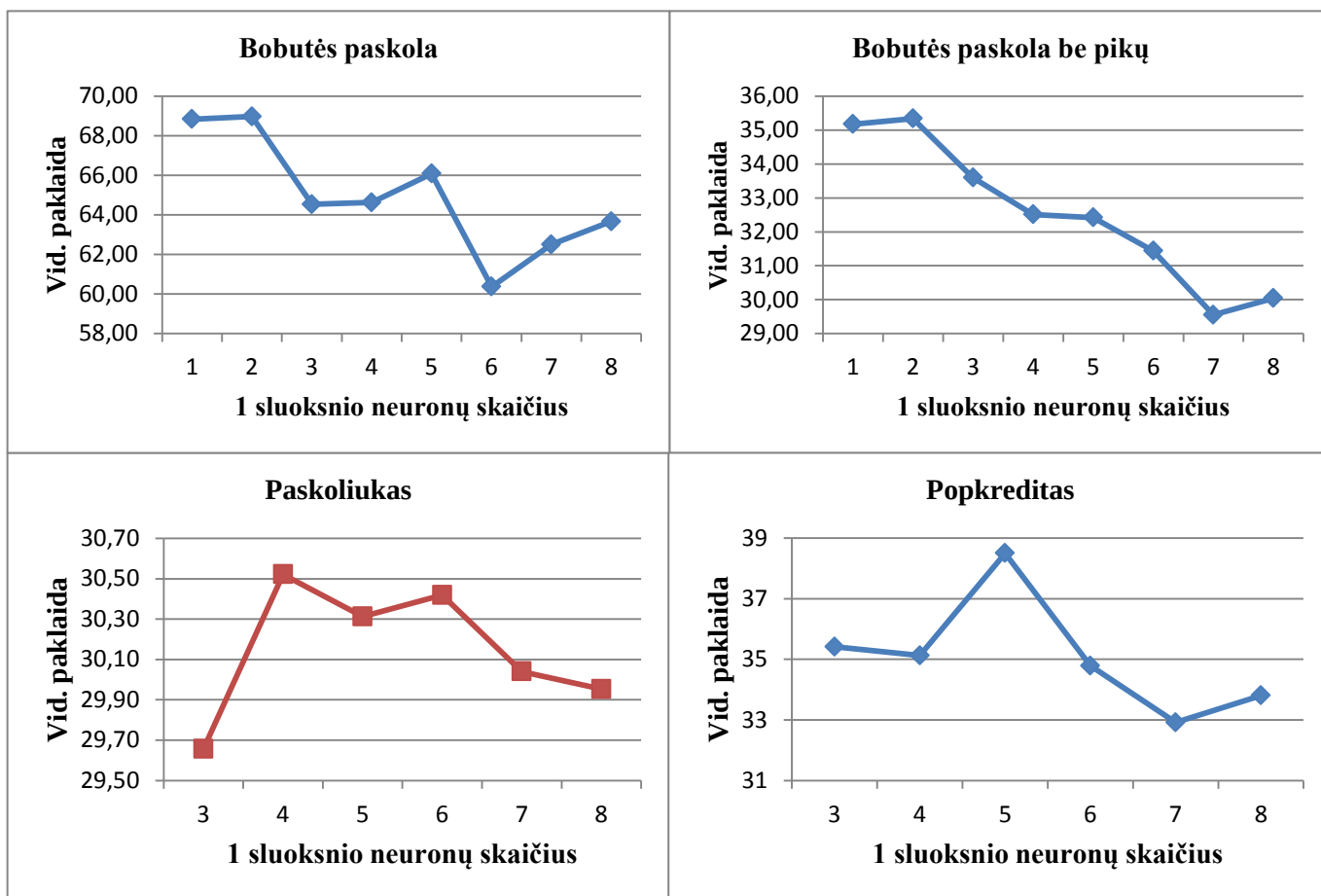
5.5 programos kodas

```
1. y=xlsread('BobutesPaskola.xls','Lapas1', 'V44:V130');
2. x1=xlsread('BobutesPaskola.xls','Lapas1', 'B44:B130');
3. x2=xlsread('BobutesPaskola.xls','Lapas1', 'C44:C130');
4. X=mat2cell([x1';x2'],2,ones(1,numel(x1)));
5. T=mat2cell(y',1,ones(1,numel(y)));
6. kaupiu_Y=[];
7. kaupiu_perf=[];
8. for sluoksnio_neuronai=1:8
9.     for velinimas=1:18
10.        net = timedelaynet(1:velinimas,sluoksnio_neuronai);
11.        net.trainParam.max_fail=40;
12.        [Xs,Xi,Ai,Ts] = preparets(net,X,T);
13.        for mokymas=1:20
14.            net = train(net,Xs,Ts,Xi,Ai);% tinklo mokymas
15.            Y = net(Xs,Xi,Ai);% paleidžiamas apmokytas tinklas paduodant ()
16.            perf = perform(net,Ts,Y); % vidutinė klaida?
17.            kaupiu_perf=[kaupiu_perf,perf];
18.        end
```

19. end
20. end

5.5.1. TDNN metodo efektyvumo priklausomybių nuo pirmo sluoksnio neuronų skaičiaus tyrimas

Tiriant TDNN metodo efektyvumo priklausomybę nuo pirmo sluoksnio neuronų skaičiaus, buvo matuojama vidutinė paklaida, skaičiuojama visų mokymų ir visų vėlinimo variacijų vidutinė paklaida tam tikram neuronų skaičiui. „Bobutės paskola“ projektui neuronų skaičius buvo keičiamas nuo 1 iki 8, kitų projektų sumažinta nuo 3 iki 6. 5.19 paveiksle atvaizduota vidutinės paklaidos priklausomybė nuo neuronų skaičiaus.



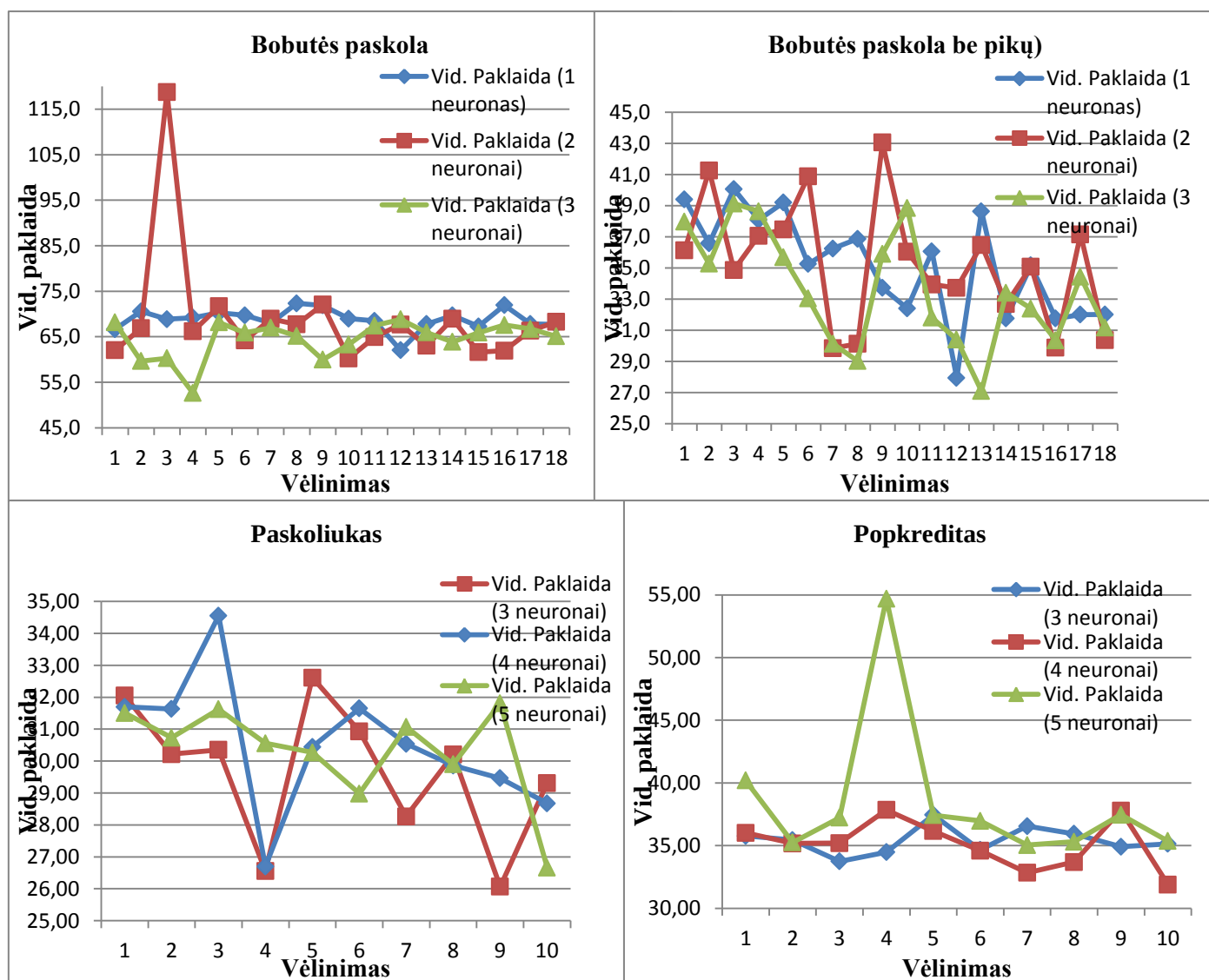
5.19 pav. Vidutinės paklaidos priklausomybė nuo neuronų skaičiaus, taikant TDNN modelį

