

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
PROGRAMŲ INŽINERIJOS KATEDRA



Ališer Haidari

# VIRTUALAUS FINANSŲ RINKOS MODELIO PROGRAMINIS REALIZAVIMAS IR TYRIMAS

Baigiamasis magistro darbas

**Darbo vadovas:** prof. habil. dr.  
Jonas Mockus

KAUNAS,  
2011



KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
PROGRAMŲ INŽINERIJOS KATEDRA

Ališer Haidari

# VIRTUALAUS FINANSŲ RINKOS MODELIO PROGRAMINIS REALIZAVIMAS IR TYRIMAS

Baigiamasis magistro darbas

**Recenzentas:** prof. dr.  
Alfonsas Misevičius

**Darbo vadovas:** prof. habil. dr.  
Jonas Mockus

**Atliko:** IFM-9/2 gr. studentas  
Ališer Haidari

KAUNAS,  
2011



---

# Anotacija

Virtualus akcijų biržos modelis (SEGM), modeliuojantis virtualių investuotojų elgseną, kurie prie fiksuotos palūkanų normos naudoja fiksuotą pirkimo/pardavimo maržą, buvo pristatytas [1, 2]. Kuriamas patobulintas virtualios akcijų biržos modelis (USEGM), kurio privalumas yra Nešo pusiausvyros (NE) taikymas modeliams, kurie priskiria pirkimo/pardavimo maržas bei gaunamų ir skolinamų palūkanų procentinį skirtumą (angl. *haircuts*; HC) investuotojams dinamiškai. Taipogi pridėtas naujumas - transakcijos mokesčio įvertinimas. Šių naujovių įvedimas leidžia modeliuoti ir tirti rinkos likvidumą įvairiais scenarijais [3].

Šiame dokumente yra detalai nagrinėjami prognozavimo (angl. *prediction*) modeliai. Virtualių investuotojų atvaizdavimui, kurie naudoja tiesines naudingumo funkcijas, USEGM įtraukia Vynerio (*Wiener*) procesą-modelį  $RW$  ir du autoregresinius modelius:  $AR(p)$ , kuris optimizuoja mažiausius kvadratinus nuokrypius, bei  $AR - ABS(p)$ , kuris optimizuoja absoliučias reikšmes.  $AR - ABS(p)$  prognozavimo modelis nebuvo naudojamas SEGM modelyje ir yra įtraukimas į USEGM. Šiame darbe bus tiriami bei tobulinami išvardinti prognozių sudarymo modeliai. USEGM modelio esminė užduotis yra ne prognozuoti realias rinkos vertes, bet modeliuoti rinkos laiko eilutes, kurioms turi įtaką virtualių investuotojų sudaromos prognozės. Šis darbas remiasi SEGM modeliu, kuriame tyrimo metu paaiškėjo, kad prognozavimai vyksta ne visiškai teisingai. Tyrimo metu įrodoma, kad visi šie prognozavimo modeliai -  $RW$ ,  $AR(p)$  bei  $AR - ABS(p)$  - yra pritaikomi naujam kuriamam USEGM modeliui ir, juos patobulinus, atlieka prognozes korektiškai.

---

# Summary

A simple Stock Exchange Game Model (SEGM) was introduced in [1, 2] to simulate the behavior of several stockholders using fixed buying-selling margins at fixed bank yield. The new model, Extended Stock Exchange model (USEGM), is under development, which advantage is application of the Nash Equilibrium (NE) to strategies that define buying-selling margins and bank haircuts (HC) dynamically. In addition, USEGM includes the transaction costs to reflect the reality better. This enables us to simulate market illiquidity in many different scenarios [3].

This paper describes in deep the mathematical models of making the financial prediction . To represent virtual users that prefer linear utility functions, USEGM adds Wiener process-model  $RW$  and two autoregressive models:  $AR(p)$  which optimising least square deviations and  $AR - ABS(p)$ , which optimising the absolute values. The  $AR - ABS(p)$  prediction model were not used previously in SEGM model and currently is included in USEGM as an improvement. In this work the prediction methods described above will be investigated. The objective of USEGM is not forecasting, but simulation of financial time series that are affected by predictions of the virtual participants. This work is based under SEGM model in which in the time of investigation were revealed inconsistencies in predictions. In the time of explicit investigation are proved, that all of these prediction methods -  $RW$ ,  $AR(p)$  and  $AR - ABS(p)$  are suitable for new USEGM model to make a well predictions. In the time of investigation the found inconsistencies in prediction algorithms were removed and finally the methods are working correctly.

# Turiny

Terminų žodynas	1
Santrumpų žodynas	3
<b>1 ĮVADAS</b>	<b>5</b>
1.1 Projektas	6
1.2 Bendrinės sąvokos	7
<b>2 PROJEKTO ANALIZĖ</b>	<b>9</b>
2.1 Prognozavimų modelių tyrimas	9
2.1.1 Prognozės sąvoka	9
2.1.2 Prognozavimo posistemės uždaviniai ir tikslai	10
2.1.3 ARMA modelis	11
2.1.3.1 $AR(p)$ modelis	12
2.1.3.2 $AR-ABS(p)$ modelis	15
2.1.3.3 $RW$ ( <i>Wiener</i> ) modelis	15
2.1.4 Paklaidų įvertinimas	16
2.1.4.1 Paklaidų skaičiavimo metodai	16
2.2 USEGM karkasas	18
2.2.1 Dalykinės srities analizė	18
2.2.2 Siūloma įgyvendinimo logika	18
2.2.3 Kritinių sekcijų ( <i>CS</i> ) nustatymas	19
2.2.4 Paralelinio programavimo karkasas	19
<b>3 PROJEKTINĖ DALIS</b>	<b>21</b>
3.1 Biržos žaidimo modelis	21
3.2 Nešo pusiausvyra	22
3.3 Kuriamas USEGM modelis	25
3.3.1 Pagrindiniai naujumo aspektai	25
3.4 Architektūriniai sprendimai	25

3.4.1 Dizaino šablonai . . . . .	26
3.5 Testavimo įrankiai . . . . .	26
<b>4 REALIZUOTO PROJEKTO TYRIMAS</b>	<b>27</b>
4.1 Prognozavimų posistemės tobulinimas . . . . .	27
4.1.1 Kiti prognozavimo modeliai . . . . .	27
4.1.2 Dirbtiniai neuroniniai tinklai ( <i>ANN</i> ) . . . . .	28
4.2 Karkaso tobulinimas . . . . .	29
4.2.1 Architektūra . . . . .	29
4.2.2 Grafinių resursų panaudojamumas ( <i>GPU</i> ) . . . . .	30
<b>5 PROGNOZAVIMO MODELIŲ EKSPERIMENTINIS TYRIMAS</b>	<b>33</b>
5.1 Parametrų optimizavimo tyrimas ( <i>1-as tyrimo etapas</i> ) . . . . .	34
5.1.1 <i>RW</i> modelis . . . . .	34
5.1.2 <i>AR(p)</i> modelis . . . . .	34
5.1.3 <i>AR-ABS(p)</i> modelis . . . . .	37
5.1.4 Gautų rezultatų apibendrinimas . . . . .	37
5.2 Prognozavimo tikslumo tyrimas ( <i>2-as tyrimo etapas</i> ) . . . . .	38
5.2.1 <i>RW</i> modelis . . . . .	40
5.2.2 <i>AR(p)</i> modelis . . . . .	41
5.2.3 <i>AR-ABS(p)</i> modelis . . . . .	43
5.2.4 Gautų rezultatų apibendrinimas . . . . .	45
5.3 Tyrimo Rezultatų įvertinimas . . . . .	45
5.3.1 <i>RW</i> modelio įvertinimas . . . . .	45
5.3.2 Autoregresinių modelių įvertinimas . . . . .	46
5.3.3 Metodų taikymas . . . . .	47
<b>IŠVADOS</b>	<b>49</b>
<b>LITERATŪRA</b>	<b>51</b>
<b>PADĖKA</b>	<b>55</b>
<b>6 PRIEDAI</b>	<b>57</b>
6.1 Priedas 1. Projekto pasiekiamumas . . . . .	57
6.2 Priedas 2. Papildomi prognozių tyrimo grafiniai rezultatai . . . . .	58
6.3 Priedas 3. USEGM generuojamos skaičiavimų išvestys . . . . .	62
6.4 Priedas 4. Papildomi optimizavimo rezultatai . . . . .	65



# Iliustracijų sąrašas

3.1	Pradiniai SEGM nustatymo parametrai . . . . .	23
3.2	Skirtingų žaidėjų modeliuojamas pelnas SEGM modelyje . . . . .	24
3.3	Modeliuojamos turto kainos ir maržų (slenksčių) apribojimai SEGM modelyje . . . . .	24
3.4	USEGM modelio architektūra . . . . .	25
5.1	<i>RW</i> , realių ir bazinių prognozavimų reikšmių laiko eilutės . . . . .	35
5.2	<i>AR(1)</i> , parametrų optimizavimas naudojant realius duomenis . . . . .	36
5.3	<i>AR(1)</i> , parametrų optimizavimas naudojant sumodeliuotus duomenis . . . . .	36
5.4	<i>AR-ABS(8)</i> , parametrų optimizavimas naudojant realius duomenis . . . . .	37
5.5	<i>AR-ABS(8)</i> , parametrų optimizavimas naudojant sumodeliuotus duomenis . . . . .	38
5.6	Duomenų analizavimo schema prognozių sudarymui . . . . .	40
5.7	<i>RW</i> , prognozių sudarymas realiems duomenims . . . . .	40
5.8	<i>RW</i> , prognozių sudarymas modeliuojamiems duomenims . . . . .	41
5.9	<i>AR(1)</i> , prognozių sudarymas realiems duomenims . . . . .	42
5.10	<i>AR(1)</i> , prognozių sudarymas sumodeliuotiems duomenims . . . . .	43
5.11	<i>AR-ABS(8)</i> , prognozių sudarymas realiems duomenims . . . . .	44
5.12	<i>AR-ABS(8)</i> , prognozių sudarymas sumodeliuotiems duomenims . . . . .	44
5.13	Modelių algoritmų sudėtingumo palyginimas . . . . .	47
6.1	<i>AR(1)</i> , parametrų optimizavimas, <i>MSFT</i> , $RMSE = 0.025502938$ . . . . .	58
6.2	<i>AR(1)</i> , parametrų optimizavimas, sugeneruoti duom., $RMSE = 0.109073974$ . . . . .	59
6.3	<i>AR(1)</i> , prognozė, <i>MSFT</i> , $RMSE = 5.615557442$ . . . . .	59
6.4	<i>AR(1)</i> , prognozė, sugeneruoti duom., $RMSE = 0.496580846$ . . . . .	60
6.5	<i>AR-ABS(8)</i> , parametrų optimizavimas, <i>MSFT</i> , $RMSE = 0.043231444$ . . . . .	60
6.6	<i>AR-ABS(8)</i> , parametrų optimizavimas, sugeneruoti duom., $RMSE = 0.038376583$ . . . . .	61
6.7	<i>AR-ABS(8)</i> , prognozė, <i>MSFT</i> , $RMSE = 0.038376583$ . . . . .	61
6.8	<i>AR-ABS(8)</i> , prognozė, sugeneruoti duom., $RMSE = 0.038376583$ . . . . .	62
6.9	AR modelio skaičiavimo išvestys . . . . .	63
6.10	AR-ABS modelio skaičiavimo išvestys . . . . .	64
6.11	Parametrų optimizavimo statistika. 1-5 . . . . .	66

6.12 Parametrų optimizavimo statistika. 2-5 . . . . .	67
6.13 Parametrų optimizavimo statistika. 3-5 . . . . .	68
6.14 Parametrų optimizavimo statistika. 4-5 . . . . .	69
6.15 Parametrų optimizavimo statistika. 5-5 . . . . .	70
6.16 RMSE priklausomybė nuo parametro $p$ . . . . .	71
6.17 AME priklausomybė nuo parametro $p$ . . . . .	72
6.18 RMSE pasiskirstymas $AR(p)$ atžvilgiu . . . . .	73
6.19 RMSE pasiskirstymas $AR - ABS(p)$ atžvilgiu . . . . .	74

# Lentelių sąrašas

3.1	USEGM naudojami dizaino šablonai . . . . .	26
5.1	$RW$ , gauti duomenys po optimizacijos . . . . .	34
5.2	Prognozavimui naudojamos baltojo triukšmo vertės . . . . .	39
5.3	$RW$ , ribinės ir suvidurkintos paklaidų reikšmės . . . . .	41
5.4	$AR(1)$ , prognozavimo kokybės įvertinimas. . . . .	42
5.5	$AR-ABS(8)$ , prognozavimo kokybės įvertinimas. . . . .	45

# Terminų žodynas

**Portfelis** – investuotojo turimas tam tikras akcijų, obligacijų ar tam tikrų vertybinių popierių kiekis, vadinamas investiciniu portfeliu (angl. portfolio);

**Subversion, SVN** – programinė įranga projekto išėties teksto versijų kūrimui (angl. *Version Control System*);

**Laiko eilutės** – šiame darbo kontekste - akcijų biržos realūs ar modelio sugeneruoti rinkos duomenys, kvotos ar indeksų reikšmės, kurios kito per tam tikrą laiką (angl. *time series*);

**Deviacija** – vidutinis kvadratinis nuokrypis;

**Marža** – ekonomikos terminas, nusakantis skirtumą tarp pirkimo ir pardavimo kainos (angl. *margin*);

**Likvidumas** – vertybinių popierių arba materialinių vertybių pavertimo grynaisiais pinigais galimybė;

**Haircut, HC** – ekonomikos terminas, nusakantis skirtumą tarp skolinamų palūkanų dydžio (pagal kurias skolina vartotojui pinigus) ir banko mokamų palūkanų normos (vartotojui už tam tikrus laikomus indėlius);

**Nešo pusiausvyra** – žaidimų teorijoje Nešo pusiausvyra vadinami vieno ar daugiau žaidėjų žaidimo sprendimai, kuriuose nė vienas žaidėjas negali padidinti savo laimėjimo vienpusiškai pakeitęs savo sprendimą, todėl reikalingas bendras žaidimo dalyvių pasirinkimas (angl. *Nash Equilibrium*);

**Java** – bendros paskirties programavimo kalba, kuria įgyvendintas USEGM projektas;

**Autoregresija** – grįžimas į praeitį (angl. *autoregression*);

**Atidarymo / uždarymo kainos** – *atidarymo kaina* - tai kaina, kuri nustatoma dienos pradžioje prieš pradėdant prekiauti vertybiniais popieriais (čia - akcijomis), o *uždarymo kaina* - tai kaina, kuri yra nustatoma prekybos uždarymo metu (angl. *open / close price*);

**Karkasas** – programinės įrangos tam tikra sistema ar biblioteka, įgyvendinanti žemesnio lygio funkcionalumą kuriamos sistemos atžvilgiu (angl. *framework*).



# Santrumpų žodynas

USEGM – patobulintas virtualus akcijų biržos žaidimo modelis, šiuo tyrimo objektas (angl. *Extended Virtual Stock Exchange Game Model*);

VSE – virtuali akcijų birža (angl. *Virtual Stock Exchange*);

SEGM – virtualus akcijų biržos žaidimo modelis (angl. *Stock Exchange Game Model*);

ANN – dirbtiniai neuroniniai tinklai (angl. *Artificial Neural Network*);

SVN, Subversion – programinė įranga projekto išeities teksto versijų kūrimui (angl. *Version Control System*);

GIT – paskirstytoji programinė įranga projekto išeities teksto versijų kūrimui (angl. *Distributed Revision Control System*);

XML – (angl. *Extensible Markup Language*);

CSV – bylos tipas, kuriame duomenys atskiriami kableliais. Naudojamas USEGM įėjimo testams įvesti ir išėjimo rezultatams išvesti (angl. *Comma-Separated Values*);

ARMA – autoregresinis slenkančio vidurkio matematinis prognozavimo modelis ilgam periodui (angl. *Autoregressive Moving Average*);

AR – autoregresinis matematinis prognozavimo modelis trumpam periodui;

MA – autoregresinio modelio slenkančioji dalis (angl. *Moving Average*);

AR-ME – autoregresinis maksimalios entropijos Burgo (*Burg*) prognozavimo modelis trumpam periodui (angl. *Autoregressive Maximal Entropy*);

ARIMA – autoregresinis integracijos slenkančio vidurkio prognozavimo modelis, kuriame įtraukiami papildomi faktoriai lemiantys prognozę ilgam periodui (angl. *Autoregressive Integrated Moving Average*);

ARFIMA – autoregresinis dalinės integracijos slenkančio vidurkio prognozavimo modelis, kurame įtraukiami papildomi faktoriai lemiantys prognozę ilgam periodui (angl. *Autoregressive Function Integrated Moving Average*);

AR-ABS – autoregresinis absoliučių verčių prognozavimo matematinis modelis trumpam periodui (angl. *Autoregressive Absolute Values*);

RW – tai prognozavimo metodas, dar žinomas kaip Vynerio procesas (angl. *Random Walk, Wiener process*);

- ANN-FF – neuroninis tinklas turintis tiesioginę perdavimo architektūrą (angl. *feed-forward*);
- ANN-ART1 – neuroninis tinklas su *ART1* architektūra (angl. *Adaptive Resonance Theory 1*);
- CPU – procesorius (angl. *Central Processing Unit*);
- GPU – vaizdo procesorius (angl. *Graphics Processing Unit*);
- CS, KS – kritinė kodo dalies vieta, kurią vykdo tik viena CPU gija tam tikru laiko momentu (angl. *Critical Section*);
- RMS(E) – vidutinė kvadratinės šaknies paklaida (angl. *Root Mean Square (Error)*);
- AME – absoliučioji vidutinė paklaida (angl. *Absolute Mean Error*);
- NE – Nešo pusiausvyra (angl. *Nash Equilibrium*);
- HC, Haircut – ekonomikos terminas, nusakantis skirtumą tarp skolinamų palūkanų dydžio (pagal kurias skolina vartotojui pinigus) ir banko mokamų palūkanų normos (vartotojui už tam tikrus laikomus indėlius);

# Skyrius 1

## ĮVADAS

2007, 2008-ais metais finansų sektorius patiria neeilinius įvykius – krizę, kuri glaudžiai susijusi su rinkos nepastovumu [4, 3].

Buvusiai nepalankiai ir įtemptai situacijai finansų sektoriuje, vis dažniau atsižvelgėme į turimas galimybes ar būdus, kurie nurodo, kaip aplenkti situaciją, išvengti biudžeto nuosmukio ar siūlo būdus šiai padėčiai keisti reikiama linkme [5]. Aktyviai vykdomi tyrimai šioje srityje, siekiant nustatyti galimus metodus, sušvelninsiančius pasekmes: kuriami modeliavimo modeliai, kurie paaikškins finansų rinkos dominančias savybes ar dėsningumus [3].

Vyrauja be galo daug nuomonių bei sprendimų. Vieni remiasi prognozavimu, paremtu prielaidomis, jog ateities įvykiai yra visiškai ar bent dalinai paremti anksčiau stebėtais ar praeities įvykiais, kiti - euristiniais metodais [3, 6]. Įmanoma tik dalinai nuspėti ir priartėti prie teisingo pasirinkimo. Bendru atžvilgiu galime teigti, jog ateities finansiniai rinkos duomenys priklauso nuo spėjimų bei subjektų elgsenos. Ekonomikai bei finansų analitikai bando išsiaiškinti, kas daro įtaką šiam procesui neskaitant praeities įvykių; ieškoma sprendimų, išeičių. Kyla klausimas: ar įmanoma ateityje išvengti panašių ekonomikos situacijų pritaikius sistemas su tam tikrais prognozavimo modeliais? [7]. Atsakymas nėra vienprasmiškas ir jokiū būdu negalimas atsakymas “taip” arba “ne”. Atsakymas - “galbūt”.

Šiuo metu užsienio rinkoje sąlyginai gausu virtualių akcijų biržos modelių (VSE), kurie įgyvendinti internetinių svetainių pagrindu ar, keli jų, - aplikacijos. Tai tokie modeliai, kaip Marketwatch.com, Howthemarketworks.com, Forevex.com ar TradeMonster.com. Kiekvienam vartotojui, susidomėjusiam šia sritimi (realia akcijų birža), yra puiki galimybė išbandyti savo jėgas perkant ir parduodant vertybinius popierius netikrais pinigais ir stebint savo sėkmę – gautą pelną. Ar vartotojas naujokas, ar profesionalas, - visuomet atras naujų metodikų ir būdų asmeninei sėkmei didinti bei kvalifikacijai kelti. Šie virtualūs akcijų biržos modeliai dažniausiai laisvai prieinami: registracija nemokama. Vartotojas juose gali kurti asmenines strategijas, sudaryti portfelį, valdyti riziką, už papildomą mokestį nusipirkti asmenines pamokas – treniruotes. Pranašesni modeliai teikia tokias galimybes, kaip laiko eilučių peržiūrą realiu metu ar net kainų prognozavimą į ateitį pasitelkus realius duomenis



[8].