Pagal 5.19 paveiksle pateiktus grafikus galima spręsti, kad didėjant neuronų skaičiui, paklaida mažėja. Mažėjimas matomas pagal bendrą tendenciją, bet taip pat labai akivaizdus paklaidos svyravimas. Paklaidos mažėjimas labiau matomas „Bobutės paskola“ atveju. Analizuojant „Paskoliukas“ vidutinės paklaidos kitimo kreivę, vidutinės paklaidos mažėjimo tendencija yra matoma, bet kai neuronų yra 3, vidutinė paklaida yra mažiausia, o kai neuronų skaičius 4, įvyksta didžiulis šuolis ir paklaida labai išauga. Panašiai yra ir su „Popkreditu“: vidutinės paklaidos mažėjimo tendencija aiški, bet kai pirmame sluoksnyje neuronų yra 5, paklaida labai išauga.

Vidutiniškai visų projektų paklaida mažiausia kai neuronų skaičius yra 7. Norint kiekvienam atvejui tiksliai įvertinti kokį neuronų skaičių parinkti reikia atlikti individualias analizes.

5.5.2. TDNN metodo efektyvumo priklausomybių nuo vėlinimo tyrimas

Atliekant TDNN metodo tyrimą, buvo analizuojama vidutinės paklaidos priklausomybė nuo vėlinimo, esant skirtingam neuronų skaičiui. „Bobutės paskolos“ ir „Bobutės paskolos“ be pikų, „Paskoliuko, bei „Popkredito“ priklausomybių kreivės pateiktos 5.20 paveiksle.

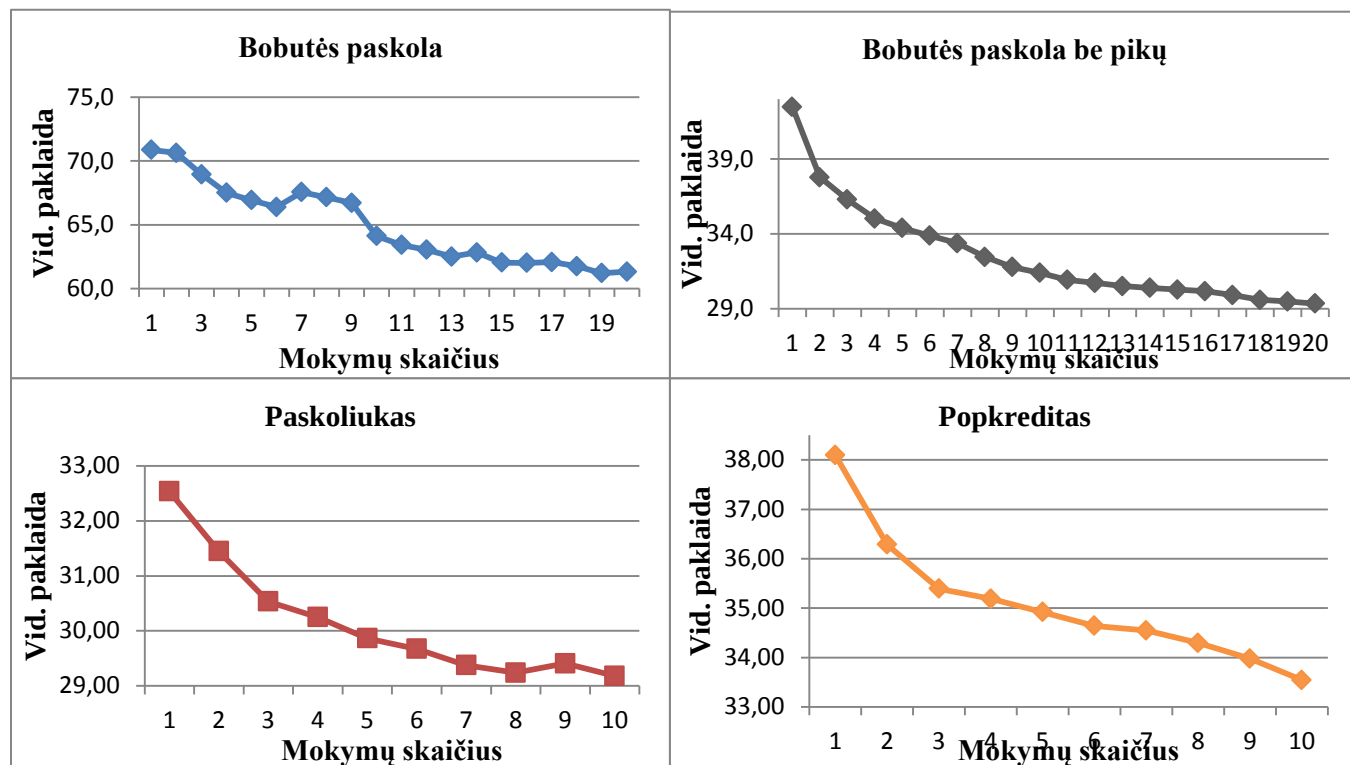


5.20 pav. Vidutinės paklaidos priklausomybė nuo vėlinimo, taikant TDNN modelį

Didinant vėlinimo reikšmę, vidutinė paklaida labai svyruoja. Pagal bendrą kitimo tendenciją galima matyti, kad didėjant vėlinimui vidutinė paklaida minimaliai mažėja. Kadangi svyravimas didelis, sunku įvertinti, kokios vėlinimo reikšmės apskritai geriausios, kiekvienam projektui, duomenims reikia atlikti individualias analizes ir interpretacijas.

5.5.3. TDNN metodo efektyvumo priklausomybių nuo mokymų skaičiaus tyrimas

Taikant 1 neuronų sluoksnį, buvo atliekama 10 – 20 mokymų, kiekvieną kartą vidutinė paklaida skirdavosi (žr. 5.17 pav.).



5.21 pav. Vidutinės paklaidos priklausomybė nuo mokymų skaičiaus, taikant TDNN modelį

Kuo daugiau kartų yra vykdomas apmokymas, tuo paklaida yra mažesnė. Vidutinės paklaidos kitimas didėjant mokymų skaičiui primena hiperbolę (žr. 5.21 pav.).

„Bobutės paskola“ mažiausia vidutinė paklaida gaunama, kai vykdomas 19 mokymas, vidutinė paklaida – 61,3. Kai analizuojama „Bobutės paskolos“ dalis be pikų, mažiausia vidutinė paklaida gaunama apmokant dvidešimtą kartą – 29,3. „Paskoliuko“ mažiausia vidutinė paklaida gaunama apmokant paskutinįjį – dešimtą kartą – 29,2, o „Popkreditą“ – taip pat apmokant dešimtą kartą – 33,5.

5.3 lentelė. Mažiausios vidutinės paklaidos, taikant TDNN modelį, geriausio apmokymo metu

Projektas	Vidutinė paklaida	Vidutine paklaida, %
„Bobutės paskola“	61,2	30,1
„Bobutės paskola“ be pikų	29,3	16,2
„Paskoliukas“	29,2	24,5
„Popkreditas“	35,5	29,3

Geriausias rezultatas gaunamas „Bobutės paskola“ be pikų atveju. Esminė priežastis – šiuo konkrečiu atveju buvo sumažinta duomenų apimtis ir išvengta didelių reikšmių kitimo, taip pat

apmokymas buvo vykdomas 20 kartų, o kuo daugiau apmokymų yra vykdoma tuo paklaida kaip rodo 5.21 paveiksle – mažėja.

5.6. Skyriaus apibendrinimas

Apibendrinant prognozavimo metodų efektyvumo tyrimą, 5.4 lentelėje pateikti metodai ir parinkti parametrai, su kuriais gautos mažiausios vidutinės paklaidos.

5.4 lentelė. Prognozės metodų parametrai, su kuriais gauta mažiausia vidutinė paklaida

Metodas	Kintamieji	Projektai			
		Bobutės paskola	Bobutės paskola be pikų	Paskoliukas	Popkreditas
LPC	Lango dydis	37	76	110	48
	Filtro eilė	8	8	8	8
ARMA	Lango dydis	170	x	76-80	84-92
	Modelio eilė	[1 1]	x	[1 1]	[1 1]
NAR	Neuronų sk.	1	x	1	1
	Išėjimų vėlinimas	6	x	6	13
	Mokymų sk.	13 iki 20	x	6 iki 10	6 iki 10
NARX 1 neuronų sl.	Neuronų sk.	7	8	8	6
	Iėjimų vėlinimas	8	14	8	5
	Išėjimų vėlinimas	2	6	8	8
	Mokymų sk.	10 iki 20	13 iki 20	10	4 iki 10
NARX 2 neuronų sl.	1 sl. Neuronų sk	6	x	4	7
	2 sl. Neuronų sk	5	x	7	7
	Iėjimų vėlinimas	17	x	4	6
	Išėjimų vėlinimas	2	x	8	8
	Mokymų sk.	16 iki 20	x	4 iki 10	5 iki 10
TDNN	Neuronų sk.	6	7	8	7
	Vėlinimas	17	17	7	6
	Mokymų sk.	13 iki 19	11 iki 20	6 iki 10	9 iki 10

Pagal 5.4 lentelę matoma, kad taikant LPC metodą mažiausia paklaida gauta lango dydį parenkant kiekvienam projektui. Mažiausia paklaida gauta kai filtro eilė – 8.

Taikant ARMA modelį geriausia modelio eilė – [1 1], „Bobutės paskola“ atveju lango dydis maksimalus, kitų projektų atveju lango dydis apie 76 – 92, su kuriais vidutinė paklaida mažiausia. „Bobutės paskola“ atveju mažiausia paklaida gauta, kai apmokymui panaudotas maksimalus duomenų skaičius dėl to, kad duomenų svyravimas kur kas didesnis bei didesni duomenų šuoliai.

NAR metodo atveju, prognozuojant „Bobutės paskola“ duomenis, tyrimas buvo atliekamas neuronų skaičių keičiant nuo 1 iki 8, išėjimo vėlinimą – nuo 6 iki 18, mokymų atlikta 20, kitų projektų atvejais neuronų skaičius keičiamas nuo 1 iki 6, išėjimo vėlinimas – nuo 6 iki 14, mokymų atlikta 10. Visais atvejais geriausias neuronų skaičius yra minimalus – 1, išėjimų vėlinimas taip pat minimalus – 6, išskyrus „Popkreditas“ – 13. Kuo mokymų skaičius didesnis, tuo paklaida mažėja.

Taikant NARX metodą, kai naudojamas vienas neuronų sluoksnis, neuronų skaičius buvo keičiamas nuo 3 iki 8, išėjimo vėlinimas – nuo 2 iki 8, įėjimų vėlinimas – nuo 1 iki 17 ir nuo 1 iki 10, mokymai 20, atitinkamai kitiems projektams sumažintas iki 10. Geriausias neuronų skaičius svyruoja nuo 6 iki 8. Kai įėjimo vėlinimas 8, mažiausia paklaida gauta analizuojant „Bobutės paskola“ ir „Paskoliukas“ atveju, „Bobutės paskola“ be pikų – 14, o „Popkreditas“ – 5. Išėjimo vėlinimas svyruoja nuo 2 iki 8. Naudojant du neuronų sluoksnius, pirmo sluoksnio neuronų skaičius su kuriais paklaida mažiausia kinta tarp projektų nuo 4 iki 7, antro sluoksnio – nuo 5 iki 7.

Atliekant tyrimą su TDNN metodu mažiausia paklaida gaunama kai neuronų skaičius 6 – 8, vėlinimas – 17, 6 ir atitinkamai 5.

Visais atvejais didėjant mokymų skaičiui, vidutinė paklaida mažėja. Dėl duomenų svyravimo panašumo, parametrai, su kuriais paklaida mažiausia, panašiausi „Paskoliuko“ ir „Popkredito“. „Bobutės paskolos“ rezultatai skiriasi, nes kintamųjų parametru intervalai yra didesni, duomenų svyravimas taip pat didesnis. Analizuojant „Bobutės paskola“ be pikų duomenis, parametru dydžiai skirtingi, nes nagrinėjamas duomenų intervalas kur kas mažesnis nei kitų, bet bendra tendencija matoma visais atvejais.

5.5 lentelėje pateiktos mažiausios paklaidos ir jų procentinės išraiškos kiekvieno projekto ir metodo atveju. Žalia spalva pažymėtuose langeliuose – geriausi konkretaus projekto rezultatai, šviesia žalia – mažiau geri rezultatai. Prastesni rezultatai pažymėti tamsesnėmis raudonos spalvos atspalviais.

5.5 lentelė. Mažiausios paklaidos gautos kiekvieno prognozavimo metodo ir projekto atveju

	Projektai			
	Bobutės paskola	Bobutės paskola be pikų	Paskoliukas	Popkreditas
Metodas	Vidutinė paklaida/ vidutinė paklaida, %			
LPC	51,6	28,1	25,2	31
	25,5	15,5	21,3	25,7
ARMA	24,7		13,7	14,7
	12,2		11,5	12,1
NAR	49,4		28,1	30,4
	24,3		23,6	25,1
NARX 1 neuronų sl.	25,1	3	13,8	15,5
	12,4	1,7	11,6	12,8
NARX 2 neuronų sl.	18,1		12	13,4
	8,9		10,1	11,1
TDNN	41,3	8,1	21,8	23,7
	20,3	4,5	18,3	19,6

Pagal 5.5 paveikslą. matome, kad mažiausia paklaida gauta analizuojant „Bobutės paskola“ be pikų, pritaikius NARX metodą su 1 neuronų sluoksniu, mažiausia paklaida – 3, tai yra 1,7 % nuo vidutinio paspaudimo skaičiaus analizuojamame intervale. Taikant TDNN metodą, mažiausia paklaida – 4,5 % ir didesnė taikant LPC – 15,5 %.