Užsienio rinkoje modelių, panašių į USEGM modelį, programinių realizacijų tyrimų tikslais nebuvo rasta. Čia reiktų pabrėžti, jog VSE įgyvendinimai užpildo tik pirmąją projekto tikslų dalį – įgyvendina patį modelį, tačiau neatlieka su jo generuojamais duomenimis analizių: neatsižvelgia į tai, ką mano investuotojai bei nesudaro galimų prognozių. Šie modeliai yra mėgėjiški, paremti tikrais rinkos duomenimis bei surogatine valiuta. Esmėje, tai nėra tyrimų modeliai. Tai – mokomosios aplikacijos, skirtos neprofesionaliems vartotojams (finansų sektoriaus) išmokti valdyti akcijas biržoje ir tobulinti savo įgūdžius. Šios priemonės neįvertina pokyčių, neprognozuoja galimų rinkos verčių, neveda statistinių duomenų, kurie gali būti panaudoti vėlesniems tyrimams. Kitaip tariant, tai žaidimas netikrais pinigais. Šios programos naudoja realius rinkos duomenis, vėlinamus keliom dešimtmin minučių [9, 7].

Lietuvos rinkoje tiesioginių VSE modelių realizacijų nėra. Dabartiniu metu atliekami tyrimai mokslinėse įstaigose – universitetuose, naudojant supaprastintus modelius - simulatorius bei analizuojant sugeneruotus duomenis [2, 3]. Bandoma išaiškinti priežastis, kas lemia bei įtakoja finansinius nuosmukius ir kaip juos aplenksti. Taikomi įvairūs matematiniai modeliai bei aktyviai vystomos prognozavimų neuroniniais tinklais (ANN) daugiaagentės sistemos, kurios adaptyviai prisitaikytu prie pakitusios aplinkos (pvz. krizė) ir padėtų nuspėti rytdienos kainas rinkoje įvertinant rizikos faktorius (mažinama išėigos paklaida, sugeneruota netiesinės funkcijos) [10].

Pagrindinis VSE bruožas toks, kad VSE virtualumas pasireiškia realių investuotojų strategijomis netikrais pinigais bei, galimas atvejis, prognozių sudarymais. Realūs investuotojai - tai subjektai, žmonės, kurie priima tam tikrus sprendimus ko pasekoje virtualizacija pasiekama rinkos modeliavime. USEGM modelio tikslas, kitaip nei VSE, yra tyrinėti susietumą tarp realių duomenų ir Nešo pusiausvyros (NE) ir išsiaiškinti kitus galimus gautus rezultatus naudojant USEGM modelį. T.y. modeliuoti rinką laikantis tam tikrų taisyklių ir tyrinėti rezultatus, kurie gali parodyti, kokie rinkos žaidėjų veiksmai yra optimalūs (optimali strategija maksimizuojanti pelną), o kokie - ne. Dinaminis pirkimų-pardavimų maržos bei HC nustatymas sudaro galimybę simuliuoti finansų rinkos likvidumo nebuvimą, kas yra krizės susidarymo priežastis [3].

Abstrakčiai apžvelgėme du skirtingus akcijų biržos modelius VSE ir USEGM, esminius skirtumus bei panašumus. USEGM modelio struktūra susideda iš prognozavimo posistemės bei virtualios akcijų biržos logikos dalies (pirkimai / pardavimai, maržų bei HC nustatymai, kainos dydžio nustatymai atidarymo bei uždarymo metu ir kitų manipuliacijų).

## 1.1 Projektas

Šio darbo tikslas yra įgyvendinti prognozavimų posistemę USEGM modeliui taikant autoregresinius matematinius modelius ( $RW$ ,  $AR(p)$ ,  $AR - ABS(p)$ ) bei atlikti jų veikimo analizę su pasirinktais duomenimis. Projekto naujumas -  $AR - ABS(p)$  optimizavimo modelio įtraukimas į USEGM

modelį, kuris nebuvo naudojamas SEGM modelyje [2]. Kadangi, pagal specifikaciją [3], šis modelis turi dirbti multiprocesoriniu režimu (taipogi naujumas), atitinkamai bus sukurtas karkasas, kuris yra pamatas kuriant prognozavimo posistemę. Veikimo stabilumo multiprocesorinėje sistemoje netirsime. Karkaso realizavimo aspektus apžvelgsime bendrai.

USEGM projekto uždavinys yra tirti modeliuojamą vartotojų elgseną ir daromus sprendimus. Šio modeliavimo rezultatai yra rinkos tyrimo pagrindinis šaltinis siekiant nustatyti ir išsiaiškinti, kurie padaryti virtualių investuotojų žingsniai yra optimalūs ir atnešė naudos, ir kurie privedė prie bankroto. Virtualių vartotojų apsisprendimą modeliuoja prognozavimas rytojaus dienai [11].

Kuriamos ir analizuojamos posistemės tikslas yra pateikti virtualiam investuotojui kiek galima geresnę prognozę rytojaus dienai, pagal kurią jis galėtų vykdyti atitinkamus veiksmus. Prognozės sudarymui tirti virtualius (tyrime naudosime ir realius duomenis [9]) istorinius duomenų rinkinius ir pagal pasirinktus autoregresinius skaičiavimo modelius sudaryti prognozę sekančios dienos akcijų kainoms nuspėti.

Šio darbo motyvacija yra tobulinti prognozavimo modelių pritaikymą virtualiems modeliams [2, 3], kurių tyrimo rezultatai yra labai priklausomi nuo prognozavimų modelių rezultatų. Prognozių tobulinimas yra esminis motyvas tiek sudarant prognozes virtualiems modeliams, tiek realiems duomenims.

Toliau šiame darbe bus detaliam nagrinėjami pasirinkti realizuoti autoregresiniai modeliai. Aprašomas tinkamumas prognozavimui bei realizavimo ypatumai. Projektinėje dalyje pateiksime detalią modelių realizaciją. Tyrimo dalyje apžvelgsime įgyvendintą projektą, jo trūkumus bei pasiūlymus tobulinimo darbams ateityje. Prognozavimo modelių įgyvendinimas ir tyrimas toliau bus tiriamas USEGM modelio kontekste.

## 1.2 Bendrinės sąvokos

Prieš pradėdant detalią projekto apžvalgą reiktų paminėti kelis esminius skirtumus tarp dažnai darbe naudojamų sąvokų žymėjimo, idant skaitytojui būtų lengviau suvokti, apie ką yra kalbama:

- USEGM modelis yra šiuo metu vystomas modelis (dalinai realizuotas šiame darbe), kurio pirmtakas yra SEGM modelis. Modelio sąvoka apima virtualių akcijų biržos modelį;
- RW, AR, AR-ABS – tai modelių santrumpos, o  $RW$ ,  $AR(p)$ ,  $AR-ABS(p)$  – įgyvendinami prognozavimo modeliai (žymimi kursyvu). Modelio sąvoka apima matematinius prognozavimo modelius;
- Trumpai tariant, (U)SEGM ir  $[RW, AR(p), AR-ABS(p)]$  modeliai yra du skirtingi ir absoliučiai nesusiję sąvokomis objektai;
- Parametras  $t$  yra rytojus (arba prognozė rytojaus dienai). Išraiška  $t - 1$  yra ši diena ir, atitinkamai,  $t - 2$  - vakar diena. Šios išraiškos naudojamas laiko eilutės duomenims nusakyti.

- Kadangi tiriamieji modeliai yra autoregresiniai, naudojamas parametras  $p$  prie modelių  $AR(p)$ ,  $AR - ABS(p)$  reiškia istorinį dienų kiekį, kurį analizuojame. Galime skaityti, kad tai - „atminties“ parametras (angl. *memory*, kitur literatūroje minimas kaip *order*). Į parametą  $p$  taipogi įeina ir  $t$  parametras - ši diena.

## Skyrius 2

# PROJEKTO ANALIZĖ

Šiame skyriuje detaliai analizuojami prognozavimo posistemės realizuojami autoregresiniai modeliai  $RW$ ,  $AR(p)$  bei  $AR - ABS(p)$ . Pateikiami išsamūs autoregresinių modelių aprašai, sprendimo būdai, įvertinami keičiami parametrai, pagrindžiamos įdiegiamos naujovės, aiškinamas netikslumų, rastų apgražos inžinerijos metu, eliminavimas, keliamos naujos problemos bei išdėstomi esminiai panašumai bei skirtumai.

Abstrakčiai nagrinėjamas įgyvendinamo karkaso multiprocessorinei sistemai tyrimas USEGM modeliui. Analizuojami galimi įgyvendinimo būdai naudojant tam tikras, egzistuojančias, priemones, apžvelgiami galimos architektūros aspektai bei pagrindžiamas jų pasirinkimas.

## 2.1 Prognozavimų modelių tyrimas

### 2.1.1 Prognozės sąvoka

Šių dienų aktuali problema nuspėjant finansinius rodiklius bei įvertinant galimus rizikos faktorius yra jų tikslumas. Yra daugelis veiksnių, kurie nulemia neišprognozuojamas reikšmes. Vieni jų jau žinomi, kiti - bandomi įminti. Problema yra išsiaiškinti ir nustatyti galimus faktorius, kurie lemia akcijų kainų neplanuotą pokytį ir jį optimizuoti. Šią problemą ateityje gali išspręsti daugiaagentės sistemos, prisitaikančios prie aplinkos pokyčių [6, 10].

Prognozavimas yra procesas, kuris pateikia išėjimų aibę, kurie gauti įvedus kintamųjų aibę. Šie duomenys paprastai būna istoriniai duomenys (regresija) ir tam tikri nustatyti ar numatyti papildomi kintamieji (angl. *intervention variables*) [12]. Iš esmės, prognozavimas paremtas prielaida, kad ateities įvykiai yra visiškai ar bent dalinai paremti anksčiau stebėtais ar praeities įvykiais. Daroma prielaida, kad praeities dėsningumai (angl. *patterns*) atsikartos ateityje. Praeities ryšiai gali būti atrasti stebėjimų ir tyrimų metu. Pagrindinė prognozavimo idėja – sudaryti įėjimo ir išėjimo duomenis (kuo labiau priartėti prie įėjimo ir išėjimo duomenų paskirstymo) tam, kad būtų atrastos numanomos taisyklės (matematiškai - apskaičiuoti koeficientus), valdančios stebėtus

veiksmus.

Pavyzdžiui, rinkos kainų prognozavimas gali būti nusakytas tokiu būdu. Tarkime,  $u_{t-1}$  nusako šiandieninę kainą,  $v_{t+9}$  nusako kainą, kuri bus po dešimties dienų. Jeigu kainos, kuri bus po dešimties dienų, prognozavimas bus atliekamas remiantis šiandienos rinkos kaina, turi būti funkcinis kelias nuo  $u_{t-1}$  iki  $v_{t-9}$ , kur  $v_{t-9} = G(u_{t-1})$  [6]. Naudojantis istorinių duomenų duomenų reikšmėmis (laiko eilute)  $T_n[u_{t-n}, \dots, u_{t-2}, u_{t-1}]$ , gali būti išvesta bendra funkcija  $G(T_n)$ . Apskritai, funkcijoje  $G(T_n)$  gali būti naudojami  $u_{t-n}$ , kurie turi daugiau informacijos apie esančią kainą (papildomos išskaičiuotų kintamųjų reikšmės).

Teoriškai, neskaitant vien tik prognozės tikslumo akcentavimo, reikėtų atsižvelgti ir į kitus finansinius kriterijus. Investuotojai bei tyrinėtojai linkę naudotis gerumo (angl. *goodness of fit*) ar panašiais kriterijais, kad įvertintų ar treniruotų savo sukurtus modelius finansinėje terpėje. Matematinų skaičiavimų požiūriu šis būdas yra teisingas. Realybėje tobulas prognozavimas yra neįmanomas dėl nuolat kintančių veiksnių, kurių nuspėti praktiškai neįmanoma. Nei vienas modelis negali pasiekti idealių rezultatų. Todėl tobulo prognozavimo pasiekimas, nėra esminis tikslas. Įmanoma tik optimizuoti netobulas prognozes ir naudotis kitais kriterijais, kad būtų gautas kuo realesnis rezultatas [6].

Prieš atliekant istorinių duomenų analizę, turi būti vykdomas atitinkamas duomenų organizavimas ir apdorojimas. Pradinis apdorojimas ir teisingas įvedimo duomenų atrinkimas gali turėti įtakos prognozavimo vykdymui. Rėmimasis jautrumo analize pasirenkant indikatorius kaip įėjimus gali padėti panaikinti nereikšmingus įėjimus (pasikartojančios reikšmės ir kt.).

Prognozavimo posistemės uždavinys yra

## 2.1.2 Prognozavimo posistemės uždaviniai ir tikslai

Prognozavimo posistemės uždaviniai yra:

- Gauti koeficientus  $a_n$  naudojant matematinius modelius  $RW$ ,  $AR(p)$ ,  $AR - ABS(p)$  ir žinomą laiko eilutę. T.y. atlikti modelio pritaikymą realiems ir USEGM generuojamiems duomenims (angl. *fitting*).
- Iš gautų  $a_n$  koeficientų optimizacijos metu sudaryti galimas prognozes rytojaus ar kelių dienų tikėtinioms vertėms gauti.

Prognozavimo posistemės tikslas yra sudaryti prognozes sekančiais dienai. Projekto dalinis tikslas - tobulinti prognozių sudarymo modelius bei mažinti gaunamą paklaidą (arba maksimizuoti vartotojo naudingumo funkciją USEGM kontekste (angl. *utility*)).

Šiame darbe sutelksime dėmesį į matematinius autoregresinius modelius, kuriuos integruosime į kuriamą USEGM modelį. Visi tiriamieji modeliai ( $RW$ ,  $AR(p)$ ,  $AR - ABS(p)$ ) yra autoregresiniai, analizuojantys istorinių duomenų imtis - laiko eilutes. Šie modeliai yra gerai žinomo modelio

- autoregresinio slenkančio vidurkio matematinio prognozavimo modelio trumpiems periodams (ARMA) - atskiri variantai, kurie detalizuoti ir plačiau nagrinėjami šio skyriaus 2.1.3 punkte. Prognozavimus ilgiems periodams atlieka patobulinti ARMA modeliai ARIMA, ARFIMA, kurie taipogi įvertina ir papildomus prognozavimo faktorius [13, 14].

Kuriamai prognozavimo posistemei bus taikomi prognozavimo modeliai trumpam periodui. Taipogi atsisakyta ARMA modelio antros slenkančio vidurkio dalies<sup>1</sup> dėl pernelyg sudėtingo jų įgyvendinimo (žr. plačiau 4 skyrių) esamam modeliui. Šiame darbe USEGM modelio prognozės posistemė apsiribos ARMA modelio pirmąja dalimi<sup>2</sup> ir jos atskirais patobulintais atvejais.

Prognozavimo posistemės naujumo atžvilgiu bus:

- Tobulinami esami prognozavimo modeliai šiuo metu naudojami SEGM. Tobulinimas pasireikš netikslumų šalinimu, kurie buvo pastebėti vykdant tyrimus SEGM modeliu ir programinės įrangos išeities kodo peržiūros metu tiriant ir atliekant apgražos inžineriją (angl. *reverse engineering*);
- Įtraukiamas naujas *AR-ABS(p)* modelis, kuris, teoriškai turėtų pagerinti gaunamus rezultatus. Gautų eksperimentinių rezultatų palyginimui žr. 5 skyrių „Modelių eksperimentinis tyrimas“ 33 puslapyje .

### 2.1.3 ARMA modelis

Apžvelgsime ARMA modelį ir jo naudojamus parametrus [14, 15].

$$w_t = \sum_{i=1}^p a_i w_{t-i} + \sum_{i=1}^q b_i \epsilon_{t-i} + \epsilon_t \quad (2.1)$$

Formulėje (2.1) esančių parametru reikšmės:

$w_t$  – akcijų kursas šiandien;

$w_{t-1}$  – akcijų kursas rytoj (prognozė);

$p$  – dienų kiekis, kiek atsimenam į praeitį (laiko eilutės imtis);

$q$  – skaičiuojamo slenkančio vidurkio dydžio reikšmė;

$\epsilon_t$  – atsitiktinė paklaida rytojaus prognozei, baltasis triukšmas, gaunamas atsitiktine tvarka pagal Gauso (*Gauss*) pasiskirstymą [16];

$a_i, b_i$  – įtakos koeficientai;

<sup>1</sup>Slenkantis vidurkis arba buvusių paklaidų įvertinimas (MA)

<sup>2</sup>Autoregresinė dalis (AR)

$\epsilon_t$  – apskaičiuotas paklaidos dydis (išsamiau apie paklaidų vertinimą žr. 2.1.4 skyrių „Paklaidų įvertinimas“ 16 puslapyje).

Įtakos koeficientus  $(a_i, b_i)$  nustatom optimizuodami paklaidas (2.3):

$$f(x) = \sum_{t=1}^T \epsilon_t^2 \quad (2.2)$$

Čia  $T$  - dabartinis laikas (ši diena). Kiti parametrai aprašomi (2.3):

$$x = (x_1, \dots, x_{p+q}), x = a_i, i = (1, \dots, p), x_i = b_{i-p}, i = (p + 1, \dots, p + q) \quad (2.3)$$

ARMA modelyje prognozuojamos reikšmės gavimą galima būtų padalinti į trijų dalių sumą:

1. Tam tikro kiekio  $p$  prieš tai buvusių laiko eilutės reikšmių  $w_{t-i}$  padaugintų iš optimizavimo metu rastų koeficientų  $a_i$  sumos, vadinamos autoregresijos dalimi AR;
2. Metodo apskaičiuotos  $\epsilon_{t-i}$  paklaidų sandaugos su atitinkamais koeficientais  $b_i$  sumos, vadinamos slenkančio vidurkio dalimi MA ir;
3.  $\epsilon_t$  – nežinomų aplinkos veiksnių, dar vadinamų baltuoju triukšmu, dalies.

Susidaro įspūdis, jog gauti prognozę yra visai paprasta, tereikia turėti duomenų seką t.y. laiko eilutę, tačiau čia svarbu pabrėžti, kad prognozės tikslumas labai priklauso nuo  $a$  ir  $b$  koeficientų arba modelio, kurio pagalba jie bus rasti. Žinoma, nereikia pamiršti, jog svarbi ir pati laiko eilutė: ar tai prognozuojami duomenys, ar tiesiog atsitiktinių skaičių seka. [6].

Koeficientų  $a$  skaičiavimas yra istorinių reikšmių optimizavimas su tikslu rasti koeficientus, pagal kuriuos prognozavimo modelis yra pritaikomas prognozuoti reikšmes. Pritaikymas (angl. *fitting*) gali būti tapatinamas su dirbtinių neuroninių tinklų (ANN) apmokymu: tam tikrų koeficientų įsisavinimu su testine duomenų imtimi (laiko eilutės) ir pagal kitą duomenų imtį sudarinėti prognozes ar vykdyti duomenų paskirstymą (angl. *data-mining*) [17, 18]. Šis paprastas palyginimas supaprastina matematinių modelių supratimą.

Aprašėme „AR + MA +  $\epsilon$ “ modelį (2.1). Toliau detalai nagrinėsime integruojamų į USEGM modelį prognozavimų modelius.

### 2.1.3.1 AR( $p$ ) modelis

ARMA modelio atskiras atvejis, pirmoji dalis – AR. Ji gaunama, kai parenkame parametą  $q = 0$ . Tai reiškia, kad atsisakome buvusių padarytų paklaidų įvertinimo tolimesniuose prognozavimo etapuose (2.4). MA dalis neįeina į USEGM prognozių posistemės realizavimą [3].  $p$  - analizuojamų istorinių reikšmių kiekis. Šis modelis bus tobulinamas šiame darbe.

$$w_t = \sum_{i=1}^p a_i w_{t-i} + \varepsilon_t \quad (2.4)$$

Koeficientų  $a_p$  radimui sprendžiame lygčių sistemą Gauso (Gauss) metodu [19]. Liekanas (arba, vadinkime, - paklaidas) (angl. *residuals*) skaičiuojame pagal (2.5) formulę:

$$\begin{aligned} \varepsilon_1 &= w_1 \\ \varepsilon_2 &= w_1 - a_1 w_1 \\ &\dots\dots\dots \\ \varepsilon_t &= w_t - a_1 w_{t-1} - \dots - a_p w_{t-p} \end{aligned} \quad (2.5)$$

Šio modelio koeficientų radimui, kaip minėjome, liekanų optimizavimui naudojame Gauso metodą. Egzistuoja kitas sprendimo būdas, kai optimizavimą pakeičiame į Burgo (*Burg*) metodu ir koeficientų ieškome maksimalios entropijos būdu (angl. *Max Entropy*), tad kitas AR variantas gali būti žymimas kaip AR-ME [12]. Šio metodo nenagrinėsime ir nerealizuosime, kadangi tai nėra projekto užduotis ir, galima sakyti, kad tai - ateities prognozavimo posistemės tobulinimo klausimas.

Reiktų paminėti, kad šis modelis yra sprendžiamas mažiausių kvadratų (angl. *Least Squares*) metodu, tad yra labai jautrus didesniems duomenų nukrypimams. Standartinio AR modelio aprašymas kai kur sutinkamas kaip AR-LS.