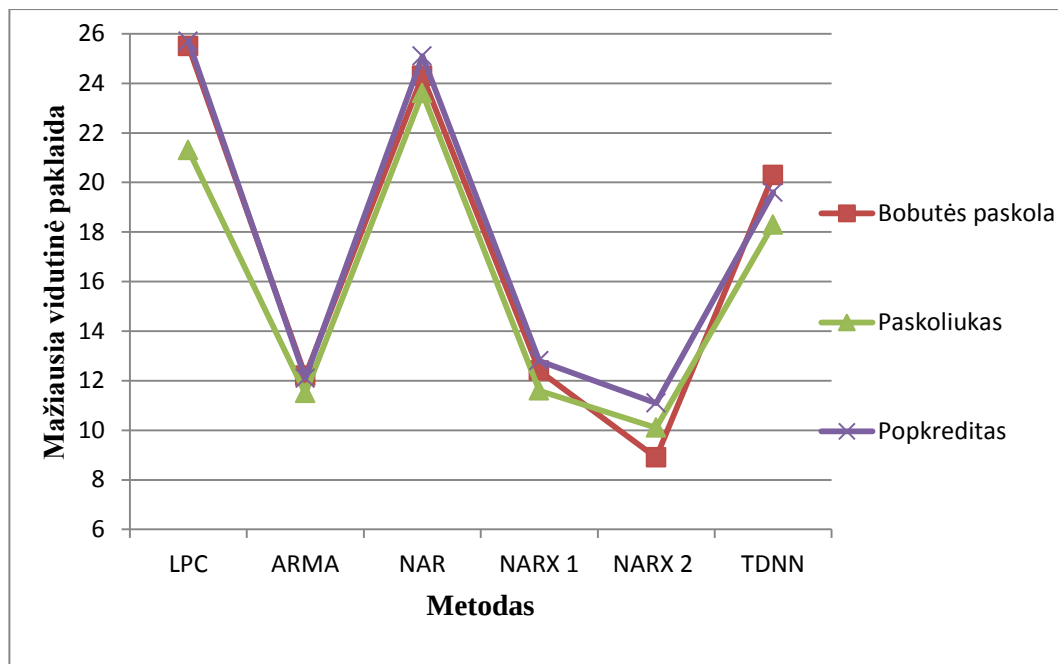
Taikant NAR metodą, „Bobutės paskolos“ mažiausia paklaida 24,3 %, TDNN metodą – 20,3 %, NARX metodą su vienu neuronų sluoksniu – 12,4 %, ARMA – 12,2 %, NARX su dvejais neuronų sluoksniais – 8,9 % ir LPC metodą – 25,5 %.

Analizuojant „Paskoliuko“ duomenis, metodai su mažiausiomis paklaidomis: NAR – 23,6 %, TDNN – 18,3 %, NARX su vienu neuronų sluoksniu – 11,6 %, ARMA – 11,5 %, NARX su dvejais neuronų sluoksniais – 10,1 %, LPC – 21,3 %.

Atliekant analizę „Popkredito“ duomenų, metodai su mažiausiomis paklaidomis: NAR – 25,1 %, TDNN – 19,6 %, NARX su vienu neuronų sluoksniu – 12,8 %, ARMA – 12,1 %, NARX su dvejais neuronų sluoksniais – 11,1 %, LPC – 25,7 %.

Bandymo metu buvo padaryta prielaida kad skirtingose signalo prognozės vietose klaida kinta nevienodai, todėl atliktas papildomas palyginimas prognozuojant tik paskutiniąsias 30 signalo verčių, esant skirtingiems lango pločiams nuo 18 iki 150, kad 4 reikšmes, o filtro eilė buvo keičiama nuo 1 iki 16. „Bobutės paskola“ atveju mažiausia paklaida gauta 49,7 (24,5 %), kai filtro eilė 10, o lango plotis – 146. „Paskoliukas“ – 28,3 (23,9 %), kai filtro eilė yra 11, o lango plotis – 122. „Popkreditas“ – 38,5(31,9 %), kai filtro eilė yra 10, o lango plotis – 146.

5.22 paveiksle pateikta mažiausios vidutinės paklaidos priklausomybė nuo taikomo metodo ir nuo analizuojamo projekto duomenų.



5.22 pav. Mažiausios klaidos priklausomybė nuo metodo ir nuo projekto

Analizuojant „Bobutės paskola“, „Paskoliukas“ ir „Popkreditas“ matoma bendra tendencija, paklaidos mažėjimo taikant tam tikrus metodus. Didžiausia paklaida gauta taikant NAR metodą ir LPC, toliau pagal eiliškumą: TDNN, ARMA, NARX su 1 neuronų sluoksniu ir geriausiai prognozuojantis – NARX su 2 neuronų sluoksniais.

6. APIBENDRINIMAS. IŠVADOS

Šio darbo tikslas buvo ištirti atvirojo kodo sistemose įgyvendintus optimizavimo metodus ir įvertinti jų tinkamumą svetainėms optimizuoti. Siekiant darbo tikslo, pirmiausiai buvo atlikta neautomatinių svetainių optimizavimo analitinė apžvalga. Pagal apžvalgos rezultatus, interneto svetainių optimizacija yra skirstoma į pagrindines dvi kategorijas: išorinę ir vidinę. Išorinės optimizacijos esminis rodiklis – išorinių nuorodų skaičius, vidinės: paspaudimų skaičius, atmetimo rodiklis, metaduomenys, svetainės puslapio atvertimo laikas, konversijų skaičius ir pan. Pagrindiniai svetainės vertinimo aspektai: žinomumas, funkcionalumas, praktiškumas ir turinys. Duomenys gali būti renkami naudojant atvirojo kodo įrankius: „Google Analytics“, „Open Web Analytics“, „Grape Web Statistics“, „Piwik“, kurie tarpusavyje skiriasi patogumu, naudotojo sąsaja, duomenų naujumu, išnaudojimo galimybėmis.

Atlikta prognozavimo metodų analizė, trumpai apžvelgta prognozavimo metodų raida. Tiek senais laikais, tiek dabar prognozavimas išlieka svarbi užduotis, kuri yra sprendžiama bei tobulinama. Prognozavimas laiko srityje yra plačiausiai taikomas įvairiuose mokslo ir gyvenimo srityse. Apžvelgti klasikiniai, žinomi ir populiarūs metodai: tiesinės prognozės metodas (LPC), slenkamųjų vidurkių metodas (ARMA), netiesinis autoregresinis neuronų tinklas (NAR), netiesinis autoregresinis su papildomų įėjimu neuronų tinklas (NARX) ir laiko vėlinimo neuronų tinklas (TDNN). Šie metodai plačiai taikomi prognozuoti duomenis kintančius laike: saulės radiacijos, elektros sąnaudų, infliacijos rodiklių, paspaudimų skaičių į svetainę ir kt. kitimai. Pagrindiniai metodų skirtumai matematinė išraiška, parametrai, struktūra, veikimo principas.