**Modelio problemos.** Šis metodas buvo įtrauktas į SEGM modelį, tačiau tyrimo metu pastaruoju modeliu pastebėti darbo netikslumai [3]. Dėl šių priežasčių šis prognozavimo modelis yra tobulinamas USEGM modeliui.

Kadangi (U)SEGM modelyje manipuliavimas rinka prasidėdavo nuo  $t$  dienos po akcijų emisijos į rinką, susidarydavo situaciją, kai  $AR(p)$  modeliui pritrūkdavo laiko eilutės istorinių duomenų atlikti analizę ir rasti tinkamus  $a_i$  koeficientus, ko pasekoje tyrimo rezultatai SEGM modeliui skaičiuodavo klaidingus rezultatus [2]. Čia emisiją reiktų suprasti kaip naujos įmonės vertybinių popierių išleidimą į rinką, kurios akcijų prieš tai rinkoje nebuvo (t.y. nesutapatinti su realios rinkos reiškiniais, papildomoms emisijom ir kt. terminais). Taigi, šiuo atveju, virtualios biržos modeliu, laiko eilutėje yra tiek kainų reikšmių, kiek dienų prabėgo nuo emisijos. (U)SEGM modelyje dienų kaitą vadinsime iteracijomis, kadangi visi procesai yra modeliuojami virtualiu laiku.

Paimkime pavyzdį. Laiko eilutėje yra  $n = 2$  reikšmės: atitinkamai, jos yra pradinė akcijų kaina<sup>3</sup> ir pirmos atidarymo dienos  $t - 1$  kaina. Nustatėme  $p = 5$ , t.y.  $a$  koeficientų sudarymui analizuosime istorinių duomenų kiekį, kuris yra laiko eilutės 5 reikšmės. Iš (2.4) formulės matome, kad modeliui įvertinti  $p = 5$  reikšmes reikia laiko eilutės  $p + 1$  duomenų kiekio. Iš čia išplaukia,

<sup>3</sup>Pradinę akcijų kainą nustato makleris (angl. *broker*)



kad turime  $p + 1 - n = 4$  verčių laiko eilutėje trūkumą. Klausimas - ką daryti, kad veiktų  $AR(p)$  modelis ?

Prieš pradėdant skaičiuoti  $a$  koeficientus, SEGM modelyje yra įgyvendintas toks algoritmas: esamų istorinių reikšmių kiekis yra  $n$ , tuomet jei  $n \leq p$ , tada  $p = n - 1$ . Tai reiškia, kad kiekvienoje iteracijoje, kai  $p + 1 - n > 0$ , imamas skirtingas nagrinėjamų duomenų kiekis, t.y. kaskart keičiamas  $p$  parametras, kuris kiekvienoje iteracijoje apskaičiuoja vis skirtingus parametrus  $a$ , ko pasekoje tyrimas neduoda gerų rezultatų USEGM modeliu [6, 2]. Teoriškai toks algoritmas galimas, tačiau praktiškai jis neveikia, todėl nuspręsta jį tobulinti.

Reiktų pastebėti, jog realioje rinkoje su realiomis kainų kvotomis (angl. *quotes*) toks reiškiny yra ypatingai retas, kadangi laiko eilučių dydžiai yra daug didesni, nei tiriamų istorinių duomenų kiekis  $p$ . Be to, finansinių reikšmių prognozavimas nepasižymi pastovumu ir geriausia prognozė rytojui yra ši diena [20], ką ir parodys tyrimų rezultatai, tad modeliuojant rinką su  $p > 10$  reikšmių tampa beprasmiška. Detaliau apie gautus rezultatus žiūrėti 5 skyrių „Modelių eksperimentinis tyrimas“ 33 puslapyje. Tačiau virtualioje rinkoje, priklausomai nuo  $AR(p)$  pasirinkto parametro  $p$ , kartais pritrūksta ir 10 iteracijų, dėl kurių tolimesni tyrimo rezultatai SEGM tampa ne itin geri.

**Siūlomas patobulinimas.** Siūlomas euristinis  $AR(p)$  modelio patobulinimas, kuris remiasi trūkstamų reikšmių generavimu atbuline eiga (generuojame reikšmes į praeitį). Siūlomas prof. habil. dr. J. Mockaus metodas, kuris remiasi tokiu labai paprastu algoritmu. Jei trūksta istorinių reikšmių, kurių kiekis yra  $p + 1 - n$ , tuomet tokį kiekį iteracijų simuliuojam prognozavimą pakeisdami neturimų istorinių duomenų kiekį  $p + 1$  tokia laiko eilute: 10; -10; 20; -20; ...; 100; -100. Gautų koeficientų pirmoji reikšmė  $a[0]$  turės reikšmę 0.1010{10}. Šie parametrai prognozei turės nedidelę įtaką.

Kitas euristinis patobulinimas, kurio autorius yra šio dokumento autorius, yra įgyvendintas šiame modelyje. Trūkstamą istorinių reikšmių kiekį kompensuojame  $n$  istorinių reikšmių sumos vidurkiu  $nVid$ , prie kurio pridėdame atsitiktinį triukšmą  $\varepsilon$ . Sugeneruojame reikšmes atbuline tvarka pradėdami nuo trūkstamos reikšmės. Šis algoritmas taipogi pateisina  $AR(p)$  modelio ankstesnio įgyvendinimo SEGM modelyje trūkumo patobulinimą (plačiau skaityti 5 skyrių „Modelių eksperimentinis tyrimas“ 33 puslapyje).

Kitas galimas metodas remtųsi standartinio nuokrypio skaičiavimu, kaip papildomu parametru  $d$ , nusakančiu generuojamo triukšmo  $\varepsilon$  magnitudę (dydį), dar žinomą kaip nepastovumo parametru (angl. *volatile*). Taigi, teoriškai, prie  $nVid$  reikšmės pridėjus triukšmą  $\varepsilon$ , kurio dydis priklauso nuo žinomų reikšmių išsibarstymo  $d$ , teoriškai turėtume gauti dar geresnį variantą. Bet tai - ateities tyrimo objektas.

### 2.1.3.2 AR-ABS(p) modelis

AR – ABS(p) modelis taipogi yra ARMA modelio atskiras atvejis - modifikacija. Tai - 2.1.3.1 skyrelyje aptartas AR(p) atvejis, tik  $a$  (žr. formulę 2.1) koeficientų radimui naudojamas ne mažiausių kvadratų optimizacijos metodas, kurį naudoja AR(p) modelis, bet optimizavimas absoliučiomis koeficientų reikšmėmis.

Kaip buvo minėta, mažiausių kvadratų metodas AR(p) yra labai jautrus dideliems nukrypimams (duomenų imtis turi didelę deviaciją). Pavyzdžiui, optimizuojant kvadratinis nuokrypius, didelis nuokrypis - 100, turi tokią pat įtaką, kaip ir 10'000 mažų lygių vienetui nuokrypių. Todėl kilo idėja mažiausių kvadratinių nuokrypių optimizavimą pakeisti nuokrypių absoliutiniais didumais optimizavimu [14, 6]. Prognozės sudarymui modelis reikalauja, kad  $n > p$ , kur  $n$  - įvedamų laiko eilučių reikšmių kiekis, antraip  $a$  koeficientai yra lygūs 0 ir prognozės negauname. Šiuo atveju taikomas toks pat euristinis algoritmas esamai situacijai apeiti: generuojami duomenys į praeitį. Plačiau apie šį algoritmą skaitykite 2.1.3.1 skyriaus „Siūlomas patobulinimas“ poskyryje.

AR – ABS(p) paklaidų optimizavimas vykdomas pagal (2.6) formulę:

$$f(x) = \sum_{i=1}^T |\epsilon_i| \quad (2.6)$$

Čia  $T$  - dabartinis laikas (šiandien), o  $x = (a_i, i = 1...p)$ .

Kaip matome iš (2.3) ir (2.6) formulių, pastaroji skiriasi tik tuo, kad liekanas optimizuojam naudodami modulį taip gaudami absoliučiuosius dydžius. Teoriškai, naudojant šį modelį, investuotojo elgesį geriau atspindi būtent šis modelis [21]. Eksperimentų metu patikrinsime ir pamėginsime įrodyti praktiškai, ar taip yra iš tikrųjų (plačiau žiūrėkite 5 skyrių „Modelių eksperimentinis tyrimas“ 33 puslapyje). Šis AR – ABS(p) nebuvo įtrauktas į SEGM modelį ir yra naujovė USEGM modelio aspektu.

### 2.1.3.3 RW (Wiener) modelis

RW modelis, dar žinomas kaip Vynerio (*Wiener*) procesas, taipogi yra autoregresinis modelis, išvestinis ARMA modelio atvejis. Tiksliau sakant, tai yra dar labiau supaprastintas modelis nei AR(p). Šio modelio parametrai atitinkamai yra  $p = 1$ ,  $a = 1$ . Taigi, naudojantis ARMA modelio formule (2.1) galime aprašyti RW matematinį modelį (2.7):

$$w_t = w_{t-1} + \epsilon_t \quad (2.7)$$

RW modelis teigia, kad geriausias sprendimas (prognozė) rytojui yra šios dienos kaina. Neturi reikšmės, kaip pakliuvom į esamą padėtį. Neturi reikšmės, kokios reikšmės buvo anksčiau [20]. Ir tai pasitvirtina praktiškai, kaip aiškiai matome iš gautų tyrimo rezultatų. Tai paprasčiausias ir mažiausiai analizės reikalaujantis metodas, kurio reikšmės grafiškai pasislenka tik per vieną

laiko reikšmę į dešinę pusę [22]. Plačiau žiūrėkite 5 skyrių „Modelių eksperimentinis tyrimas“ 33 puslapyje.

Dar daugiau, šį modelį galima traktuoti kaip paprasčiausią Markovo grandinės (angl. *Markov Chains*) modelio atvejį, kuris teigia, kad praeitis yra nereikšminga numatant ateitį ir svarbi yra tiksliai esamojo laiko informacija. Būtent šią sąvoką ir realizuoja *RW* modelis. [18].

**Modelio problemos.** SEGM realizacijoje atliekant apgražos inžineriją buvo pastebėta, kad modeliui yra paduodamos netinkamos laiko eilutės reikšmės. Buvo įvelta standartinė, vadinamoji „ $\pm 1$ “ (indekso), klaida, dėl kurios *RW* modelio prognozavimas būdavo sudaromas principu „rytojaus kaina - kaip užvakar“, t.y.  $t$  dienos prognozė sudaroma naudojant  $t-2$  istorinę reikšmę [6]. Galima teigti, kad tai atsitiko vykdant programos tobulinimo darbus, kadangi prie SEGM modelio dirbo net 8 autoriai [2].

**Siūlomas patobulinimas.** Netikslumas pašalintas realizavus USEGM modelio prognozavimo posistemę. Modelis dirba tvarkingai:  $t$  kainos prognozė yra  $t-1$  dienos istoriniai duomenys plus triukšmas  $\varepsilon$ . Eksperimentinius tyrimų rezultatus žiūrėkite 5 skyriuje.

## 2.1.4 Paklaidų įvertinimas

### 2.1.4.1 Paklaidų skaičiavimo metodai

Paklaidų įvertinimas padeda nustatyti skirtuminį dydį tarp realaus (turimo) rezultato bei gaunamo (atlikus prognozę). Terminą „paklaida“ kai kur literatūroje sutinkame kaip tam tikrą padarytą klaidą<sup>4</sup>. Paklaidos sąvoką labiau atspindi darbo su prognozėmis kontekstą. Šiame poskyryje apžvelgsime būtent prognozei taikomus ir plačiai naudojamus du paklaidos nustatymo metodus:

- Vidutinė kvadratinės šaknies paklaida (RMS(E)), dar žinoma kaip vidutinis kvadratinės šaknies nuokrypis (RMSD) [23, 24]
- Absoliutinė paklaida, kurią toliau vadinsime AME santrumpa (angl. *Absolute Mean Error*) [25].

Paklaidos esmė yra įvertinti sudaromų prognozavimų modelių efektyvumą (angl. *efficiency*) bei tikslumą prognozei. Šiuos metodus naudosime eksperimentinio tyrimo metu<sup>5</sup> siekiant įvertinti ir palyginti tiriamų *RW*, *AR(p)* bei *AR - ABS(p)* modelių charakteristikas.

Anot prof. habil. dr. Jono Mockaus, RMSE (kvadratų metodai) dažniausiai naudojami teoretikų ir atspindi „turtingo“ investuotojo elgseną, kas šiame darbe turi tokią prasmę: prie tam tikros numatytos prognozuojamos dydžio paklaidos investuotojas drąsiai investuoja nebijodamas

---

<sup>4</sup>Dėl aiškumo, visur šiame darbe naudosime terminą „paklaida“.

<sup>5</sup>Žr. 5 skyrių „Modelių eksperimentinis tyrimas“ 33 puslapyje

prarasti turtą (angl. *asset*). [21]. Šis dydis labiau naudojamas teoretikų. Tuo tarpu AME paklaida modeliuoja „bailaus“ investuotojo elgseną, kuris investuoja esant tik labai mažam paklaidos dydžiui. Atliktuose eksperimentiniuose tyrimuose ši hipotezė pasitvirtina<sup>6</sup> ir iš čia išplaukia, kad  $AR(p)$  modelis daugiau atspindi „turtingo“ investuotojo strategiją, o  $AR - ABS(p)$  - „bailaus“, atsargaus.

Apžvelgsime paklaidų skaičiavimo ypatumus.

**Vidutinė kvadratinės šaknies paklaida  $RMS(E)$ ,  $RMSD$ .** RMSE yra dažnai naudojamas palyginti vertes, kurias prognozavo modelis, su tomis, kuriomis naudotasi tą modelį sukurti. Dar žinoma kaip standartinė paklaida (angl. *Standard Error*). Kai kur RMSD naudojamas palyginti du skirtingus objektus, kurie abu skiriasi nuo tam tikro etaloninio dydžio [6]. Šią paklaidą skaičiuojame taip:

1. Pirmiausiai apskaičiuojama MSE paklaida (2.8);

(a) MSE gali būti skaičiuojama kaip variacijos ir paklaidos kvadrato suma (2.9);

2. RMSE (RMSD) skaičiuojame pagal (2.10) formulę.

$$MSE(\hat{\theta}) = E((\hat{\theta} - \theta)^2) \quad (2.8)$$

$$MSE(\hat{\theta}) = Var(\hat{\theta}) + (Bias(\hat{\theta}, \theta))^2 \quad (2.9)$$

$$RMSE(\hat{\theta}) = RMSD(\hat{\theta}) = \sqrt{MSE(\hat{\theta})} = \sqrt{E((\hat{\theta} - \theta)^2)} \quad (2.10)$$

Čia  $\hat{\theta}$  yra skaičiuotojas (angl. *estimator*), šiame darbo kontekste prognozuojama reikšmė, o  $\theta$  - realioji (modeliuota) laiko eilutės vertė.

Skaitome, kad paklaida yra maža ir tenkina , kai jos dydis neviršija  $0,01$  numatytos reikšmės. Šis dydis naudojamas daugelyje šaltinių, eksperimentuojančių su šio tipo paklaidomis [7, 10]. Eksperimentų metu remsimės šiuo dydžiu kaip labai maža paklaida, kuri reiškia, kad dydis tarp išmatuotų verčių yra itin mažas ir vertės beveik sutampa.

**Vidutinė absoliučioji paklaida  $AME$ .** Absoliučioji paklaida - tai paprasčiausias tiesinės funkcijos pavyzdys. Visuminiai paklaidai įvertinti įskaitant visus laiko eilutės duomenis bus naudojama suvidurkinta absoliučiosios paklaidos reikšmė - AME.

Ši paklaida skaičiuojama pagal formulę (2.11):

$$AME(\hat{\theta}) = |\hat{\theta} - \theta| \quad (2.11)$$

<sup>6</sup>Žr. 5 skyrių „Modelių eksperimentinis tyrimas“ 33 puslapyje

Nors šios paklaidos skaičiavimas yra labai paprastas, tačiau ji parodo teisingiausią skirtumą tarp skaičiuojamų reikšmių ir praktikoje dažniausiai pasitvirtina turint omeny „bailaus“ vartotojo strategiją.

## 2.2 USEGM karkasas

Dėl vykdomų sudėtingų algoritimų ir skaičiavimų USEGM modelio posistemėse, kurie trunka ilgą laiko tarpą (žr. iliustraciją 5.13 puslapyje 47), buvo numatyta modelį realizuoti multiprocesorinės sistemos pagrindu. Kadangi technologijos žengia sparčiu žingsniu pirmyn, o SEGM modelis vis dar tebedirba naudodamas tik vieną procesoriaus (CPU) branduolį multiprocesorinėse sistemose (angl. *multi-core*), buvo nuspręsta tobulinti modelį, kuris išnaudotų visus sistemos resursus.

Šios analizės uždavinys yra parinkti tinkamą karkasą paralelinių procesų vykdymui remiantis ištirto SEGM modelio veikimu ir USEGM specifikacijoje nurodomais patobulinimais.

### 2.2.1 Dalykinės srities analizė

Buvo pilnai ištirtas SEGM modelis naudojant apgražos inžineriją ir jame vykstantys procesai [2]. Taipogi išnagrinėti USEGM modelio specifikacijoje [3] nurodyti ateityje diegiamų naujovių algoritminiai principai. Pastebėti du esminiai kriterijai, pagal kuriuos nustatyti pagrindiniai modelio veikėjai (angl. *entities*):

- Virtualūs investuotojai  $I$ ;
- Akcijas platinanti įmonė  $J$  (angl. *stock*), turinti jai būdingą žymę-pavadinimą (angl. *tick*);
- Kiti procesai  $K$ : grafikų perpiešimas, duomenų atnaujinimas, įvykių klausytojai (angl. *event listeners*) ir t.t.

$I$  bei  $J$  kiekis nėra apibrėžtas. Jų yra  $M$  ir  $N$  atitinkamai. Modeliuojant rinką,  $I$  investuotojų perka / parduoda  $J$  akcijų kiekį. Žinome, kad  $I$  turimas biudžetas nėra paskirstytas ir visos perkamų / parduodamų  $J_i$  akcijų kainos išskaičiuojamos ar priskaičiuojamos gautos pajamos į tą patį biudžetą.

### 2.2.2 Siūloma įgyvendinimo logika

Turimų multiprocesorinės sistemos resursų dalinimu apsiima semaforai (angl. *semaphore*). Gijos (angl. *threads*), gavusios leidimą naudoti resursą, vykdo procesą - vienos dienos prognozių sudarymo bei akcijų pirkimų / pardavimų logiką. Resursas - tai vienas CPU branduolys, kuriam priskiriam vieną giją. Iš tiesų tai abstrakti sąvoka, kadangi  $K$  procesai paraleliai naudoja tą patį resursą kitiems sisteminiams procesams vykdyti. Tačiau kol kas apie tai pamirškime.

Taigi, viena gija naudoja vieną resursą. Gijai priskiriamas  $J_i$  procesas, kuris manipuliuoja  $I_m$  klientais. Vykdoma viena iteracija, per kurią yra įvykdomi  $J_n$  procesai. Resursai gijoms išduodami atsitiktine tvarka, kas randomizuoja  $J_n$  procesų eigą ir patobulina virtualų rinkos modelio realumą. Reiktų pastebėti, kad ši randomizacija nėra ištirta ir vėlesnėse karkaso tobulinimo stadijose reiktų ištirti šios dalies veikimą ir dėsningumus. Remiantis teoriniais skaičiavimais, sistemos efektyvusis resursų panaudojimas (angl. *payload*) būtų apie 85% naudojant šį modelį. Efektyvusis panaudojimas - tai sąvoka, nusakanti iteracijos apdorojimą be papildomų vėlinimų ar laukimų (angl. *busy, waiting*).

Atvirkštinė logika, kai  $I_i$  manipuluotų  $J_n$  rinkomis taipogi galima, tačiau sistemos resursų efektyvusis panaudojimas būtų tik apie 60-70% dėl to, kadangi pagrindinius skaičiavimus visgi atlieka ne kliento procesas  $I$ , o akcijų procesas  $J$ . Šiame kontekste kliento prognozavimas yra tik sąvoka, tačiau sistemiškai prognozes apskaičiuoja  $J_i$  procesas kiekvienam klientui.

Po vieno iteracijos ciklo resursus blokuoja  $K$  procesas ir atnaušina duomenis: perpiešia grafikus, skaito / rašo į duombazę rezultatus ir t.t. Šis veikimas yra ateities klausimas ir nėra realizuotas.

### 2.2.3 Kritinių sekcijų (CS) nustatymas

Į klausimą dėl ko numatyta realizacija negali pasiekti 100% resursų panaudojimo atsako susidariusios kritinės sekcijos CS. CS optimizavimas yra vienas iš būdų pagreitinti operacijų vykdymą ir optimaliai panaudoti turimus resursus bei sumažinti laiko sąnaudas. Šiame karkase yra numatytas optimaliausias variantas, tačiau tolimesnio projekto tobulinimo metu gali būti atrastas dar geresnis sprendimas.

Numatyta pagrindinė CS, kuri susidaro  $I_i$  investuotojui perkant ar parduodant akcijas. Kadangi investuotojais manipuliuoja  $J_j$  procesais ir tokių procesų veikia, sakykim, 4 (4-ių branduolių CPU), tokiu atveju, akcijų pirkimų / pardavimų metu gali susidaryti CS, kai  $J_{[1,...4]}$  procesai manipuluos tuo pačiu  $I_i$  investuotojo biudžetu lygiagrečiai. Šiam įvykiui pasireikšti tikimybė yra itin didelė. Šios CS optimizavimas - tai ateities klausimas. Šioje projekto realizacijoje nėra įgyvendintas, tačiau karkase yra numatyta tokia vieta ir priimti atitinkami architektūriniai sprendimai.

Apibendrinant reiktų pabrėžti, kad CS esanti gija priverčia laukti kitas gijas tol, kol yra atliekami tam tikri skaičiavimai. Turint omenyje, kad akcijų pirkimo / pardavimų logika remiasi sudėtingu algoritmu, darome išvadą, kad efektyvusis resursų panaudojimas, kaip buvo minėta, bus apie 85%. Tai yra geras rezultatas. Tobulumo sieksime tolesniuose projekto etapuose ateityje.

### 2.2.4 Paralelinio programavimo karkasas

Iš esamų Java kalba įgyvendintų paralelinių modelių karkasų buvo pasirinktas standartinis *Java Concurrency* karkasas [26]. Šis pasirinkimas rėmėsi šiais kriterijais:

- **Java Concurrency** karkasas - stabilus ir greičiausiai veikiantis karkasas. Įgyvendinama SMA (angl. *Shared-Memory Architecture*) architektūra.

- **OpenMP (JOMP)** siūlo tokius pačius įrankius gijų valdymui, kaip ir Java karkasas, beveik vienodą operacijų vykdymo greitį (operacijos šiek tiek vykdomos ilgiau) bei papildomą priklausomybę nuo kompiliatoriaus bei papildomų sintaksinių deklaracijų. Tinkamas spręsti srities uždaviniams, tačiau su bendros paskirties naudojamom aplikacijom suderinamas sunkiai dėl kompiliatorių versijų nesuderinamumo. Tai yra viena iš pagrindinių neigiamų savybių [27];
- **MPI** nėra tinkamas karkasas, kadangi taikomas daugiau paskirstytom sistemom ir iš esmės pakeičia storojo kliento architektūros koncepciją [28].

Java Concurrency karkaso vardinė sritis yra *java.util.concurrent* .

# Skyrius 3

## PROJEKVINĖ DALIS

Pastaruoju metu rinkos tapo labiau prieinama investavimo priemonė ne tik strateginiams investuotojams, bet eiliniams žmonėms. Taigi rinka yra ne tik susijusi su makroekonominiais veiksniais, bet jie tiesioginiu būdu įtakoja kasdienį gyvenimą. Tai yra mechanizmas, kuris lemia svarbų tiesioginį socialinį poveikį. Skirtingi asmenys žvelgia į šią problemą savaip[6]. Svarbus interesų skirtumų šaltinis yra skirtingas požiūris į riziką:

- Priešiški rizikai;
- Neutralūs rizikai;
- Linkę rizikuoti.

Apžvelgsime esamą įgyvendintą SEGM modelį ir priemones, naudojamas moksliniams tyrimams.

### 3.1 Biržos žaidimo modelis

Biržos žaidimo modelis SEGM (angl. *Stock Exchange Game Model*) yra įgyvendintas Java kalba, skelbiamas atviroje svetainėje (žr. 57 priedą) ir gali būti paleistas naudojant bet kokią naršyklę, kuri turi Java palaikymą.

SEGM modelis modeliuoja  $N$  didelių turto turėtojų elgesį, kai pirkimo / pardavimo strategijos yra apibrėžtos naudojant autoregresijos AR, arba atsitiktinės eigos RW modelių prognozes. AR-ABS modelis skiriasi nuo tradicinių AR tuo, kad optimizuoja absoliučius nuokrypius vietoje mažiausių kvadratų nuokrypių. Vertybinių popierių rinkos prognozavimo užduotis skirsto mokslininkus į dvi grupes:

- Pirmoji grupė yra tie, kurie yra įsitikinę, kad mes galime sukurti mechanizmus prognozuoti rinką pagal statistinę ir fundamentalią analizę. Statistinės analizės pavyzdys yra slenkančio vidurkio autoregresija ARMA (angl. *Autoregressive Moving Average*) skirtas trumpalaikiam



prognozavimui, o dalinės integracijos (ARFIMA) modelis vykdo ilgo laikotarpio prognozes. Fundamentalieji analitikai tiria vidinės vertės turtą ir investuoja į jį, jei jie mano, kad jo dabartinė vertė yra mažesnė už jo tikrąją vertę.

- Antroji grupė mokslininkų pripažįsta efektyvios rinkos hipotezę (EMH). Jie mano, kad rinka yra veiksminga ir, kai paaiškėja nauja informacija, rinkos absorbuoja ją koreguojant save.

Prognozuoti naudojamas RW modelis. Matematinė priemonė įgyvendinant RW yra Vynerio (*Wiener*) modelis, kuris reiškia, kad geriausia prognozė rytojui yra šiandienos vertė. Birža laikoma kaip žaidimas, kuriame konkuruojantys žaidėjai bando padidinti laimėjimus ir perka turtą mažomis kainomis, o pardavinėja didesnėmis kainomis. Pardavimo / pirkimo lygiai grindžiami prognozėmis. Taigi, žaidėjai gauna pelną, jei jie numatė teisingai (prognozuojama vertė pasitvirtino).

Nešo pusiausvyra (angl. *Nash Equilibrium*, NE) pasiekama, jei visų pagrindinių žaidėjų prognozuojama turto kaina pagal Vynerio modelį. Tai reiškia, kad žaidėjas negali gauti daugiau tikėtino pelno naudojant autoregresijos modelio prognozes, kai visi pagrindiniai žaidėjai naudoja Vynerio modelį.

AR-ABS modelis taikomas laiko eilutėms prognozuoti optimizuojant vidutinę absoliučią paklaidą (AME). Absoliutaus nuokrypio optimizavimas išplaukia iš prielaidos, kad vartotojo naudingumo funkcija (angl. *utility*) yra tiesinė. Todėl AR-ABS modelis atspindi neutralų rizikai elgesį (o tai ir yra pagrindinė priežastis naudoti AME). Papildoma priežastis yra kai kurios palankios AME įverčių statistinės savybės.

Dirbant su faktiniais duomenimis vidutinės kvadratinės paklaidos (MSE) ir vidutinės absoliučios paklaidos (AME) apskaičiavimai buvo taikomi atstovauti skirtingų vartotojų požiūriui. Mažai tikėtina, kad EMH užtikrina tikslų aprašymą, kaip elgiasi akcijų rinkos kainos. SEGM modelis gali būti naudingas, net jei jis tiksliai ir neatitinka faktų. Taigi nors ir kainų pokyčiai gali būti ne tik nepriklausomi, modelis yra pradinės aproksimacijos pavyzdys ir gali būti pagerintas gaunant papildomos informacijos apie pagrindines vertybes ir sekant kai kurių modelių akcijų kurso pokyčius.

## 3.2 Nešo pusiausvyra

Mokslinis šio modelio tikslas yra patikrinti hipotezę, kad biržą galima apytikriai apibūdinti kaip kelių žaidėjų žaidimą, naudojant strategijas, grindžiamas Nešo pusiausvyra. Sumodeliuotos vertybių kainos yra žaidimo rezultatas, susidedantis iš pagrindinių turto turėtojų ir atsitiktinio kintamojo, atstovaujamojo didelio kiekio mažųjų. Investiciniai sprendimai priklausys nuo žaidėjo prognozės dėl būsimos turto kainos ir laukiamų dividendų. SEGM modelio tikslas ištirti, kas gali būti išmokta apie žaidimo pagrindus ir rinkos teorijas su šiuo paprastu biržos modelio pavyzdžiu. Daroma prielaida, kad kiekvienas žaidėjas prognozuoja turto kainą pagal autoregresijos modelį.

AR modelio parametrai apskaičiuojami standartiniu mažiausių kvadratų algoritmu naudojant skirtingas  $p$  reikšmes. Metodas yra taikomas siekiant įvertinti parametrus  $p$ , kurie yra gaunami optimizuojant vidutinį nuokrypį nuo Nešo pusiausvyros sąlygos. Pusiausvyra dažnai pasiekama, kai visi pagrindiniai žaidėjai prognozuoja turto kainas Vynerio modeliu, kur  $p = 1$ , ir  $a = 1$ . Tai reiškia, kad vidutinis pelnas negali padidėti naudojant AR modelius, su didesniais atminties parametrais  $p > 1$ . Tai yra naujas paaiškinimas, kodėl RW modelis prognozuoja "rytoj, kaip šiandien".

Kaip jau minėjome, SEGM modelis yra įgyvendintas Java kalba ir yra prieinamas internete. Jokių specialių įgūdžių nereikia dirbant ir pritaikant modelį. Todėl šis modelis yra patogus įrankis modeliuoti ir tyrinėti elgesį biržoje kaip  $N$  asmenų žaidimą. Be to, visus rezultatus galima savarankiškai išbandyti ir palyginti su kitais modeliais. Programa įgyvendina biržos modelį su  $I \leq 8$  stambių klientų ir  $J \leq 8$  akcinių bendrovių skaičiumi. 3.1 pavyzdyje pateikiamas 4-ių pagrindinių žaidėjų pradinių parametru nustatymas.

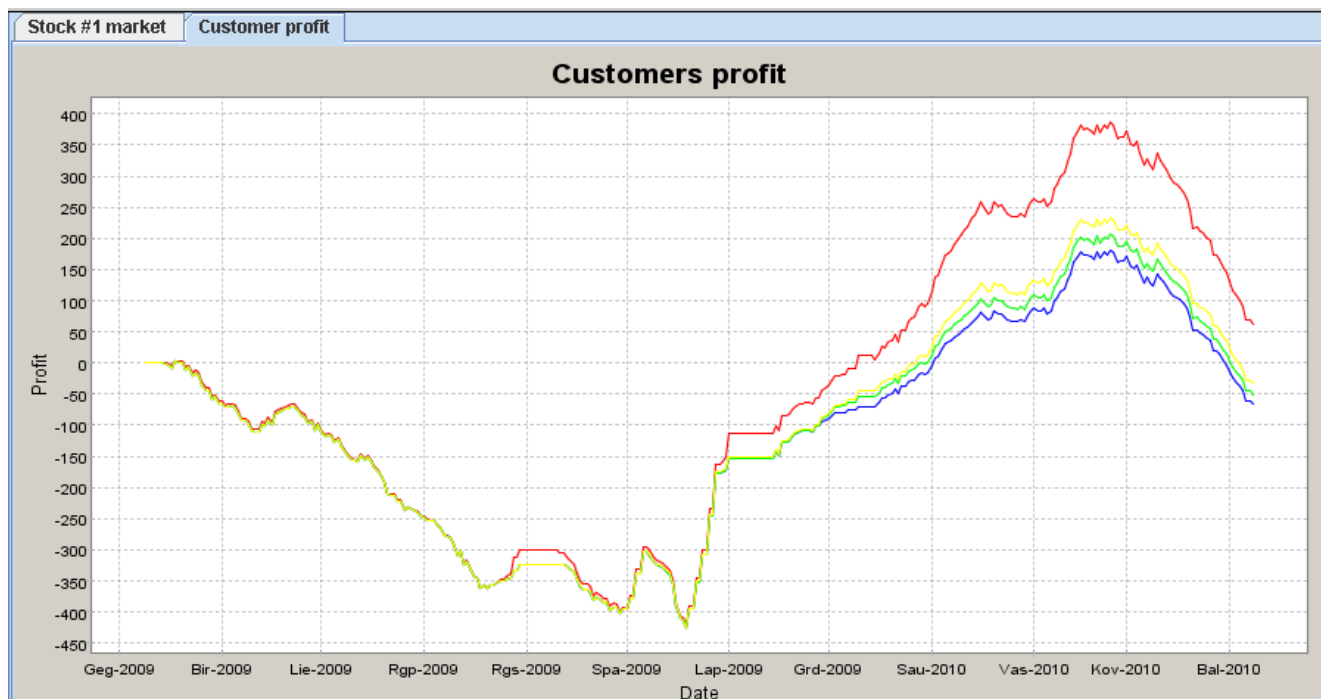
Customer #0	Customer #1	Customer #2	Customer #3
Method: WIENER	Method: AR1	Method: AR2	Method: AR3
Has stock #1: 0	Has stock #1: 0	Has stock #1: 0	Has stock #1: 0
Has stock #2: 0	Has stock #2: 0	Has stock #2: 0	Has stock #2: 0
Has stock #3: 0	Has stock #3: 0	Has stock #3: 0	Has stock #3: 0
Has stock #4: 0	Has stock #4: 0	Has stock #4: 0	Has stock #4: 0
Has stock #5: 0	Has stock #5: 0	Has stock #5: 0	Has stock #5: 0
Kbuy2: 0.08	Kbuy2: 0.08	Kbuy2: 0.08	Kbuy2: 0.08
Kbuy1: 0.05	Kbuy1: 0.05	Kbuy1: 0.05	Kbuy1: 0.05
Kbuy: 0.025	Kbuy: 0.025	Kbuy: 0.025	Kbuy: 0.025
Ksell: -0.025	Ksell: -0.025	Ksell: -0.025	Ksell: -0.025
Ksell1: -0.05	Ksell1: -0.05	Ksell1: -0.05	Ksell1: -0.05
Ksell2: -0.08	Ksell2: -0.08	Ksell2: -0.08	Ksell2: -0.08

3.1 pav.: Pradiniai SEGM nustatymo parametrai

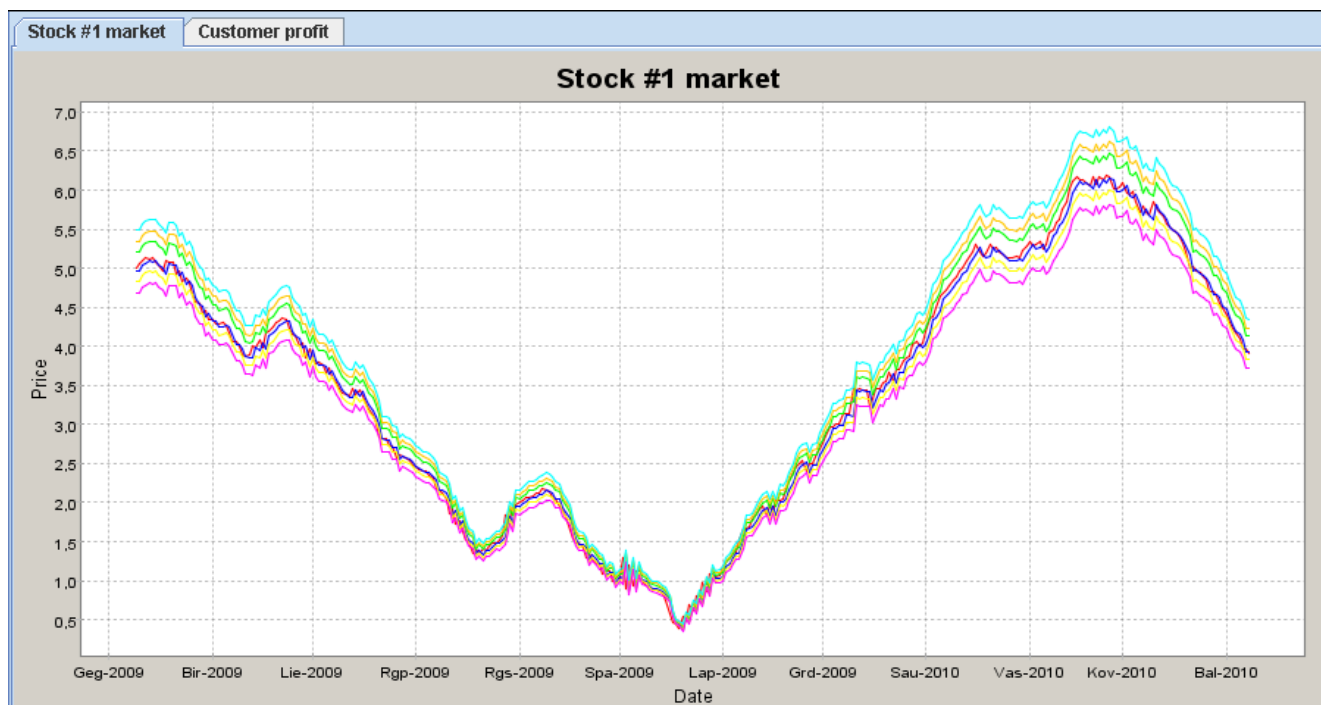
3.2 pavyzdyje matomas skirtingų žaidėjų pelnas sumodeliuotame žaidime, kai pirmasis žaidėjas numato turto kainas remdamasis Vynerio modeliu, o likę žaidėjai prognozuoja naudodami AR modelį.

Raudona linija rodo pirmojo žaidėjo pelną. Kitos linijos rodo likusių dalyvių pelną. 3.3 pavyzdys rodo eilutes, kurios rodo sumodeliuotas turto kainas ir skirtingų veikėjų nustatytus pirkimo / pardavimo apribojimus - maržą (SEGM modelyje ji nėra apskaičiuojama dinamiškai).

### 3.2. Nešo pusiausvyra



3.2 pav.: Skirtingų žaidėjų modeliuojamas pelnas SEGM modelyje



3.3 pav.: Modeliuojamos turto kainos ir maržų (slenksčių) apribojimai SEGM modelyje

### 3.3 Kuriamas USEGM modelis

USEGM modelis yra analogas SEGM modelio, kuria yra tobulinama prognozavimo posistemė, kečiamas karkasas, naudosiantis visus multiprocesorinės sistemos resursus bei įgyvendinsiantis patobulinimus virtualios akcijos biržos modeliavime.

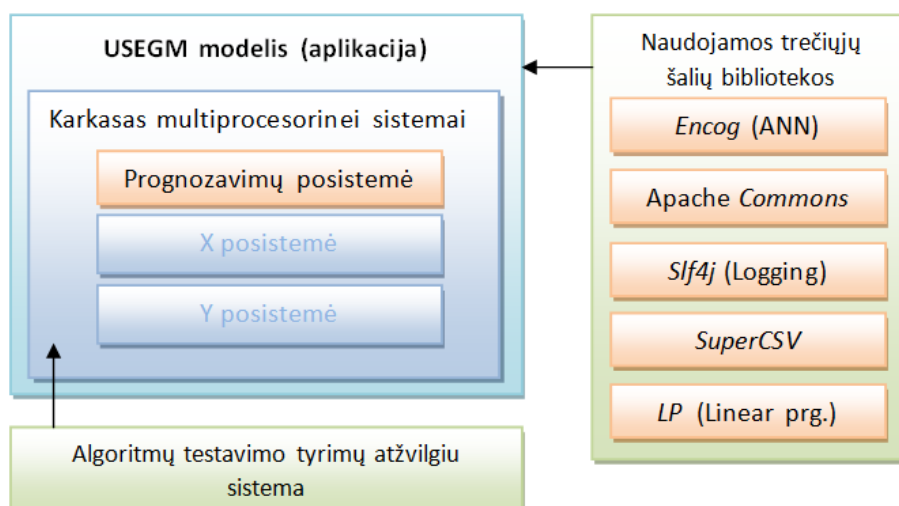
#### 3.3.1 Pagrindiniai naujumo aspektai

USEGM modelis įtraukia žemiau išvardintus naujumus [3]:

- Į prognozavimo posistemę įtraukiamas  $AR - ABS(p)$  modelis;
- Modelis turi dirbti su  $I_m$  klientais ir  $J_n$  akcijų biržomis, kurių procesų vykdymas būtų paskirstytas multiprocesorinei sistemai;
- Dinamiškai apskaičiuojamas maržas;
- Bankų elgseną įtraukiant skaičiuojamąjį HC (angl. *haircuts*) dydį, kuris apibrėžia skirtumą tarp banko klientui išmokamų palūkanų už tam tikrą indėlį ir palūkanų, už kurias jam skolinama;
- Patobulintos pirkimų / pardavimų strategijos.

### 3.4 Architektūriniai sprendimai

Šio darbo įvade paminėtiems darbo tikslams yra pateikiami architektūriniai modeliai. 3.4 pavyzdyje pateiktas modelis - tai elementariausia įgyvendinto USEGM modelio architektūros koncepcija.



3.4 pav.: USEGM modelio architektūra

USEGM modelio detaliąją architektūrą galite rasti šaltiniuose [29, 30].

### 3.4.1 Dizaino šablonai

Įgyvendinant USEGM modelį remtasi dizaino šablonais (angl. *design patterns*)[31, 32] dėl to, kad ateityje programų palaikomumo ir priežiūros darbai nebūtų sudėtingi, kaip tai aiškiai matoma iš SEGM modelio architektūros atlikus apgražos inžineriją. N

Nuspręsta laikytis griežto objektinio programavimo (angl. *Object oriented*, OO) stiliaus ir naudoti nusistovėjusius bei patikrintus architektūros sprendimus žinomoms problemoms spręsti. 3.1 lentelėje pateikiami jau įgyvendinti šablonai ir paaiškinamas jų panaudojamumas USEGM atžvilgiu.