Prieš atliekant eksperimentus, taikant išvardintus prognozavimo metodus, pirmiausiai buvo analizuojami ir renkami duomenys, su kuriais buvo atliekami tyrimai. Atrinktos interneto svetainės, esančios toje pačioje labai populiarioje ir konkurencingoje paskolų rinkoje: „Bobutės paskola“, „Paskoliukas“ ir „Popkreditas“. Atrinkti priežastiniai duomenys (nuo kurių netiesiogiai priklauso pasekminiai rodikliai): atmetimo rodiklis, vidutinė vizito trukmė, išorinių nuorodų skaičius, socialinių profilių aktyvumas ir kiti, pasekminiai duomenys (galutiniai duomenys, kurie tiesiogiai teikia vertę svetainės savininkams): raktinių žodžių pozicijos Google natūralioje paieškoje, srautas į svetainę, konversijų skaičius. Duomenys renkami pasitelkus įvairiais įrankiais, pavyzdžiui: „moz“ įskiepis, „Google Analytics“, AWR, socialiniai profiliai, „ahrefs.com“.

Su atrinktais duomenimis (paspaudimo skaičius, vidutinė vizito trukmė, atmetimo rodiklis) iš skirtingų svetainių, taikant atrinktus klasikinius prognozės metodus, buvo prognozuojamos duomenų vertės. Atlikta prognozė kelių tipų, pirmiausiai prognozuojamas paspaudimo skaičius pagal to paties kintamojo – paspaudimo skaičiaus, istorinius duomenys (LPC, ARMA, NAR) ir

prognozuojamas paspaudimų skaičius, kai metodai apmokomi pagal du skirtingus įėjimus: atmetimo rodiklį ir vizito trukmės (NARX, TDNN).

Atliktus prognozavimo metodų efektyvumo tyrimą, vertinama mažiausia paklaida tarp realios ir prognozuojamos vertės. Atitinkami metodų parametrai buvo keičiami ir vertinami, su kuriomis parametų kombinacijos gaunama mažiausia paklaida, atliekant bandymus su visais paminėtais metodais ir duomenimis iš skirtingų svetainių. Šio tyrimo metu, tiksliausius prognozės rezultatus pavyko gauti taikant NARX metodą (naudojant 2 perceptronų sluoksnius) ir kiek prastesnius: NARX (naudojant 1 perceptrono sluoksnį), ARMA, TDNN, ir prasčiausiai veikė – NAR bei LPC metodai. Tiksliausiai prognozuojami duomenys buvo „Bobutės paskola“ su kur kas mažesniu duomenų intervalu, kai išvengiama didesnių duomenų signalo šuolių, mažiausia vidutinė paklaida gauta 1,7 % taikant NARX metodą su vienu perceptronų sluoksniu. Prognozuojant visus duomenys su visomis svetainėmis, geriausiai veikiančių metodų eiliškumas yra tas pats, kaip jau minėtas. Pateikiamos mažiausios paklaidos, konkretaus prognozavimo metodo ir svetainės duomenų atveju. „Bobutės Paskola“: NARX 2 perceptronų sluoksniai (8,9 %), NARX 1 perceptronų sluoksnis (12,4 %), ARMA (12,2 %), TDNN (20,3 %), NAR (24,3 %), LPC (25,5 %), „Paskoliukas“: NARX 2 perceptronų sluoksniai (10,1 %), NARX 1 perceptronų sluoksnis (11,6 %), ARMA (11,5 %), TDNN (18,3 %), LPC (21,3 %), NAR (23,6 %). „Popkreditas“: NARX 2 perceptronų sluoksniai (11,1 %), NARX 1 perceptronų sluoksnis (12,8 %), ARMA (12,1 %), TDNN (19,6 %), NAR (25,1 %), LPC (25,7 %).

Buvo padaryta prielaida, kad skirtingose signalo prognozės vietose klaida kinta nevienodai, dėl tos priežasties buvo atliktas papildomas tyrimas, prognozuojant tik paskutiniąsias 30 signalo verčių, keičiant lango plotį nuo 18 iki 150, o filtro eilę nuo 1 iki 16 imtinai. „Bobutės paskola“ atveju mažiausia paklaida gauta 49,7 (24,5 %), kai filtro eilė 10, o lango plotis – 146. „Paskoliukas“ – 28,3 (23,9 %), kai filtro eilė yra 11, o lango plotis – 122. „Popkreditas“ – 38,5 (31,9 %), kai filtro eilė yra 10, o lango plotis – 146.

Prognozavimo metodo efektyvumas priklauso nuo duomenų tipo, stabilumo, kiekio, sezoniškumo, duomenų įėjimo/išėjimo ir panašiai. Šio tyrimo metu, kai duomenys kinta laike apytiksliai intervale nuo 38 iki 610, kuriems įtaką daro sezoniškumas (savaitės dienos, mėnesio dienos), parinkus tinkamus parametrus geriausiai veikiantis metodas yra NARX (2 perceptronų sluoksniai), kurių atveju vidutinę paklaidą galima gauti vidutiniškai apie 10 %.

Literatūra

1. *E-verslas ir optimizavimas* [interaktyvus]. Įmonės Goodone tinklapis [žiūrėta 2014 m. gegužės 28 d.]. Prieiga per internetą: <<http://goodone.lt/e-verslas>>
2. *Web tracking* [interaktyvus]. 2011. Niklas Schmucker, Berlin University of Technology [žiūrėta 2014 m. gegužės 28 d.]. Prieiga per internetą: <http://www.snet.tu-berlin.de/fileadmin/fg220/courses/SS11/snet-project/web-tracking_schmuecker.pdf>
3. KRUG, S. 2006. *Don't make me think – a common sense approach to web usability*. California, USA: New Riders Publishing.
4. 3 Open Source Web Analytics Tools If you're Not A Google Analytics Fan [interaktyvus]. 2011. Danny Stieben [žiūrėta 2014 m. gegužės 28 d.]. Prieiga per internetą: <<http://www.makeuseof.com/tag/3-open-source-web-analytics-tools-google-analytics-fan/>>
5. *Web analytics Pakistan* [interaktyvus]. Įmonė „Web Solution provider“ [žiūrėta 2014 m. gegužės 28 d.]. Prieiga per internetą: <<http://www.websolpro.com/services/marketing/internet-marketing/web-analytics-pakistan.html>>
6. *How to quantify the use experience* [interaktyvus]. 2004. Robert Rubinoff [žiūrėta 2014 m. gegužės 28 d.]. Prieiga per internetą: <<http://www.sitepoint.com/quantify-user-experience/>>
7. *Data Driven Design Using Web Analytics to Improve Information Architectures* [interaktyvus]. 2007. Andrea Wiggins [žiūrėta 2014 m. gegužės 28 d.]. Prieiga per internetą: <<http://www.andreawiggins.com/research/AndreaWigginsIASummit2007.pdf>>
8. *Google Analytics for measuring website performances* [interaktyvus]. 2010. Beatriz Plaza [žiūrėta 2014 m. gegužės 28 d.]. Prieiga per internetą: <<http://www.sciencedirect.com/science/article/pii/S0261517710000622>>
9. RABUNAL, J.; DORADO, J. 2006. *Artificial Neural Networks in Real- Life Applications*. University of Coruna, Spain: Idea Group publishing.
10. *Neural Networks for Time Series prediction* [interaktyvus]. 2006. Dave Touretzky and Kornel Laskowski [žiūrėta 2014 m. gegužės 28 d.]. Prieiga per internetą: <<http://www.cs.cmu.edu/afs/cs/academic/class/15782-f06/slides/timeseries.pdf>>
11. NAVAKAUSKAS, D.; PAULIKAS, Š. 2004. *Skaitmeninis signalų apdorojimas programa MATLABTM*: mokomoji knyga. Vilnius: Technika. 7 p.
12. *ARIMA Models and the Box-Jenkins methodology* [interaktyvus]. Makridakis, S, Hibon. M. [žiūrėta 2014 m. gegužės 28 d.]. Prieiga per internetą: <<http://www.insead.edu/facultyresearch/research/doc.cfm?did=46900>>
13. *Quantization of LPC Parameters* [interaktyvus]. K. K. Paliwal, W. B. Kleijn [žiūrėta 2014 m. gegužės 28 d.]. Prieiga per internetą: <https://maxwell.ict.griffith.edu.au/spl/publications/papers/book_sc_kkp.pdf>
14. *Forecasting of the Austrian Inflation Rate* [interaktyvus]. 2007. Nadir Shahzad Virkun Tomas Sedliacik [žiūrėta 2014 m. gegužės 28 d.]. Prieiga per internetą:

- <http://homepage.univie.ac.at/tomas.sedliacik/files/seminar_papers/forecasting_of_the_austrian_inflation_rate.pdf>
15. *Forecast of hourly average wind speed with ARMA models in Navarre (Spain)* [interaktyvus]. 2005. J. L. Torres, A. García, M. De Blas, A. De Francisco [žiūrėta 2014 m. gegužės 28 d.]. Prieiga per internetą: <<http://www.sciencedirect.com/science/article/pii/S0038092X04002877>>
 16. *Time Series Analysis of Household Electric Consumption with ARIMA and ARMA Models* [interaktyvus]. 2013. Pasapitch Chujai, Nittaya Kerdprasop, and Kittisak Kerdprasop [žiūrėta 2014 m. gegužės 28 d.]. Prieiga per internetą: <http://www.iaeng.org/publication/IMECS2013/IMECS2013_pp295-300.pdf>
 17. *Bandwidth Expansion of Narrowband Speech using Linear Prediction* [interaktyvus]. 2004. Alborgo universitetas [žiūrėta 2014 m. gegužės 28 d.]. Prieiga per internetą: <<http://kom.aau.dk/group/04gr742/rapport.pdf>>
 18. *LPC modeling with speech production constrains* [interaktyvus]. 2000. Sacha Krstulovic [žiūrėta 2014 m. gegužės 28 d.]. Prieiga per internetą: <<http://publications.idiap.ch/downloads/papers/2000/sacha-sps00sk.pdf>>
 19. *Artificial Neural Networks Technology* [interaktyvus]. Toronto psichologijos universitetas [žiūrėta 2014 m. gegužės 28 d.]. Prieiga per internetą: <<http://www.psych.utoronto.ca/users/reingold/courses/ai/cache/neural2.html>>
 20. *Neural Networks* [interaktyvus]. C. Stergiou, D. Siganos [žiūrėta 2014 m. gegužės 28 d.]. Prieiga per internetą: <http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html>
 21. *Neural Network Time Series Prediction and Modeling* [interaktyvus]. 2014. MATLAB® [žiūrėta 2014 m. gegužės 28 d.]. Prieiga per internetą: <<http://www.mathworks.se/help/nnet/gs/neural-network-time-series-prediction-and-modeling.html>>
 22. *Narnet* [interaktyvus]. 2014. MATLAB® [žiūrėta 2014 m. gegužės 28 d.]. Prieiga per internetą: <<http://www.mathworks.se/help/nnet/ref/narnet.html>>
 23. *Narxnet* [interaktyvus]. 2014. MATLAB® [žiūrėta 2014 m. gegužės 28 d.]. Prieiga per internetą: <<http://www.mathworks.se/help/nnet/ref/narxnet.html>>
 24. *Timedelaynet* [interaktyvus]. 2014. MATLAB® [žiūrėta 2014 m. gegužės 28 d.]. Prieiga per internetą: <<http://www.mathworks.se/help/nnet/ref/timedelaynet.html>>
 25. *Time Series Prediction using LS-SVMs* [interaktyvus]. Marcelo Espinoza, Tillmann Falck, Johan A. K. Suykens, Bart De Moor [žiūrėta 2014 m. gegužės 28 d.]. Prieiga per internetą: <<ftp://ftp.esat.kuleuven.be/stadius/falck/estsp08.pdf>>
 26. *Prediction of hourly solar radiation using a novel hybrid model of ARMA and TDNN* [interaktyvus]. 2011. Wu Ji, Keong Chan Chee [žiūrėta 2014 m. gegužės 28 d.]. Prieiga per internetą: <<http://www.sciencedirect.com/science/article/pii/S0038092X11000259>>

PRIEDAI

A priedas. ARMA metodo pritaikymo programinis kodas

```

1. %%Duomenų nuskaitymas
2. clc
3. clear
4. x=xlsread('BobutesPaskola.xls','Lapas1', 'V4:V185');
5. kaupiu=[]; %prognozuojamu signalo reikšmiu rinkinys
6. kaupiu_e2=[];
7. klaida=[];
8. for w1=6:2:170
9. % modelio sudarymas
10. na = 2; % A daugianario eilė, kokią parinkti nežinau, reiktų teoriją
    paskaityti, kokiose situacijose kas tinka
11. nb = 4; % B daugianario eilė
12. s=x(1:w1); % signalo fragmentas modeliui apskaičiuoti
13. sys = armax(s,[na nb]);
14. K=1; % Nurodo kiek žingsnių prognozuos į priekį
15.
16. for idx1=(w1+1):(numel(x)-1)
17.     y = forecast(sys,x((idx1-w1):idx1-1),K);
18.     e=x(idx1-1)-y; % x(36) nenaudota mokymui reikšmė, y - prognozuojama
19.     kaupiu_e2 = [kaupiu_e2 e];
20.     kaupiu=[kaupiu y];
21.
22.     % 1. pabandykime atnaujinti modelį imant visus tuo metu (tai
        iteracijai)
23.     % prieinamus istorinius duomenis - naudosime ne w1, o idx1:
24.     % s=x(idx1:-1:1);
25.     % sys = armax(s,[na nb]);
26.     % % 2. pabandykime atnaujinti modelį imant naujus paskutiniuosius
        w1 elementų (tai iteracijai)
27.     s=x(idx1:-1:idx1-w1);
28.     sys = armax(s,[na nb]);
29.
30. end
31. figure(1)
32. vkk=mean(abs(kaupiu_e2).^2);
33. klaida=[klaida vkk];
34. end
35. klaida=klaida';
36. %%
37.
38. %%spektras
39. figure (2)
40. %periodogram(x(1:5));
41. %hold on
42. %periodogram(x(1:7));
43. %periodogram(x(1:35));
44.
45. periodogram(x(83:91));
46. hold on
47. periodogram(x(21:91));
48. periodogram(x(87:91))
49.

```

B priedas. NAR metodo pritaikymo programinis kodas

```

1. close all
2. clear
3. clc
4. y=xlsread('BobutesPaskola.xls','Lapas1', 'V4:V185'); % Bendras

```

```

5. % paspaudimų skaičius
6. T=mat2cell(y',1,ones(1,numel(y))); % paruošiamas teisingų atsakymų
    vektorius
7. kaupiu_Y=[]; % paruošiamas kintamasis į kurį gali būti dedamos tinklo
    išėjimo reikšmės
8. kaupiu_perf=[]; % paruošiamas kintamasis saugoti tinklo "performance"
    reikšmes
9. %% Eksperimentuojama, kai bandoma prognozuoti:
10. % dabartines y reikšmes,
11. % pagal 5-7 eil. rodiklius + 4 eil. esančias reikšmes (išėjimų istorija)
12. % jei tiriamas vienas neuronų sluoksnius
13. for sluoksniu_neuronai=1:8 % eksperimentuojama, kai pirmajame sluoksnyje
    esančių neuronų skaičius yra nuo 1 iki 8
14. for isejimu_velinimas=6:18 % eksperimentuojama, kai imamos praeitos
    išėjimų (y) reikšmės
15. net = narxnet(1:isejimu_velinimas,[sluoksniu_neuronai]);
16. net.trainParam.max_fail=40;
17. [Xs,Xi,Ai,Ts] = preparets(net,{}, {},T); % paruošiami įėjimo duomenys
    (atrenkami įėjimo pavyzdžiai pagal tai, kiek jie turi būti suvėlinti
    išėjimų atžvilgiu)
18. for mokymas=1:20
19. net = train(net,Xs,Ts,Xi,Ai);% tinklo mokymas
20. Y = net(Xs,Xi);% paleidžiamas apmokytas tinkas paduodant ()
21. perf = perform(net,Ts,Y); % vidutinė klaida?
22. kaupiu_perf=[kaupiu_perf,perf];
23. end
24. end
25. end
26. %jei tiriamas neuronų tinklas su keliais sluoksniais
27. % % for sluoksniu_neuronai=3:8 % eksperimentuojama, kai pirmajame
    sluoksnyje esančių neuronų skaičius yra nuo 3 iki 8
28. % %for antro_sluosniu_neuronai=1:8 % eksperimentuojama, kai antrajame
    sluoksnyje esančių neuronų skaičius yra nuo 1 iki 8
29. % %for isejimu_velinimas=2:8 % eksperimentuojama, kai imamos praeitos
    išėjimų (y) reikšmės
30. % %net = narxnet(1:isejimu_velinimas,[sluoksniu_neuronai
    antro_sluosniu_neuronai]);
31. % %net.trainParam.max_fail=40;
32. % % [Xs,Xi,Ai,Ts] = preparets(net,{}, {},T); % paruošiami įėjimo duomenys
    (atrenkami įėjimo pavyzdžiai pagal tai, kiek jie turi būti suvėlinti
    išėjimų atžvilgiu)
33. % %for mokymas=1:20
34. % %net = train(net,Xs,Ts,Xi,Ai);% tinklo mokymas
35. % %Y = net(Xs,Xi);% paleidžiamas apmokytas tinkas paduodant ()
36. % %perf = perform(net,Ts,Y); % vidutinė klaida
37. % %kaupiu_perf=[kaupiu_perf,perf];
38. % %end
39. % %end
40. % %end
41. % %end
42. %% Rezultatų analizė
43. % Vizuali analizė
44. figure(10)
45. plotresponse(Ts, Y(end-numel(Ts)+1:end));% atvaizduojami paskutiniojo
    bandymo rezultatai
46. figure(11)
47. plotresponse(Ts, kaupiu_Y(1,end-numel(Ts)+1:end));% atvaizduojami
    pirmojo bandymo rezultatai

```

C priedas. NARX metodo pritaikymo programinis kodas

```

1. close all
2. clear
3. clc

```

```

4. y=xlsread('BobutesPaskola.xls','Lapas1', 'V44:V130'); % Bendras
    paspaudimųskaičiu
5. x1=xlsread('BobutesPaskola.xls','Lapas1', 'B44:B130');% Atmetimo
    rodiklis (bounce rate),% GA
6. x2=xlsread('BobutesPaskola.xls','Lapas1', 'C44:C130');% Vidutinė vizito
    trukmė, s GA
7. X=mat2cell([x1';x2'],2,ones(1,numel(x1))); % paruošiama trijų įėjimų
    pavyzdžių matrica iš trijų rodiklių
8. T=mat2cell(y',1,ones(1,numel(y))); % paruošiamas teisingų atsakymų
    vektorius
9. kaupiu_Y=[]; % paruošiamas kintamasis į kurį gali būti dedamos tinklo
    išėjimo reikšmės
10. kaupiu_perf=[]; % paruošiamas kintamasis saugoti tinklo "performance"
    reikšmės
11. %% Eksperimentuojama, kai bandoma prognozuoti:
12. % dabartinės y reikšmės,
13. % pagal 5-7 eil. rodiklius + 4 eil. esančias reikšmes (išėjimų istorija)
14. % jei tiriamas vienas neuronų sluoksnius
15. for sluoksniu_neuronai=3:8 % eksperimentuojama, kai pirmajame sluoksnyje
    esančių neuronų skaičius yra nuo 3 iki 8
16. for iejimu_velinimas=1:17 % eksperimentuojama, kai imamos dabartinės
    rodiklių reikšmės (iejimu_velinimas=1), anksčiau gautos
    (iejimu_velinimas=2) ir t.t.
17. for isejimu_velinimas=2:8 % eksperimentuojama, kai imamos praeitos
    išėjimų (y) reikšmės
18. net =
    narxnet(1:iejimu_velinimas,1:isejimu_velinimas,[sluoksniu_neuronai]);
19. net.trainParam.max_fail=40; %(number of max Validation checks, default
    6)
20. [Xs,Xi,Ai,Ts] = preparets(net,X,{},T); % paruošiami įėjimo duomenys
    (atrenkami įėjimo pavyzdžiai pagal tai, kiek jie turi būti suvėlinti
    išėjimų atžvilgiu)
21. for mokymas=1:20
22. net = train(net,Xs,Ts,Xi,Ai);% tinklo mokymas
23. Y = net(Xs,Xi,Ai);% paleidžiamas apmokytas tinkas paduodant ()
24. perf = perform(net,Ts,Y); % vidutinė klaida?
25. kaupiu_perf=[kaupiu_perf,perf];
26. end
27. end
28. end
29. end
30.
31. % % jei tiriamas neuronų tinklas su keliais sluoksniais
32. % for sluoksniu_neuronai=3:8 % eksperimentuojama, kai pirmajame
    sluoksnyje esančių neuronų skaičius yra nuo 3 iki 8
33. %for antro_slauksniu_neuronai=1:8 % eksperimentuojama, kai antrajame
    sluoksnyje esančių neuronų skaičius yra nuo 1 iki 8
34. %for iejimu_velinimas=1:17 % eksperimentuojama, kai imamos dabartinės
    rodiklių reikšmės (iejimu_velinimas=1), anksčiau gautos
    (iejimu_velinimas=2) ir t.t.
35. %for isejimu_velinimas=2:8 % eksperimentuojama, kai imamos praeitos
    išėjimų (y) reikšmės
36. %net =
    narxnet(1:iejimu_velinimas,1:isejimu_velinimas,[sluoksniu_neuronai
    antro_slauksniu_neuronai]);
37. % net.trainParam.max_fail=40;
38. [Xs,Xi,Ai,Ts] = preparets(net,X,{},T); % paruošiami įėjimo duomenys
    (atrenkami įėjimo pavyzdžiai pagal tai, kiek jie turi būti suvėlinti
    išėjimų atžvilgiu)
39. %for mokymas=1:20
40. %net = train(net,Xs,Ts,Xi,Ai);% tinklo mokymas
41. %Y = net(Xs,Xi,Ai);% paleidžiamas apmokytas tinkas paduodant ()
42. %perf = perform(net,Ts,Y); % vidutinė klaida?
43. %kaupiu_perf=[kaupiu_perf,perf];

```

```

44. %end
45. %end
46. %end
47. %end
48. %end
49. %% Rezultatų analizė
50. % Vizuali analizė
51. figure(10)
52. plotresponse(Ts, Y(end-numel(Ts)+1:end));% atvaizduojami paskutiniojo
    bandymo rezultatai
53. figure(11)
54. plotresponse(Ts, kaupiu_Y(1,end-numel(Ts)+1:end));% atvaizduojami
    pirmojo bandymo rezultatai