Eil. Nr.	Dizaino šablono pavadinimas	Pritaikymas USEGM modelyje
1	<i>Abstract Factory / factory method</i>	Algoritmų, dinaminių sąrašų laiko eilutėms, objektų kūrimui
2	<i>Singleton</i>	Pagalbiniai algoritmai, factory objektai, <i>helpers</i> metodai
3	<i>Iterator</i>	Perdengiamas esamas <i>Iterator&lt;E&gt;</i> interfeisas dėl <i>RandomIterator&lt;E&gt;</i>
4	<i>Facade</i>	Griežtai suskirstytos aplikacijos, srities, pagalbinės ir kt. posistemės
5	<i>Command</i>	Planuojama įvesti dėl galimybės naudoti serveriniu režimu
6	<i>Chain of Responsibility</i>	<i>Exception</i> atvejai USEGM modeliui
7	<i>Template method</i>	Įgyvendinamo VSE modelio vykdymo pažingsniui, etapais
8	<i>Strategy</i>	Įgyvendinamų prognozavimo algoritmų klasifikavimui
9	<i>Observer</i>	Grafikų perbraižyme po pasikeitusios skaičiavimų būsenos

3.1 lentelė: USEGM naudojami dizaino šablonai

## 3.5 Testavimo įrankiai

Neskaitant programinės realizacijos išeities teksto testavimo atvejų (*Unit* bei *integracinių* testų) programinės įrangos kokybei užtikrinti, sudarytas testavimo įrankis algoritmų eksperimentinių tyrimams atlikti testuojant įvairiausių atvejus. Tokių testinių atvejų sugeneruotos išeitys pateiktos 6.4 priede.

## Skyrius 4

# REALIZUOTO PROJEKTO TYRIMAS

Šiame skyrelyje abstrakčiai apžvelgsime esminius realizuoto projekto aspektus. Pateiksime sprendimus ateičiai. Iškelsime naujas problemas, kurias reikėtų apsvarstyti ateityje siekiant tobulinti USEGM projektą.

USEGM - tai virtualios akcijų biržos modelis, tai nėra baigtinio algoritmo modelis, kurį galima būtų įgyvendinti ir laikui bėgant atlikti programinės įrangos priežiūros darbus. Šis modelis reikalauja pastovaus tobulinimo bei aptiktų netikslumų šalinimo tiek programinėje realizacijoje, tiek žvelgiant ir iš teorinės pusės: naujų parametrų įvedimas, staigus pasikeitimas finansų sektoriuje (krizė) ir kiti niuansai.

Atlikus projekto programinės įrangos naujumo, funkcionalumo bei galimų pakeitimų, patobulinimų ateityje tyrimą, paaiškėjo, kad tiek į USEGM modelio karkasą, tiek į prognozavimo modelį galima būtų įtraukti daug įvairiausių naujovių bei funkcionalumo, kuris leistų virtualiai modeliuojamą sritį ištirti nuodugniau.

Kaip ir ankstesne tvarka, aprašysime siūlomus tobulinimus prognozavimo posistemės tobulinimui ir dalinai peržvelgsime karkaso tobulinimo galimybes.

### 4.1 Prognozavimų posistemės tobulinimas

Prognozavimų posistemės tobulinimas gali būti padalintas į 2 sritis:

1. Įgyvendintų algoritmų tobulinimas;
2. Naujų modelių į prognozavimo posistemę įtraukimas.

#### 4.1.1 Kiti prognozavimo modeliai

Analizės metu paaiškėjo, kad  $AR(p)$  modelio algoritme yra neoptimizuotų vietų, kurias galima būtų panaikinti ir tokiu būdu priversti atlikti prognozes sparčiau. Reiktų išskirti algoritmų optimizavimo

klausimą. Šis darbas nėra pirmaeilis, tačiau, galimas dalykas, bus prieita prie sistemos resursų ribos, kai ilgai trunkantys skaičiavimai pagaliau iškels šį opų klausimą, todėl reiktų iškart išnagrinėti ir patobulinti skaičiavimo algoritmus. Reiktų paminėti, kad rasta vieta  $AR(p)$  modelio algoritme kaip tik yra Gauso lygčių sistemos sprendimo uždavinys, kuriame dirbama su matricomis.

Naujų modelių į prognozavimo posistemę įtraukimas yra būtinas tobulinimas atliekamų tyrimo aspektu ir pirmiausiai nagrinėjamas klausimas. Šiuo metu yra įgyvendinti autoregresiniai  $RW$ ,  $AR(p)$  bei  $AR - ABS(p)$  modeliai. Netolimoje ateityje reiktų šiuos modelius išplėsti. Yra žinomas AR-ME modelis (angl. *Autoregressive Maximal Entropy*) [12], pagal kurį  $a$  koeficientai yra optimizuojami sprendžiant ne lygčių sistemą Gauso metodu, bet manipuluojant laiko eilutės duomenimis bei jų entropijos dydžiais (nedetalizuosime). Šis patobulinimas skatintų ieškoti naujų galimybių parametrų optimizavimo klausimais.

ARMA modelis yra vienas geriausiai žinomų prognozavimo modelių, kurio pirmoji dalis yra įgyvendinta šiame projekte. Antroji dalis darbų planuose jau yra. Šis modelis bus įgyvendintas pirmučiausiai, kadangi tai yra pagrindinis modelis, į kurį badomi integruoti kiti modeliai, tobulesni, dirbantys su papildomais įvedimo parametrais ir kt. Tai tokie modeliai, kaip ARIMA – autoregresinis integruotas slenkamųjų vidurkių modelis bei ARFIMA – autoregresinis funkcinio integravimo slenkamųjų vidurkių modelis. Šiais metodais remiasi profesionalūs investuotojai, o eilinis mėgėju praktiškai užtenka ir ARMA modelio [21]. Šie modeliai daugiau naudojami realių duomenų prognozėms sudaryti.

USEGM modelio veikimas paremtas eilinio investuotojo strategijai modeliuoti, tad pastarieji aptarti ARMA atskiri atvejai kol kas yra tik galimas, bet nebūtinai, sprendimas.

#### 4.1.2 Dirbtiniai neuroniniai tinklai (ANN)

Vis dažniau akcentuojamas dirbtinių neuroninių tinklų (ANN) pritaikomumas spręsti prognozavimo uždaviniams ar klasifikuoti turimas laiko eilutes, kad atpažinti ir išskirti tam tikrus šablonus (angl. *patterns*). Šiuo metu į USEGM jau yra įtrauktas ANN modelis, gebantis sudaryti prognozę, kurios  $RMSE < 0,07$  ir kuris visai neblogai prognozuoja *Microsoft Corporation (MSFT)* akcijų kursą trumpalaikiu prognozavimu. Tiesa, iteracinis permokymo procesas yra būtinas ir visą šį procesą reiktų automatizuoti. Taikant specialius apmokymus su papildomų kintamųjų įvedimu į modelį (angl. *intervention variables*), kurie paremti statistiniais ir euristiniais skaičiavimais, gaunami tikslesni prognozės rezultatai. Įdomi ir susijusi sritis ateičiai būtų išanalizuoti galimybes panaudoti klasifikatorius (pvz. ID3 [18]), kaip papildomas priemonės nustatant duomenų eilutėse tam tikras tendencijas ir integruoti kartu su ANN standartiniu *feed forward* tipo tinklu (ANN-FF). Šis dalinai įgyvendintas modelis, kuris šiuo metu vis dar yra tyrinėjimų stadijoje, naudoja „*Encog*“ karkasą treniravimui bei kitoms operacijoms [33]. Karkasas pasižymi itin sparčiu tinklo apmokymu: parametrų aibės radimo, pritaikymo (angl. *fitting*). Naudojant *resilient propagation* algoritmą apmokymui bei hiperbolinio tangento parametrų aktyvacijos funkciją, modelis išmokomas prognozuoti

per itin trumpą laiką.

Taipogi būtų įdomu išanalizuoti ANN-ART1 (angl. *Adaptive Resonance Theory 1*) algoritmu paremtą modelio pritaikymą šiame projekte.

Kaip matome, sritis tobulinimui prognozavimų modelių realizavimo klausimu yra labai plati.

## 4.2 Karkaso tobulinimas

### 4.2.1 Architektūra

USEGM modelio karkaso tobulinimas taipogi neapsiriboja keliais tobulinimo ar naujovių įtraukimo darbais.

Vienas iš esminių reikalaujamų atlikti darbų - resursų optimizavimas. Kadangi Java kalba ir taip virtualizuojama sistemoje naudojant JVM (angl. *Java Virtual Machine*), resursų naudojimas, atliekant sudėtingus skaičiavimus, yra ypatingai didelis. Negalime daryti sprendimo perrašant visą logiką į C++ ar C# kalbą, kadangi kompiliatoriai yra priklausomi nuo naudojamos operacinės sistemos (OS), kai, tuo tarpu, Java gali būti integruojama (angl. *embedded*) į bet kokį įrenginį, kuris turi Java sąsajos palaikymą. Netgi į išmanųjį telefoną. Todėl turime kaip įmanoma geriau išnaudoti visus turimus resursus, ko pasekoje reikia tobulinti įgyvendinto karkaso pritaikymą multiprocesorinei sistemai ir tobulinti algoritmą, pagal kurį karkasas naudotų resursus. Į šiuo darbus įeina ir CS analizavimas bei optimizavimas.

Šiuo metu USEGM karkasas įgyvendina storjo kliento (angl. *thick*) architektūrą. Žinant, kad skaičiavimai reikalauja daug resursų bei laiko (skaičiavimai gali trukti ir kelias dienas), ateityje reiktų įtraukti naujumą - skaičiavimų posistemės (į ja įeina ir šiame darbe nagrinėjama prognozavimų posistemė) iškėlimą į serverį, kuri būtų valdoma prisijungus tam tikru momentu. Šiuo principu architektūra taptų plonojo kliento (angl. *thin*), o, dar labiau patobulinus - išmaniojo (angl. *smart*). Serverio dalyje tuomet veiktų Java kalba parašytas servisas (*servlet*), kuris komunikotų (keistųsi duomenimis) su klientu naudodamas RESTful arba RESTless [34] modelių koncepcijas.

Šiam įgyvendinimui reiktų įtraukti *Command* architektūros dizaino modelį (angl. *pattern*), kuris supaprastintų manipuliavimą perduodamoms ir serveryje interpretuojamoms komandoms.

Taigi, abstrakčiai palyginus buvusio SEGM modelio karkaso architektūrinius trūkumus (naujumo klausimu iš technologinės pusės) su esamu nauju USEGM karkasu, nekyla abejonių, kad įgyvendinta architektūra multiprocesorinei sistemai teiks didelę naudą.

Funkcionalumo atžvilgiu USEGM modelyje yra patobulintas paduodamų parametrų valdymo galimybės. Kol kas tai nėra realizuota grafiškai<sup>1</sup>, tačiau sudarytos galimybės architektūroje nustatyti tam tikrų parametrų reikšmes neapribojant tyrimų srities. Tai yra labai svarbus įgyvendintas

---

<sup>1</sup>Turima omenyje grafinė vartotojo sąsaja (angl. GUI)



niuansas.

Kalbant apie parametrų lankstų panaudojimą, kaip vieną iš pavyzdžių, USEGM karkasas įgyvendina investuotojo pirkimo / pardavimo kainų lygių (angl. *levels*) dinaminį nustatymą, ko nebuvo įgyvendinta SEGM modelyje: nustatyti ir įprogramuoti (angl. *hardcoded*) 3 lygiai. Teoriškai, investuotojas prieš atlikdamas pirkimo / pardavimo operacijas gali analizuoti tiek lygių, kiek jis nori. Tačiau praktikoje pasitaiko, kad daugiau nei 3 lygių neanalizuojame. Tačiau tyrimų atžvilgiu, investuotojas, kuris turi 2 lygius elgsis kitaip, nei kitas investuotojas, kuris turi 3 ar 1 lygį kainų nustatyme.

Prie to paties reiktų paminėti, kad taipogi buvo įvestas  $p$  parametrų lankstumas  $RW$ ,  $AR(p)$  bei  $AR-ABS(p)$  modeliams, kas davė rezultatų (žr. 6.4 priedą) ir įrodė, kad kai kur didesnės istorinių duomenų imties analizė gali apmokyti modelį dirbti geriau prognozavimo atžvilgiu<sup>2</sup>.

Šiuo metu USEGM karkase investuotojų pirkimo / pardavimo logikai atlikti sudaroma iteracija pagal eilę, t.y. investuotojams priverstinai taikoma taisyklė - kas pirmas, tas gudresnis. Kad išvengti šio nesusipratimo ir, tuo pačiu, priartinti modelį prie realybėje vykstančių veiksnių, siūlau ateities darbuose pakeisti iteracinį klientų operacijų vykdymą randomizacine tvarka paremtu modeliu. Ši koncepcija iš dalies įgyvendina Lamporto (*Lamport*) sąlygų simuliaciją *happened-before* įvykių samprata, kurie realiai taikomi paskirstytosioms sistemoms (angl. *distributed systems*), besidalinančioms resursais pagal konkurencingumo principus (angl. *concurrency*)<sup>3</sup> [35].

Šiuo metu USEGM modelis generuoja programos vykdymo metu kuriamas išvestis (angl. *logs*) paprastu struktūrizuotu tekstiniu formatu. Buvo pastebėta, kad išvedamų duomenų gausumas apsunkina jų analizavimą. Kadangi duomenų struktūrizavimas yra įgyvendintas, šios išvestys bus standartizuotos XML kalba, kas užtikrins itin patogų priėjimą prie reikiamų duomenų bet kokių įrankių, kuris palaiko XML atvaizdavimą medžio tipo struktūra.

### 4.2.2 Grafinių resursų panaudojamumas (GPU)

Dar viena naujovė, kurią galima būtų įgyvendinti ir kuri labai padėtų išnaudoti turimus sistemos resursus yra į multiprocesorinės sistemos (multi-CPU) koncepciją įtraukti ir grafinius procesoriaus resursus (GPU). Juo labiau, šiandieniniai resursai jau skaičiuojami kaip multi-GPU sąsajos (SLI technologijos). Resursų paskirstymo užduotį Java kalba atlieka *OpenCL* karkasas [36]. Tad, kodėl gi nepanaudojus šių resursų? Yra vienas niuansas: vėlinamas operacijų atlikimo laikas, sugaištamą paskirstant resursus. Ši idėja yra svarstoma ir, galbūt, vėlesniuose projekto etapuose bus įgyvendinta. USEGM atžvilgiu, šios naujovės įtraukimas turėtų tokį plusą, kaip multiprocesorinės sistemos panaudojamumas iki 90% multi-CPU resursų ir pridėdant papildomus (multi-)GPU resursus, kurie pasitarnautų, kaip grafikų perskaičiavimas / perbraižymas realiu laiku<sup>4</sup>, ko pasekoje

---

<sup>2</sup>Apie tai plačiau skaitykite 5 skyriuje „Prognozavimo modelių tyrimas“ 33 puslapyje).

<sup>3</sup>Šavoka yra lygiagretaus (angl. *parallel*) programavimo atskiras atvejais.

<sup>4</sup>Nereiktų painioti su *OpenGL*, *Direct3D* ar kitom grafikos generavimo bibliotekom.

turėtume 4-ių branduolių sistemoje specialiai nestabdomus visus 4 procesus, kurie yra paskirti virtualios akcijų biržos modeliavimui.



## Skyrius 5

# PROGNOZAVIMO MODELIŲ EKSPERIMENTINIS TYRIMAS

Atliksime įgyvendintų modelių  $RW$ ,  $AR(p)$  ir  $AR - ABS(p)$  eksperimentinį tyrimą dviem etapais (kriterijais):

1. Tirsime parametrų optimizavimą autoregresiniais modeliais ir įvertinsime gautų  $a$  koeficientų (žr. formulę 2.1 puslapyje 11) tikslumą: modelio apmokymą ateities prognozėms (angl. *parameter fitting*) ir ;
2. Sudarysime prognozes ateičiai remdamiesi gautais  $a$  koeficientais optimizavimo metu pirmame etape. Įvertinsime prognozuojamų reikšmių tikslumą.

Tyrimus atliksime naudojant šiuos istorinius laiko eilutės duomenis:

- $D_r$  - realios rinkos duomenis, gautus iš [9] šaltinio ir analizuojant *Microsoft Corporation (MSFT)* akcijų atidarymo kainų kursus (iki 2011-05-23 dienos imtinai);
- $D_m$  - sugeneruotus USEGM modelio laiko eilutės duomenis atliekant virtualios rinkos modeliavimą.

Šiame tyrime naudosime 100 laiko eilutės verčių  $a$  parametrų optimizavimui.

Gautus tyrimo rezultatus pavaizduosime diagramomis bei aprašysime atskirus, įdomius tyrimo atžvilgiu, atvejus. Kiekvieno etapo pabaigoje pateiksime apibendrintus rezultatus, susijusius su tiriamuoju objektu. Šio skyriaus pabaigoje pateiksime apibendrinančias autoregresinius modelius tyrimo išvadas.

Aiškumo sumetimais, grafikuose žalia spalva bus žymima reali (žinoma) laiko eilutės vertė, o raudona - prognozuojamoji. RMSE paklaidą žymėsime mėlyna spalva, o AME - oranžine.

Priede 6.2 yra pateikti papildomi eksperimentinių tyrimų grafiniai rezultatai, kai duomenų imtis - 1000 laiko eilutės verčių parametrų optimizavimui (žr. 58 puslapį), o priedo 6.4 puslapyje 65 pateikiamos optimizavimo metu gautos reikšmės ir paklaidos.

Sudarysime testus su skirtingomis duomenų imtimis ir nustatysime, kokią priklausomybę parametrams  $a$  turi istorinių duomenų imtis<sup>1</sup>. Taip pat ištirsime, ar galioja persimokymo (angl. *overfitting*) dėsningumas, kuomet  $a$  parametrai naujoms prognozėms tampa nelankstūs (angl. *flexible*).

## 5.1 Parametru optimizavimo tyrimas (1-as tyrimo etapas)

Šiame poskyryje tirsime parametru optimizavimo kokybę ( $a$  koeficientų gavimą, apmokymą) visiems modeliams ( $RW$ ,  $AR(p)$  bei  $AR - ABS(p)$ ).

Realiai, mes šiame etape nevykdome tiesioginės prognozės dienai  $t$ . Optimizuodami sudarome  $a$  parametru rinkinį, kurių sandaugų sumą su atitinkamom istorinėm laiko eilutės reikšmėm vadinsime bazine (angl. *base*) prognoze šiai dienai  $(t - 1)_b$ . Tai vertė, pagal kurią skaičiuojame modelio gautų parametru (apmokymo) paklaidą lyginant su realiais duomenimis. Šio poskyrio lentelėse nurodomos bazinės kainų vertės tam tikrai dienai.

### 5.1.1 RW modelis

Šiam modeliui sąvoka parametru „optimizavimas“ taikoma tik iš dalies, kadangi optimizavimo koeficientas  $a = 1$  ir  $p = 1$  (plačiau žr. 2 skyrių „Analizė“ 9 puslapyje). Kadangi optimizacijos realiai nevykdėme, turime šios dienos reikšmę  $t - 1$  kaip prognozavimo bazinę reikšmę šiai dienai  $(t - 1)_b$ . Lentelėje 5.1 atvaizduoja tam tikrą kiekį gautų reikšmių po optimizavimo. reikšmės ir prognozuojamos reikšmės sekančiam dienai, RMSE bei MSE naudojant  $D_r$  duomenis. Šiuo atžvilgiu  $D_m$  yra analogiškas atvejis.