```

D priedas. TDNN metodo pritaikymo programinis kodas

```

1. close all
2. clear
3. clc
4. y=xlsread('BobutesPaskola.xls','Lapas1','V44:V130'); % Bendras
5. % paspaudimų skaičius
6. x1=xlsread('BobutesPaskola.xls','Lapas1','B44:B130');% Atmetimo
    rodiklis (bounce rate),% GA
7. x2=xlsread('BobutesPaskola.xls','Lapas1','C44:C130');% Vidutinė vizito
    trukmė, s GA
8. X=mat2cell([x1';x2'],2,ones(1,numel(x1))); % paruošiama trijų įėjimų
    pavyzdžių matrica iš trijų rodiklių
9. T=mat2cell(y',1,ones(1,numel(y))); % paruošiamas teisingų atsakymų
    vektorius
10. kaupiu_Y=[]; % paruošiamas kintamasis į kurį gali būti dedamos tinklo
    išėjimo reikšmės
11. kaupiu_perf=[]; % paruošiamas kintamasis saugoti tinklo "performance"
    reikšmes
12. %% Eksperimentuojama, kai bandoma prognozuoti:
13. % dabartines y reikšmes,
14. % pagal
15. for sluoksniu_neuronai=1:8 % eksperimentuojama, kai sluoksnyje esančių
    neuronų skaičius gali būti nuo 1 iki 8
16. for velinimas=1:18 % eksperimentuojama, kai imamos dabartinės rodiklių
    reikšmės (velinimas=1), anksčiau gautos (velinimas=2) ir t.t.
17. net = timedelaynet(1:velinimas,sluoksniu_neuronai);
18. net.trainParam.max_fail=40;
19. [Xs,Xi,Ai,Ts] = preparets(net,X,T); % paruošiami įėjimo duomenys
    (atrenkami įėjimo pavyzdžiai pagal tai, kiek jie turi būti suvėlinti
    išėjimų atžvilgiu)
20. for mokymas=1:20
21. net = train(net,Xs,Ts,Xi,Ai);% tinklo mokymas
22. Y = net(Xs,Xi,Ai);% paleidžiamas apmokytas tinklas paduodant ()
23. perf = perform(net,Ts,Y); % vidutinė klaida?
24. kaupiu_perf=[kaupiu_perf,perf];
25. end
26. end
27. end
28. %% Rezultatų analizė
29. % Vizuali analizė
30. figure(10)
31. plotresponse(Ts, Y(end-numel(Ts)+1:end));% atvaizduojami paskutiniojo
    bandymo rezultatai
32. figure(11)
33. plotresponse(Ts, kaupiu_Y(1,end-numel(Ts)+1:end));% atvaizduojami
    pirmojo bandymo rezultatai

```

E priedas. LPC metodas

```

1. %Duomenų nuskaitymas
2. clc
3. clear
4. x=xlsread('BobutesPaskola.xls','Lapas1', 'V4:V185');
5. %w1=91;
6. % w1=28;
7. langų_plociai = 8:1:170;
8. rezultatai=zeros(numel(langų_plociai),7);% 1-6 stulpeliai skirti klaidai
   įrašyti
9. rezultatai(:,7)=langų_plociai;% į 7 stulpelį įrašomi naudoti langų
   pločiai
10. % rezultatai(:,2)=1:6;% į 2 stulpelį įrašomos naudotos filtro eilės
11. langų_indeksas=[langų_plociai;1:numel(langų_plociai)];
12. for w1=langų_plociai
13. for p=1:8
14. kaupiu=[]; %prognozuojamu signalo reikšmių rinkinys
15. kaupiu_e2=[];
16. s=x(w1:-1:1); % imamos pirmosios w1 signalo reikšmės
17. a=lpc(s,p); % pirmosioms signalo reikšmėms apskaičiuojame filtro
   koeficientus
18. for idx1=(w1+1):(numel(x)-1) % imame signalo dalį pradedant nuo w1+1
   elemento
19. % prognozavimui:
20. s2 = x(idx1:-1:(idx1-p)); % imama viena iteracija naujesnė signalo
   dalis, pakanka paskutiniųjų "p" reikšmių
21. if p==1
22. y=-a(2)*s(1); % p=1; skaičiuojame y reikšmę, naudodami w1 signalo
   reikšmę
23. y2=-a(2)*s2(1);% p=1; skaičiuojame (prognozuojame) y2 reikšmę ,
   naudodami w1+1 reikšmę
24. else if p==2
25. y=-(a(2)*s(1)+a(3)*s(2)); % p=2
26. y2=-(a(2)*s2(1)+a(3)*s2(2));%p=2
27. else if p==3
28. y=-(a(2)*s(1)+a(3)*s(2)+a(4)*s(3)); % p=3
29. y2=-(a(2)*s2(1)+a(3)*s2(2)+a(4)*s2(3));% p=3
30. else if p==4
31. y=-(a(2)*s(1)+a(3)*s(2)+a(4)*s(3)+a(5)*s(4)); % p=4
32. y2=-(a(2)*s2(1)+a(3)*s2(2)+a(4)*s2(3)+a(5)*s2(4)); % p=4
33. else if p==5
34. y=-(a(2)*s(1)+a(3)*s(2)+a(4)*s(3)+a(5)*s(4)+a(6)*s(5)); % p=5
35. y2=-(a(2)*s2(1)+a(3)*s2(2)+a(4)*s2(3)+a(5)*s2(4)+a(6)*s2(5)); % p=5
36. else if p==6
37. y=-(a(2)*s(1)+a(3)*s(2)+a(4)*s(3)+a(5)*s(4)+a(6)*s(5)+a(7)*s(6)); %
   veiks tik kai p=6,
38. y2=-(a(2)*s2(1)+a(3)*s2(2)+a(4)*s2(3)+a(5)*s2(4)+a(6)*s2(5)+a(7)*s2(6));
39. else if p==7
40. y=-
   (a(2)*s(1)+a(3)*s(2)+a(4)*s(3)+a(5)*s(4)+a(6)*s(5)+a(7)*s(6)+a(8)*s(7)
   ); % veiks tik kai p=7,
41. y2=-
   (a(2)*s2(1)+a(3)*s2(2)+a(4)*s2(3)+a(5)*s2(4)+a(6)*s2(5)+a(7)*s2(6)+a(8)
   )*s2(7));
42. else p==8
43. y=-
   (a(2)*s(1)+a(3)*s(2)+a(4)*s(3)+a(5)*s(4)+a(6)*s(5)+a(7)*s(6)+a(8)*s(7)
   +a(9)*s(8)); % veiks tik kai p=8,
44. y2=-
   (a(2)*s2(1)+a(3)*s2(2)+a(4)*s2(3)+a(5)*s2(4)+a(6)*s2(5)+a(7)*s2(6)+a(8)
   )*s2(7)+a(9)*s2(8));
45. end

```

```

46. end
47. end
48. end
49. end
50. e=x(idx1)-y; % x(36) nenaudota mokymui reikšmė, y - prognozuojama
51. e2=x(idx1+1)-y2;
52. kaupiu_e2 = [kaupiu_e2 e2];
53. kaupiu=[kaupiu y];
54. s=x(idx1:-1:idx1-w1);
55. a=lpc(s,p);
56. end
57. end
58. end
59. figure(3)
60. plot(1:numel(x)-1,x(1:end-1),'b',w1+1:numel(x)-
    1,kaupiu,'g',w1+1:numel(x)-1,kaupiu_e2,'r')
61. % w1 yra langas "a" koeficientams skaičiuoti - prognozės filtro mokymui
62. grid on
63. l = line((w1+1)*[1 1], get(gca, 'ylim'));
64. set(l, 'color', [0,0,0]);
65. legend('Išmatuoti duomenys','Prognozuoti duomenys','Prognozavimo
    klaida','Prognozavimo pradžia');
66.
67. % Vidutinė klaida
68. mean(abs(kaupiu_e2).^2)
69. % %
70. % % figure
71. % % plot(1:numel(x), x);
72. % % l = line(w1*[1 1], get(gca, 'ylim'));
73. % % set(l, 'color', [0,0,0]);
74. % % hold on;
75. % % plot((w1+1):numel(x), (y((w1+1):numel(x))), 'g');
76. % % legend('Simulated data','Predicted data');

```