Laiko eil. reikšmės nr.	Kaina šiai dienai, $t - 1$	Bazinė kaina prognozei, $(t - 1)_b$	RMSE	MSE
1	30.37	30.37	0	0
2	30.02	30.02	0	0
3	29.58	29.58	0	0
4	29.94	29.94	0	0
5	29.96	29.96	0	0

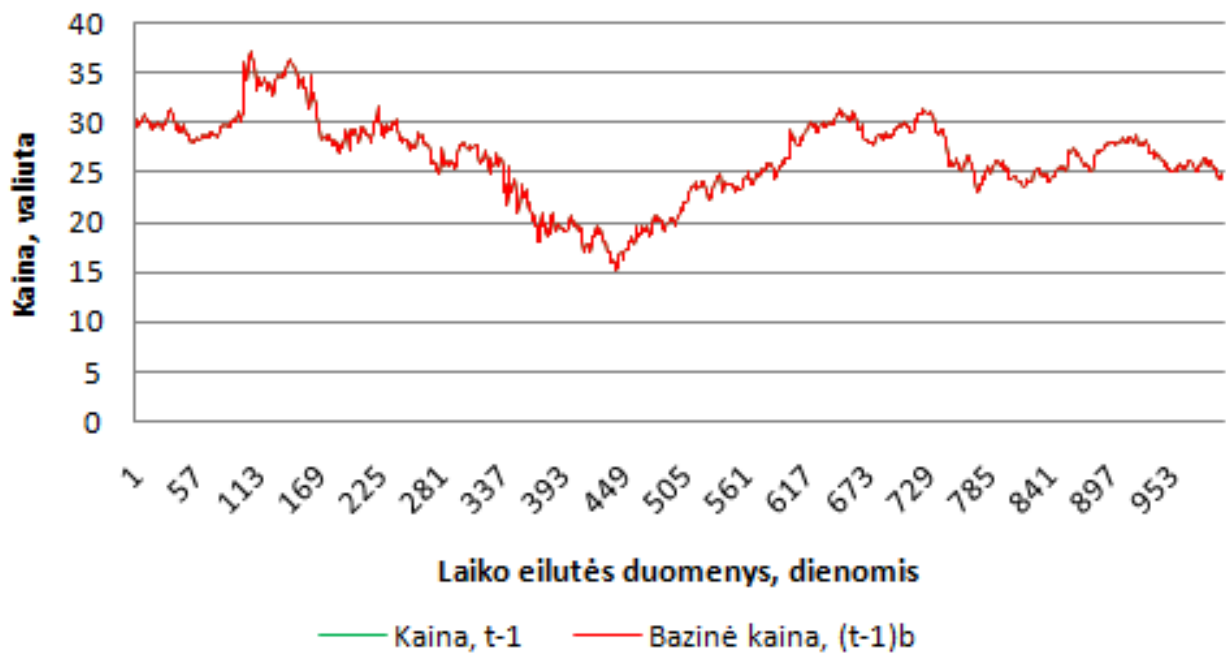
5.1 lentelė:  $RW$ , gauti duomenys po optimizacijos

Modelio idėja - prognozuoti šiai dienai, todėl natūralu, kad RMSE bei MSE yra lygūs 0. Pavyzdyje 5.1 pateikiame grafinį vaizdą realių laiko eilutės reikšmių ir bazinės kainas tai dienai. Matome, kad laiko eilutės perdengia viena kitą. Tokį rezultatą ir turėjome gauti.

### 5.1.2 $AR(p)$ modelis

Nagrinsime  $AR(p)$  modelio atskirus atvejus naudodamiesi 6.4-ame priede esančiais duomenimis.

<sup>1</sup>Taip pat žr. 6.2 priedą 58 puslapyje.



5.1 pav.: *RW*, realių ir bazinių prognozavimų reikšmių laiko eilutės

Tyre naudosime *MSFT* kursus  $D_r$  ir sumodeliuotus duomenis  $D_m$ , kurių imtis - 100. 6.2 priede yra pateikiami pavyzdžiai, kai analizuojamų duomenų kiekis yra 1000 laiko eilutės reikšmių (žr. 58puslapį).

Iš 6.4-ame priede esančių optimizavimo metu gautų reikšmių parenkame mums reikiamą parametru  $p$ , prie kurio turime mažiausią RMSE paklaidą. Minimalios ir maksimalios RMSE paklaidų reikšmės 6.4-ame priede pažymėtos atitinkamai žalia ir raudona spalvomis.

Išrenkame optimalias  $p$  reikšmes atitinkamiems duomenims:

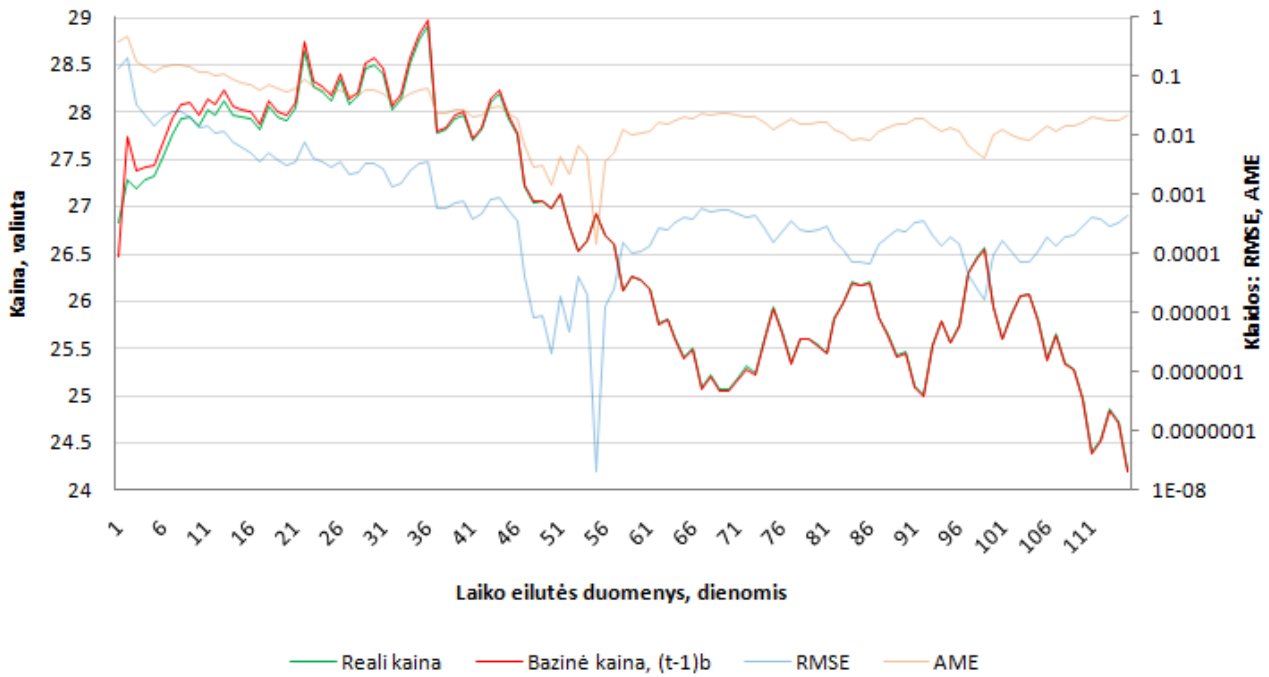
- $D_r$  :  $p = 1$ ; RMSE = 0.142328157;
- $D_m$ :  $p = 1$ ; RMSE = 0.061414359 .

Kaip pastebėjome, mažiausią RMSE gaunami optimizuodami duomenis naudojant  $p = 1$ .

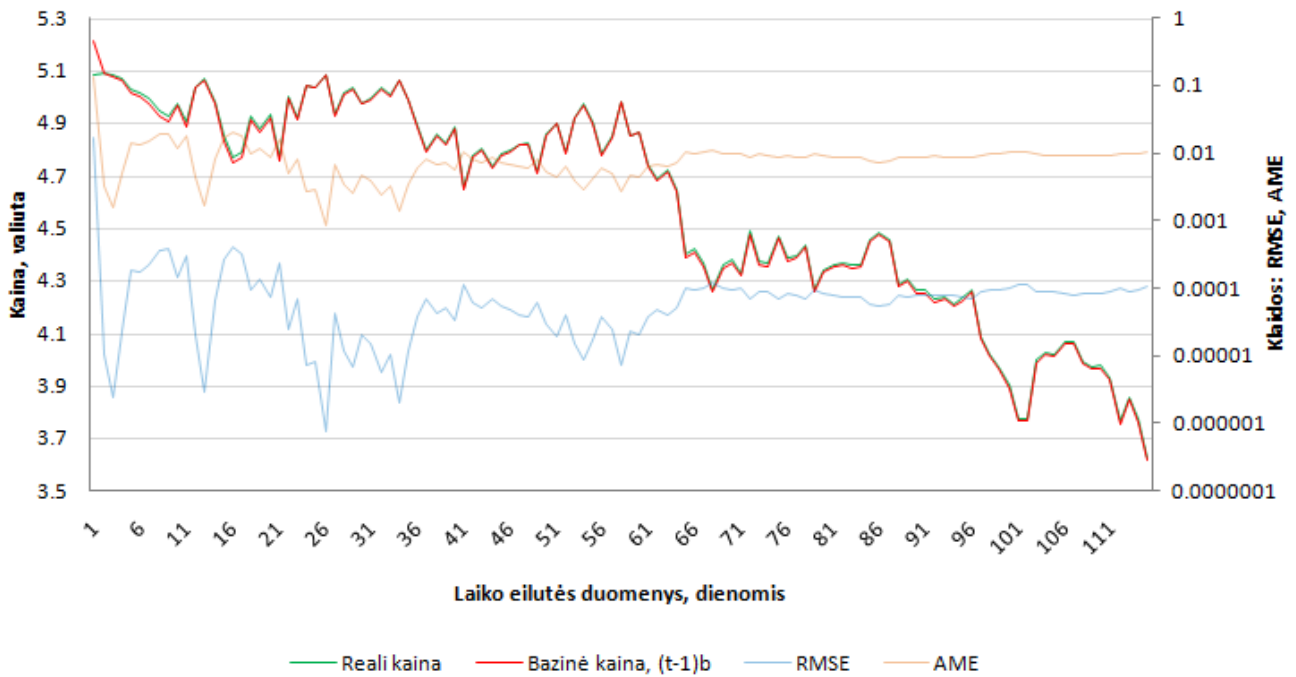
Grafike 5.2 matome pavaizduotą parametru optimizavimo metu gautas reikšmes bei RMSE ir AME paklaidas naudojant  $D_r$ . Pradžioje RMSE buvo gana didelė dėl mažo istorinių duomenų kiekio, o nuo antro ketvirčio optimizavosi ir 56-ą dieną pasiekė minimalią vertę. Vėliau stabilizavosi.

Grafike 5.3 matoma kiek kitokia situacija: modelis pirmame ir antrame ketvirčiuose optimizavo vertę gana neblogai, o nuo 2-ojo ketvirčio - persimokė (įsiminė reikšmes). Tai nėra gerai. Norint gauti optimalų rezultatą šiam modeliui reikia skirti < 50 reikšmių apmokymui. Šis modelis dirbo su  $D_m$ .

5.1. Parametru optimizavimo tyrimas (1-as tyrimo etapas)



5.2 pav.:  $AR(1)$ , parametru optimizavimas naudojant realius duomenis



5.3 pav.:  $AR(1)$ , parametru optimizavimas naudojant sumodeliuotus duomenis

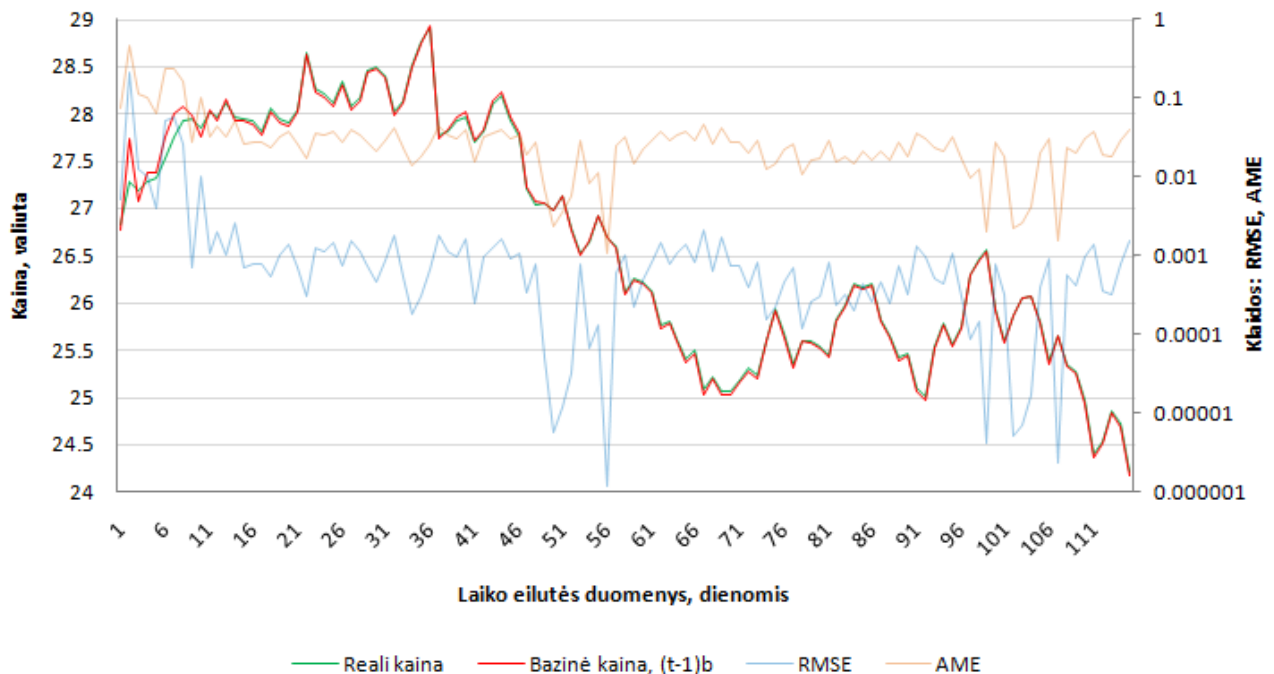
### 5.1.3 AR-ABS( $p$ ) modelis

Naudosimės ta pačia testavimo metodika, kaip ir  $AR(p)$  modelio tyrimui.

Turime optimalias  $p$  reikšmes atitinkamiems duomenims:

- $D_r$  :  $p = 8$ ; RMSE = 0.067658551;
- $D_m$ :  $p = 8$ ; RMSE = 0.020501314.

Grafike 5.4 optimizuojamos  $D_r$  reikšmės. Per pirmąsias 11 dienų modelis darė itin dideles paklaidas skaičiavimuose, tačiau po 15 dienų darbas pasikeitė ir stabilizavosi.  $a$  parametru sudarymui šiam modeliui reikia  $2p$  duomenų kiekio.



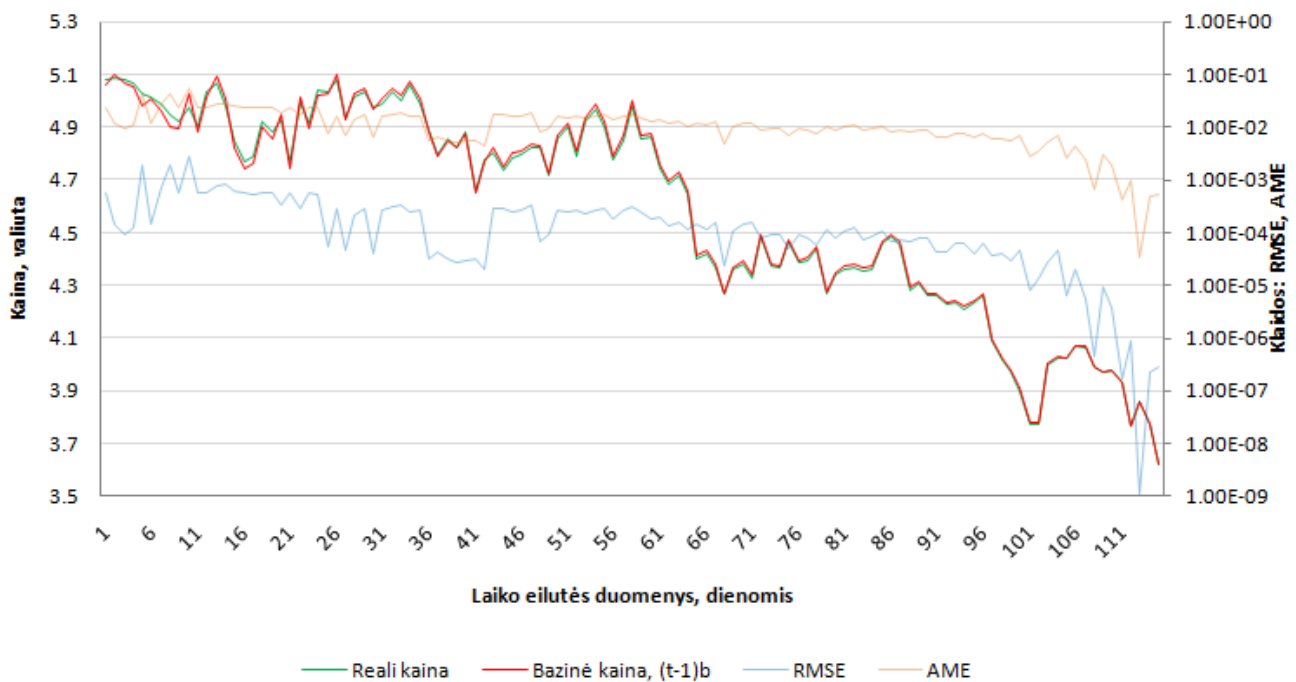
5.4 pav.:  $AR-ABS(8)$ , parametru optimizavimas naudojant realius duomenis

Grafike 5.5 optimizuojamos  $D_m$  reikšmės davė kiek kitokį rezultatą. Modelis mažino paklaidą artėdamas į 4-ą ketvirtį. Šis reiškinį galime paaikškinti tuo, kad modeliuojamų kainų tendencija nuo pat 3-čio ketvirčio ėmė kristi.

### 5.1.4 Gautų rezultatų apibendrinimas

Apibendrinant gautus rezultatus galima būtų teigti, kad tiek  $AR(1)$ , tiek  $AR - ABS(8)$  pateikė gan neblogus rezultatus turint omeny RMSE paklaidas, kurios vertina modelio naudingumą itin subjektyviai. Sulyginti modelių tarpusavyje negalime, kadangi naudotas skirtingas parametras  $p$   $AR - ABS(8)$  modeliui.





5.5 pav.:  $AR-ABS(8)$ , parametrų optimizavimas naudojant sumodeliuotus duomenis

Įvertinant  $AR(1)$  ir  $AR-ABS(8)$  pagal RMSE galime teigti, kad su  $D_r$  duomenimis parametrus  $a$  labiau sekėsi parinkti  $AR(1)$  modeliui (atitinkamai RMSE yra  $2.0692E-08$  ir  $1.17971E-06$ ), o su  $D_m$  -  $AR-ABS(8)$  modeliui (atitinkamai  $7.47497E-07$ ,  $1.10936E-09$ ).

Palyginus autoregresinį modelį su  $RW$  modeliu galime teigti, kad  $AR(p)$  modelis daro didesnę paklaidą pradžioje. Detaliau apie šį tyrimą žiūrėkite 6.2 ir 6.4 priede, kur pateikiamos detalios išvestys įvertinant ir RMSE bei AME paklaidas.

## 5.2 Prognozavimo tikslumo tyrimas (2-as tyrimo etapas)

Sudarydami prognozes naudosimės pirmajame etape gautais  $a$  koeficientais ir aprašysime gaunamą paklaidą. Prognozuosime naudodami tiek  $D_r$ , tiek  $D_m$  duomenimis. Išskyrus  $RW$  modelį, kuriam prognozė sudaroma tik rytojui  $t$ ,  $AR(p)$  ir  $AR-ABS(p)$  modeliams sudarysime prognozę 15 dienų į priekį  $[t, t + 1, t + 2, \dots, t + 14]$  bei įvertinsime daromas paklaidas.

Lentelėje 5.2 pateiktas baltojo triukšmo dydis atspindės nežinomą  $\varepsilon$  prognozuojamai dienai. Šios vertės bus naudojamos visiems modeliams tirti, o reikšmės imamos paeiliui. Šie dydžiai sudaryti naudojant:

- $D_m$  duomenims: 0,02 nepastovumo (angl. *volatility*) vertę padauginus iš atsitiktinės reikšmės, pasiskirsčiusios pagal Gauso dėsnį [16].
- $D_r$  duomenims: 0,2 nepastovumo vertę padauginus taipogi iš atsitiktinės vertės, pasiskirsčiusios

pagal Gauso dėsnį.

Nepastovumo vertė  $\sim 0,02$  taikoma  $D_m$  ir paimta pagal standartinę reikšmę SEGM modelyje, kuri modeliuoja nepagrindinių investuotojų elgseną rinkoje, o  $\sim 0,2$  taikoma  $D_r$  pagal dėsnį, vykstančius realioje rinkoje. Skaiciavimo būdų čia neapžvelgsime.

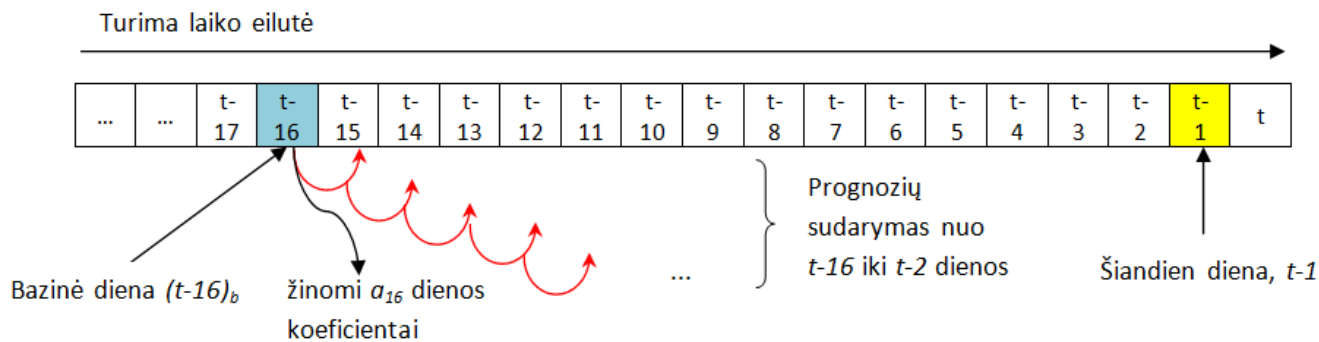
Eil. Nr.	Prognozės diena, $t$	$D_m$ triukšmo dydis, $\varepsilon$	$D_r$ triukšmo dydis, $\varepsilon$
1	$t - 16$	0.018057284	0.21889688
2	$t - 15$	-0.003336065	0.038193927
3	$t - 14$	-0.006868523	0.430087821
4	$t - 13$	-0.022044047	0.202242082
5	$t - 12$	0.004698660	0.048665728
6	$t - 11$	0.024686889	-0.025330161
7	$t - 10$	0.020434376	0.01663748
8	$t - 9$	-0.019991162	-0.217452283
9	$t - 8$	-0.012601562	-0.23480251
10	$t - 7$	0.010275444	-0.214418197
11	$t - 6$	-0.026613774	0.155288779
12	$t - 5$	-0.000815688	3.04E-01
13	$t - 4$	-0.004505119	0.335448255
14	$t - 3$	0.005187371	-0.014269002
15	$t - 2$	0.001893331	-0.221368321

5.2 lentelė: Prognozavimui naudojamos baltojo triukšmo vertės

Sudarydami prognozes naudosimės pirmame etape gautomis  $a$  koeficientų reikšmėmis. Koeficientų reikšmės bus imamos iš laiko eilutės tam tikros istorinės pozicijos - praeities - tam, kad gavę prognozę galėtumėm įvertinti jos daromas paklaidas. Realiai, prognozę galime sudaryti ir dienai  $t$ , tačiau tyrimui toks prognozavimas neturi jokios reikšmės, kadangi mes nežinome realių ateities reikšmių, tad apsiribojame modeliavimu iš praeities  $t - 1 - n$  į dabartį, kur  $n = 15$  - būsimų prognozių kiekis. Šių modelių sudaromų prognozių dienų skaičiaus didinti neverta, kadangi visi realizuoti autoregresinės dalies modeliai yra pritaikyti trumo periodo prognozavimui.

Pagal aprašytus parametrus gauname, kad bazinė diena, kurios  $a$  koeficientus imsime, yra  $(t - 16)_b$ , t.y. prognozuosime 15 dienų į dabartį pradedant  $t - 16$  diena ir baigiant  $t - 2$ , kad galiausiai galėtume palyginti  $t - 2$  su jau turima  $t - 1$  (šios dienos) reikšme.  $(t - 16)_b$  koeficientai a visą prognozavimo laiką išliks tie patys. Keisis tik sekančiais dienais prognozės sudarymui naudojami istoriniai duomenys  $p$ .

Kad būtų aiškiau, 5.6 pavyzdyje pateikiama schema, pagal kurią bus imami duomenys sudaryti prognozėms.

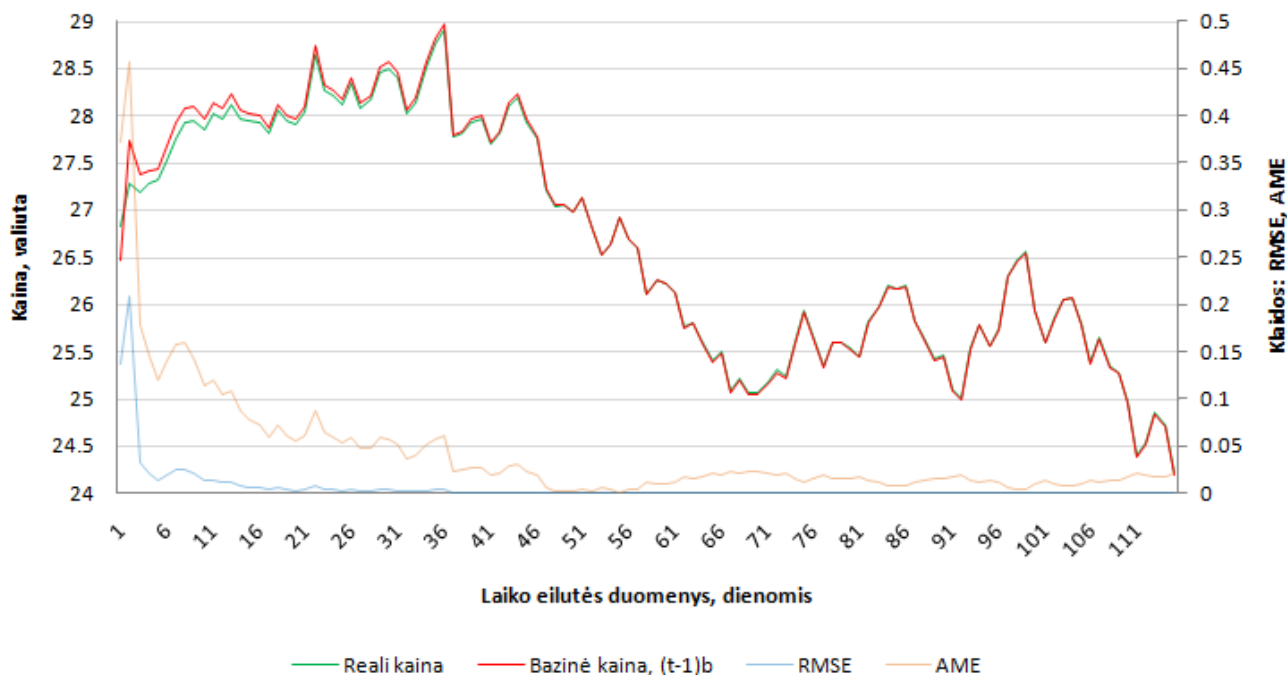


5.6 pav.: Duomenų analizavimo schema prognozių sudarymui

### 5.2.1 RW modelis

Modeliuosime *RW* modelio prognozavimą. Šiuo modeliu 15-os dienų prognozės nesudarysime, kadangi tai yra modelis, kuris prognozuoja tik sekančią dieną, todėl, pagal formulę 2.7, sudarome prognozes sekančioms dienoms iteraciniu būdu ir apskaičiuojame RMSE bei MSE padarytas paklaidas.

Pavyzdyje 5.7 matome prognozes naudojant  $D_r$ , taip pat įvertintas paklaidas.

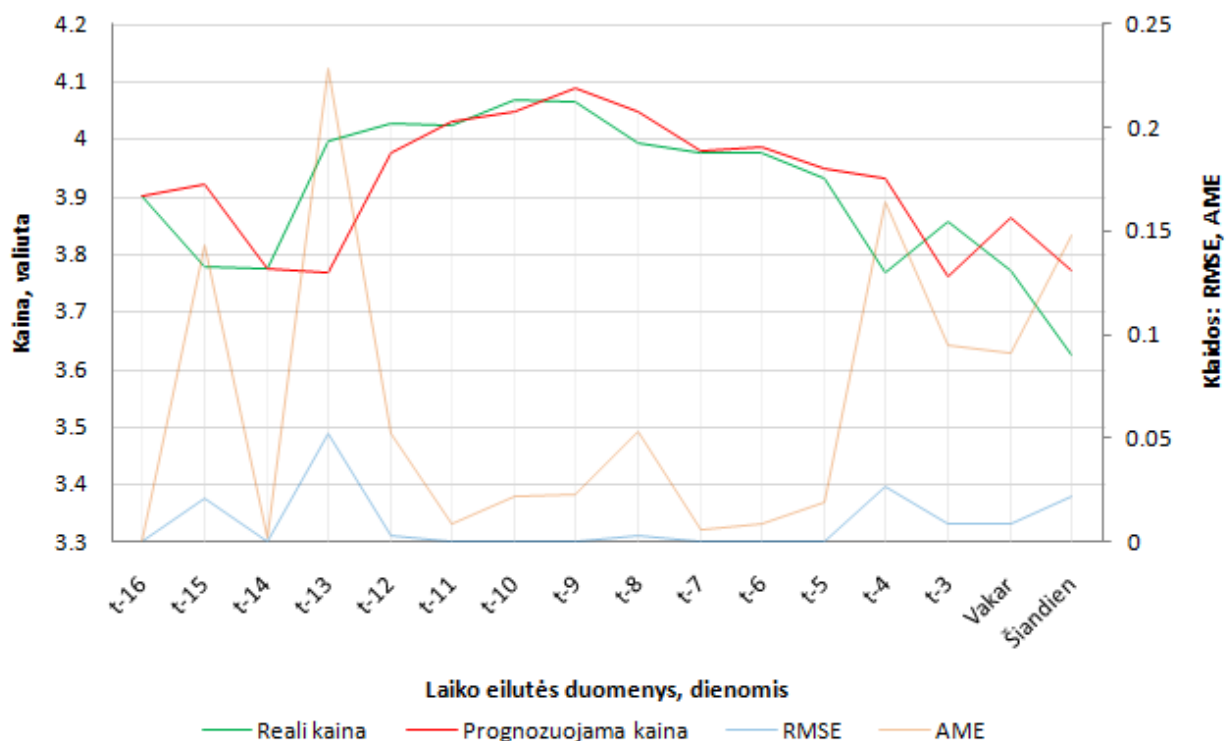


5.7 pav.: *RW*, prognozių sudarymas realiems duomenims

Kaip matome iš 5.7 grafiko, prognozuojamos kainos yra pasislinkusios per vieną žingsnį į dešinę. Taip ir turėtų būti, kadangi šis modelis prognozuoja reikšmes rytojui - kaip šiandien +  $\varepsilon$  (triukšmas). Prognozavimo rezultatų tikslumas čia priklauso nuo aplinkos poveikio  $\varepsilon$ . Kuo

tiksliu jį pasirenkame - tuo tikslesnę gausime prognozę.

Duomenų prognozavimas  $D_m$  duomenimis pavaizduotas 5.8 paveiksle.



5.8 pav.: RW, prognozių sudarymas modeliuojamiems duomenims

Lentelėje 5.3 matome suvestines ribines ir suvidurkintas reikšmes, kurios gautos sudarant prognozę RW modeliu.

Duomenų tipas	Max RMSE	Min RMSE	Max AME	Min AME	Vid. RMSE	Vid. AME
$D_r$	0.511638038	0.000238649	0.715288779	0.015448255	1.203754506	0.254148886
$D_m$	<b>0.051923151</b>	<b>3.98141E-06</b>	<b>0.227866521</b>	<b>0.001995348</b>	<b>0.381526663</b>	<b>0.070848019</b>

5.3 lentelė: RW, ribinės ir suvidurkintos paklaidų reikšmės

Kaip matome iš 5.3 lentelės, RW modeliui geriau sekėsi prognozuoti  $D_m$  duomenis (optimaliausios reikšmės pažymėtos juodesniu šriftu). Taip yra dėl to, kadangi nepastovumo reikšmė (angl. *volatility*) buvo invariantinis dydis tiek sugeneruotam triukšmui, tiek modeliuojamiems duomenims.

### 5.2.2 AR(p) modelis

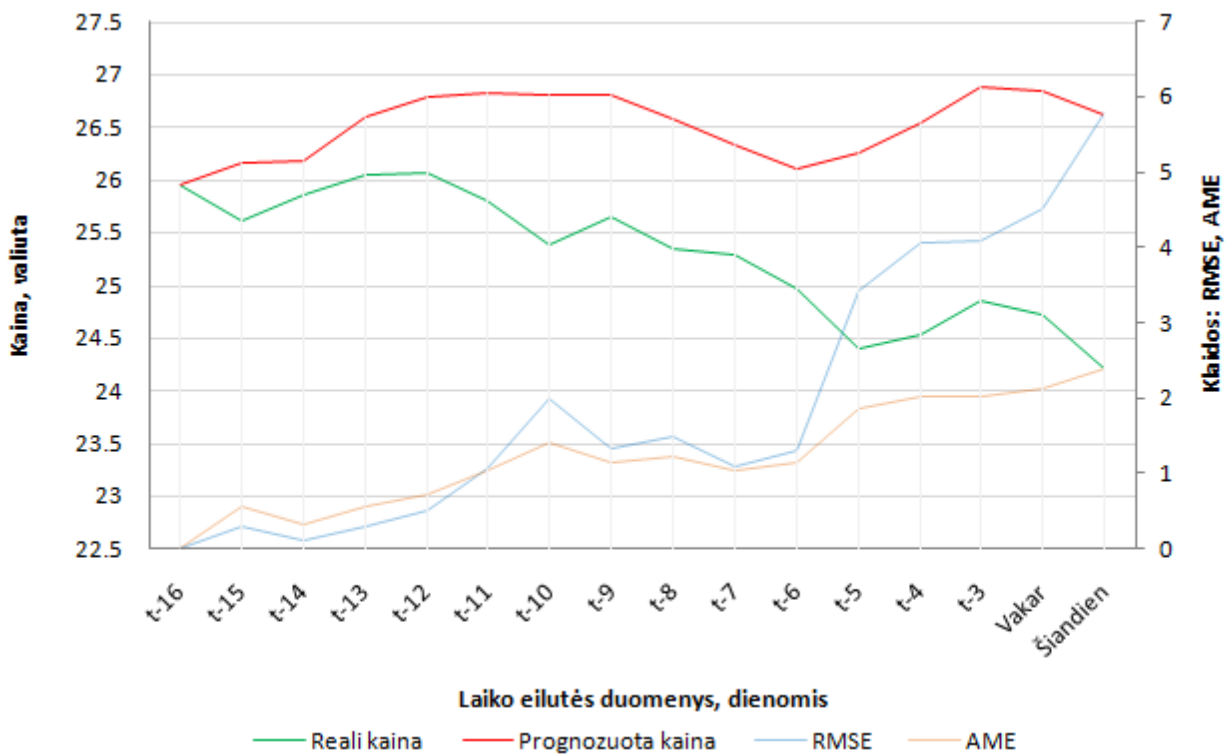
Naudodamiesi pirmame etape apskaičiuotu parametru rinkiniu  $a$  sudarysime prognozę 15 dienų (iki dienos  $t - 1$ ), analogiškai, kaip buvo prognozuojama RW modeliu.

Invariantiniai koeficientai AR(1) modeliui pradedant  $t - 16$  diena:

5.2. Prognozavimo tikslumo tyrimas (2-as tyrimo etapas)

- $D_r$  duomenims:  $a_{t-16} = [ 0.999624340 ]$ ;
- $D_m$  duomenims:  $a_{t-16} = [ 0.997411042 ]$ .

Paveiksle 5.9 matome modeliu  $AR(1)$  atliktas prognozes 15-ai dienų iki šiandien  $t - 1$ . Pirmajame ketvirtyje vertės buvo prognozuojamos su nedidele paklaida, tačiau nuo antrojo jos pradėjo didėti iki pat dienos  $t - 1$ . Tai reikštų, kad  $AR(1)$  modelis su  $p = 1$  reikšme tinka tik labai trumpam periodui prognozuoti į ateitį.



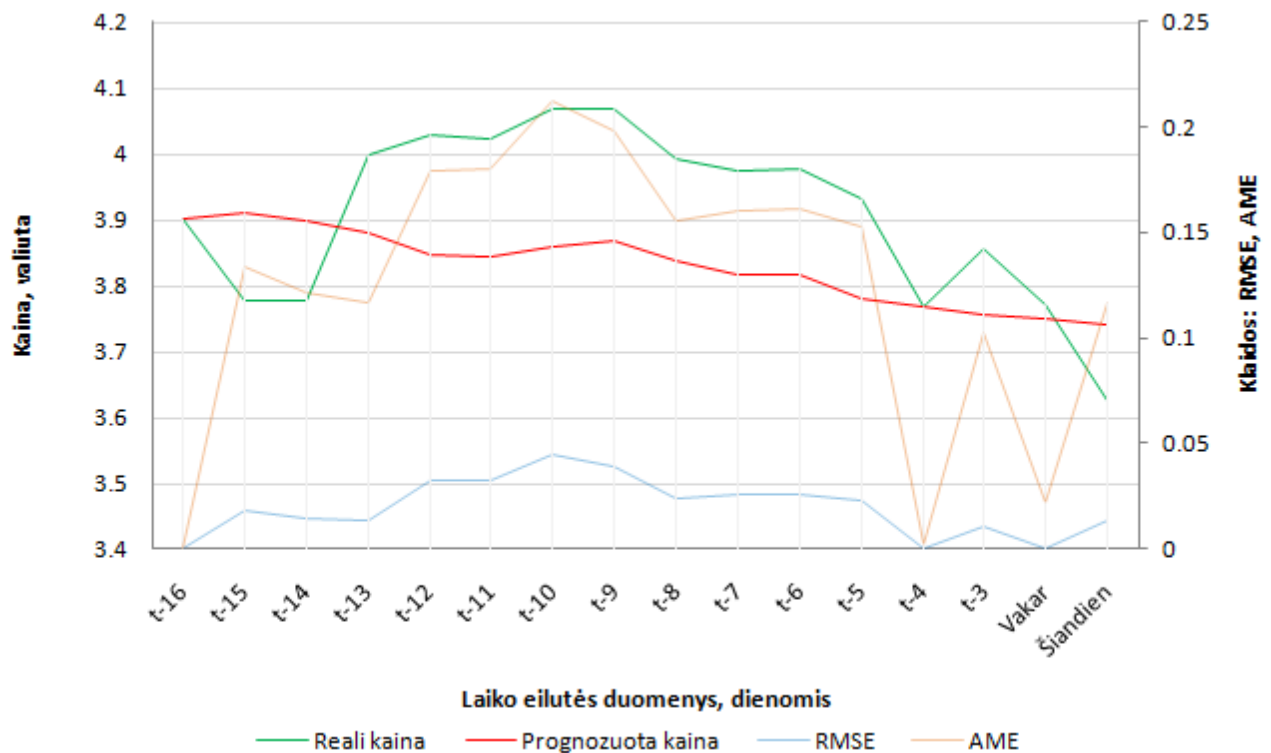
5.9 pav.:  $AR(1)$ , prognozių sudarymas realiems duomenims

Paveiksle 5.10 matome įdomų variantą. Prognozavimo vertė - realių kainos verčių koreliacija. Matoma kainų mažėjimo tendencija. Toks rezultatas nėra labai geras. Prognozė pasitvirtino tik vienoje  $t - 4$  dienai, kurios  $RMSE = 3.96071E-06$ .

Lentelėje 5.4 pateikiami gautos paklaidų ribinės ir suvidurkintos reikšmės. Matome, kad geriau modeliui sekėsi prognozuoti  $D_m$  duomenis.

Duomenų tipas	Max RMSE	Min RMSE	Max AME	Min AME	Vid. RMSE	Vid. AME
$D_r$	5.771834998	0.107271296	2.402464359	0.327522971	5.597133965	1.30367918
$D_m$	<b>0.044860198</b>	<b>3.96071E-06</b>	<b>0.211802262</b>	<b>0.001990154</b>	<b>0.564438148</b>	<b>0.134194664</b>

5.4 lentelė:  $AR(1)$ , prognozavimo kokybės įvertinimas.



5.10 pav.:  $AR(1)$ , prognozių sudarymas sumodeliuotiems duomenims

### 5.2.3 $AR-ABS(p)$ modelis

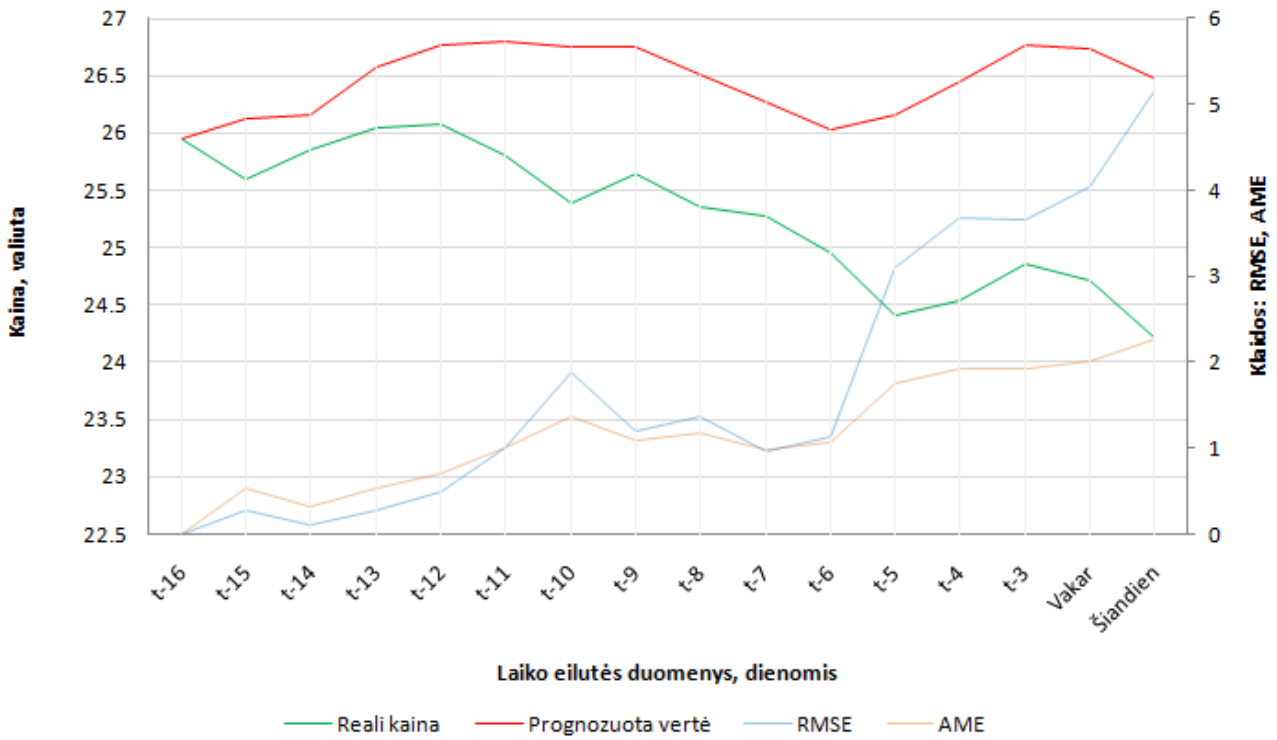
Invariantiniai koeficientai  $AR-ABS(8)$  modeliui pradedant  $t-16$  diena:

$$\bullet D_r \text{ duomenims: } a_{t-16} = \begin{bmatrix} 1.016772270 \\ -0.020408033 \\ 0.006734742 \\ -0.001722294 \\ 0.006796232 \\ 0.000125767 \\ -0.002569676 \\ -0.006484798 \end{bmatrix}; D_m \text{ duomenims: } a_{t-16} = \begin{bmatrix} 1.000643075 \\ -0.001900749 \\ -0.001006899 \\ -0.005751342 \\ 0.005958371 \\ -0.003035397 \\ -0.003652668 \\ 0.009998362 \end{bmatrix}.$$

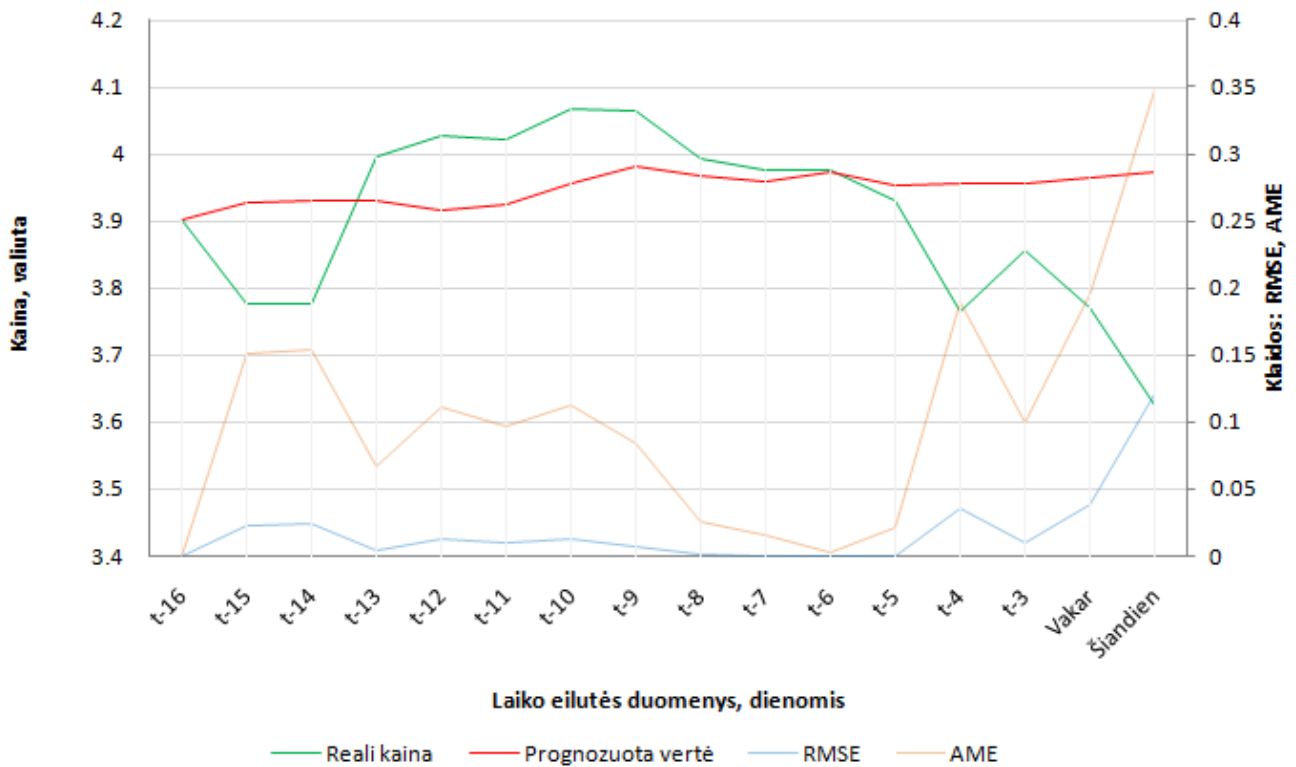
Paveiksle 5.11 matome sudarytą prognozę naudojant  $AR-ABS(8)$  modelį ir  $D_r$  duomenis. Prognozavimas, kaip ir  $AR(1)$  modeliu, tinkamas kelioms dienoms sudaryti. Vėlesnėse dienose matomas nemažas skirtumas tarp prognozuojamų reikšmių ir gautų. Įdomu tai, kad tendencijos išlaikytos, kas reikštų, jog parametrai pamokyti gerai, tačiau trūksta papildomo duomenų apdorojimo siekiant sumažinti vis didėjančias paklaidas.

Paveiksle 5.12 grafikas atspindi  $AR(1)$  modelio darbą, bet prognozių sudarymas modeliuotiems duomenims nėra visiškai geras.

5.2. Prognozavimo tikslumo tyrimas (2-as tyrimo etapas)



5.11 pav.: AR-ABS(8), prognozių sudarymas realiems duomenims



5.12 pav.: AR-ABS(8), prognozių sudarymas sumodeliuotiems duomenims

Gautus rezultatus apibendrinančias charakteristikas matome 5.5 lentelėje. Prognozuoti geriau sekėsi  $D_m$  vertes.

Duomenų tipas	Max RMSE	Min RMSE	Max AME	Min AME	Vid. RMSE	Vid. AME
$D_r$	5.154800221	0.096520628	2.270418512	0.310677692	5.320023295	1.241182989
$D_m$	<b>0.121438431</b>	<b>7.06711E-06</b>	<b>0.348480173</b>	<b>0.002658403</b>	<b>0.547149247</b>	<b>0.111688742</b>

5.5 lentelė:  $AR-ABS(8)$ , prognozavimo kokybės įvertinimas.

## 5.2.4 Gautų rezultatų apibendrinimas

Gautus prognozavimo rezultatus atspindi parinktų parametrų panaudojamumas. Matoma korektiško prognozavimo tendencija  $D_r$  duomenims, tačiau vertės yra nuolat augančios. Visi modeliai geriausiai prognozavo  $D_m$  vertes, kadangi jos buvo sudarytos generuojant atsitiktines reikšmes pagal Gauso pasiskirstymo dėsnį. Taip ir turėjo būti. Įsitikinome, kad įgyvendinti  $RW$ ,  $AR(p)$  ir  $AR - ABS(p)$  modeliai dirba korektiškai.

Apibendrinus visus gautus prognozavimo rezultatus paaikškėjo, kad geriausia prognozė rytojui yra šios dienos kaina +  $\varepsilon$  (atsitiktinis triukšmas).

## 5.3 Tyrimo Rezultatų įvertinimas

Atlikę tyrimus su įvairiais prognozavimo modeliais ir įvairiom duomenų imtim gavome 5 skyriuje aprašytus rezultatus. Apibendrinant rezultatus reiktų paminėti, kad  $RW$ ,  $AR(p)$  ir  $AR - ABS(p)$  modeliai tinka USEGM virtualioms prognozėms apskaičiuoti. Realioje rinkoje šie metodai būtų pats paprasčiausias būdas, kaip nuspėti sekančios dienos galimą rinkos kainą, ir būtų gerokai per silpni, kadangi realios rinkos prognozės įvertina papildomus kintamuosius, daromas paklaidas prognozės metu ir kt. kriterijus, kaip, pavyzdžiui - ARMA modelis. Daugiau tyrimą apibendrinančių rezultatų rasite 6.2 priede 58 puslapyje.

Tyrimo metu paaikškėjo, kad įgyvendinti modeliai dirba taip, kaip ir buvo numatyta [3] specifikacijoje. Sudaryti apie ~300 testinių atvejų, kurie įrodo algoritmų teisingą veikimą. 6.2 priedo internetinėje nuorodoje galima parsisiųsti ir peržiūrėti testinių atvejų sugeneruotas ataskaitas. Testavimo atvejų pilnumas yra ~80%; Algoritmų sudaromų verčių detalaus tyrimo pavyzdžiai pateikiami 6.3 priede.

### 5.3.1 RW modelio įvertinimas

$RW$  modeliu prognozių sudarymas yra tiksliausias ir mažiausiai resursų reikalaujantis būdas. Esminė šio metodo idėja - prognozuoti rytojui kaip šiai dienai [20], t.y. rytojaus kaina bus panaši į šios dienos kainą. Čia galime pastebėti normaliojo Gauso skirstinio [16] esminę idėją: dideli pokyčiai gali būti su labai maža tikimybe, o maži - su labai didele. Kaip nebūtų keista,



šio metodo prognozės pasitvirtina visose finansų rinkose. Šio modelio skaičiavimo netikslumai pastebėti SEGM realizacijoje ištaisyti ir įgyvendinti naujame USEGM modelyje.

### 5.3.2 Autoregresinių modelių įvertinimas

$AR(p)$  modelis sudaro optimalias<sup>2</sup> prognozes, kai reikšmių standartinis nuokrypis nėra didelis. Ryšium su tuo, kad  $AR(p)$  yra optimizuojanti funkcija ir manipuliuoja reikšmių kvadratais, bet kokios reikšmės staigus padidėjimas laiko eilutėje pablogintų tolimesnes prognozes ir generuotų didelę RMS paklaidą [23].

Autoregresiniai modeliai naudoja istorinius laiko eilutės duomenis. Atsimenamų į praeitį reikšmių kiekis yra  $p$ , taigi modelis reikalauja  $p + 1$  laiko eilutės reikšmių. SEGM realizacijoje šis metodas turėjo įgyvendintą algoritmą, kuris, jei trūkdavo duomenų, būdavo atitinkamai mažinama  $p$  vertė iki reikiamos. Tai yra abejotinas sprendimas ir tą parodė SEGM modelis pradinėse modeliuojamos prognozės sudarymo iteracijose. Ši SEGM bruožą atskleidė virtualių investuotojų daromos paklaidos pradinuose etapuose. Reiktų paminėti, kad realiose rinkose prognozės sudarymas, kai nėra tam tikro kiekio reikiamų istorinių duomenų, pasitaiko retai, tačiau virtualiame modelyje pradiniai, kad ir  $p = 9$  žingsniai, gali būti lemiami. USEGM modelyje buvo nuspręsta reikiamus  $p$  duomenis generuoti iš turimų istorinių duomenų juos suvidurkinant ir pridėdant atsitiktinį dydį  $\varepsilon$ , gautą pasitelkus atsitiktinių skaičių generatorių pagal Gauso dėsnį ir jo gautą vertę padauginus iš numatyto kintamumo rinkoje (angl. *volatility*). Šis metodas pasitvirtino praktiškai ir tai įrodo gauti tyrimų rezultatai. Šio algoritmo tikslumas yra tiesiogiai proporcingas dydis turimų istorinių duomenų kiekiui.

USEGM realizacijoje autoregresinių modelių koeficientų matricos sudarymui naudojami visi laiko eilutės duomenys. Tobulinimo darbuose siūloma šį kiekį imti kaip invariantinį dydį ( $\sim 30 - 50$ ), kadangi prognozės sudarymui visi istoriniai duomenys turi menką vertę dėl tendencijos kisti beveik nenuspėjamai. Kaip ir ANN apmokyme bei prognozavime, t.y. permokyme, duomenys „įsimenami“ ir prognozės tampa nelanksčios [17, 18]. Paveiksle 5.13 pateikiame prognozės sudarymo laiko priklausomybės grafiką nuo analizuojamų laiko eilutės fiksuoto duomenų kiekio.

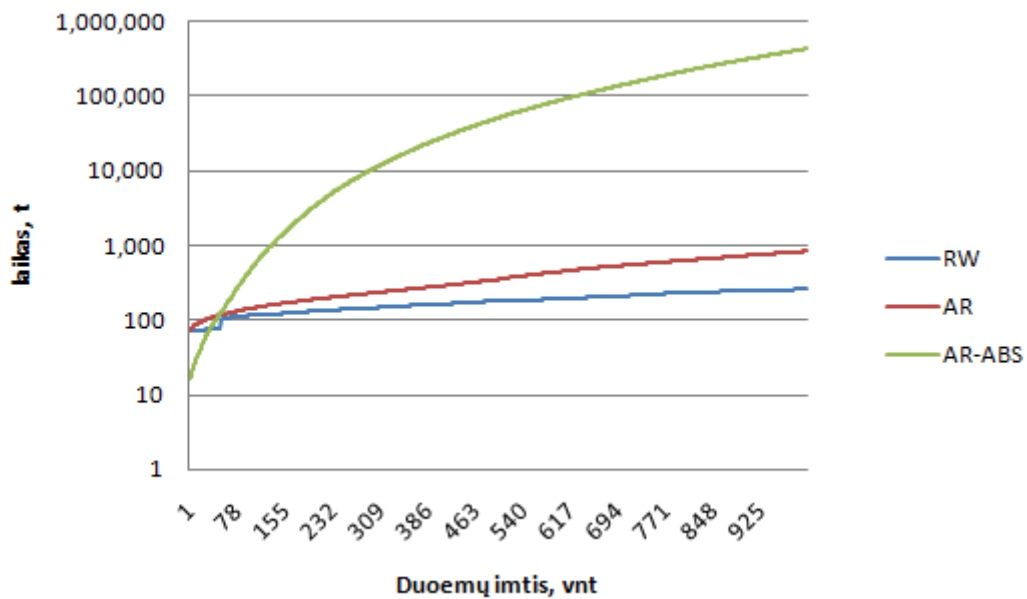
Iš 5.13 paveikslo matome, kad modeliai pagal jų algoritmo sudėtingumą pasiskirstę atitinkamai:  $RW < AR(p) < AR - ABS(p)$ . Laikas  $t$  pavaizduotas logaritminėje skalėje  $\log_{10}$  pagrindu. Ateities darbuose siūloma optimizuoti ciklus, kurie turi tuščių veiksmų. Tai sutaupyto laiko. Laikas - santykinis dydis USEGM modelyje. Realybėje prognozių sudarymas neturi viršyti dydžio  $t/2$ , kur  $t$  - viena diena (24 valandos).

Iš gautų tyrimo rezultatų galime daryti išvadą, kad ši  $AR(p)$  modelio realizacija yra patobulinta pirminio SEGM varianto atžvilgiu ir tinkama tolimesniems USEGM tyrimams.

Tyrimas parodė, kad prie didesnių  $p$  reikšmių dažniausiai gaunama didesnė RMSE paklaida. Tai dar kartą įrodo, kad atsimenamų į praeitį reikšmių kiekis - nėra būdas aproksimuoti optimalią

---

<sup>2</sup>Laikoma, kad gera prognozė yra tuomet, kai RMSE yra ne daugiau kaip 0,01 .



5.13 pav.: Modelių algoritmų sudėtingumo palyginimas

prognozę.

USEGM modelio naujumą pabrėžia įvestas  $AR - ABS(p)$  modelis, kuris mažiausių kvadratų optimizavimą keičia optimizavimu absoliučiomis reikšmėmis. Kaip matome iš tyrimų rezultatų, prognozuojamos vertės RMSE paklaidos yra daug mažesnės nei  $AR(p)$  modelio.

Lyginant autoregresijos metodus tarpusavyje tyrimo metu paaiškėjo, kad  $AR-ABS(p)$  modelis labiau atspindi smulkesniųjų investuotojų savybes. Prognozuojamų reikšmių tais pačiais duomenimis RMSE paklaidų suvidurkintų reikšmių dydis yra mažesnis nei  $AR(p)$  modelio.

### 5.3.3 Metodų taikymas

Praktikoje labiausiai taikytinas būtų  $AR - ABS(p)$  modelis dėl tikslesnio prognozės sudarymo lyginant su  $AR(p)$  metodu. Tačiau realybėje prognozes sudarantys investuotojai elgiasi skirtingai. Mūsų tikslas būtų ir yra pritaikyti įvairaus lygio prognozavimo įrankius virtualiam akcijų biržos modeliui USEGM.

Metodai patikslinti, patobulinti, ištestuoti ir palyginti tarpusavyje, žinomos jų charakteristikos. Šie tyrimai yra pagrindas tolimesniems USEGM modelio tobulinimams.



# IŠVADOS

- Ištirta SEGM modelio architektūra, aptikti netikslumai, išsiaiškinti netikslumų taisymo ir tobulinimo darbai, kurie yra įgyvendinti USEGM modelyje. Tyrimo metu paaiškėjo, kad SEGM modelio prognozavimo posistemė turėjo netikslumų ir jos daromos prognozės turėjo įtakos modeliuojant virtualios akcijų biržos vartotojo sprendimą tolimesniuose etapuose;
- Realizuotas USEGM modelio storojo kliento (angl. *thick client*) architektūros karkasas Java kalba multiprocesorinei sistemai pagal [3] reikalavimus. Panaudota standartinė *Java Concurrency* biblioteka. Tam tikras paskirtų CPU branduolių skaičius  $J$  bus paskirtas virtualioms akcijų biržoms, kurios turės  $I$  virtualių investuotojų. Ištirtos CS: investuotojų akcijų pirkimas / pardavimas, dėl kurių prognozuojamas multi-CPU sistemos panaudojamumas bus ~90% visų resursų;
- Realizuota prognozavimo posistemė, kurioje patobulinti ir ištaisyti  $AR(p)$  ir  $RW$  prognozavimo algoritmai. Įvestas naujas  $AR - ABS(p)$  modelis prognozių sudarymui;
- Atlikę tyrimus pastebėjome, kad finansinių rodiklių prognozavimas yra sudėtinga užduotis, kadangi visus pokyčius rinkoje lemia tai, ką galvoja investuotojas tam tikrą dieną ir kokius sprendimus ar veiksmus jis atlieka. Pirmaeilius dėsningumus (angl. *patterns*) laiko eilutėse atrasti sudėtinga. Antraeilis, tokius kaip savaitės diena, švenčių ar nedarbo dienos, - nesunku. Todėl visiškai tikslios prognozės sudaryti neįmanoma;
- Numatyti ateityje USEGM modelio tobulinimo darbai tyrimų atžvilgiu:
  - Į prognozavimo posistemę bus įtraukti nauji algoritmai: ARMA, ARFIMA, ANN-FF, ANN-ART1. Pastarieji du - ANN aproksimavimo funkcijos;
  - Dėl skaičiavimo posistemės naudojamų itin didelių resursų planuojama patobulinti USEGM storojo kliento architektūrą į išmaniojo kliento (angl. *smart client*) perkeliant skaičiavimų posistemę į serverį bei pridėdant GPU resursus lokalioms operacijoms atlikti naudojant *OpenCL* karkasą;
- Modelių efektyvumui įvertinti buvo pasirinkta vidutinė kvadratinės šaknies paklaida (RMSE) ir absoliutinė paklaida (AME). Gauti modelių  $AR(p)$ ,  $AR - ABS(p)$  ir  $RW$  rezultatai buvo

sulyginti tarpusavyje. Įvertintas kiekvieno atskirtai modelio prognozavimo tikslumas esant tam tikrai atsimenamų duomenų į praeitį reikšmei autoregresinių metodų atžvilgiu;

- Atliktas prognozavimo posistemės modelių  $RW$ ,  $AR(p)$  ir  $AR - ABS(p)$  tyrimas. Laiko eilutės duomenų imtys - 100 ir 1000 reikšmių. Ištirti realūs rinkos duomenys, USEGM sugeneruoti atvejai bei specialūs testai prognozavimo metodų tikslumui nustatyti. Modelių rezultatų apibendrinimas:
  - $AR(p)$  modelis dirba puikiai, kai laiko eilutės reikšmės nėra labai išsibarsčiusios, t.y. kai standartinis nuokrypis nėra didelis. Su labiau išsibarsčiusiais duomenimis gaunama vis didesnė RMSE paklaida;
  - Prognozuojant  $AR - ABS(p)$  modeliu visais atvejais buvo gauti geresni rezultatai. Galime teigti, kad įgyvendintas  $AR - ABS(p)$  modelis yra USEGM modelio prognozavimo posistemės patobulinimas. ir, tuo pačiu, naujumas;
  - $RW$  modelis yra teisingiausiai prognozuojantis modelis todėl, kad rytojaus prognozė yra šios dienos kaina pridėjus atsitiktinį triukšmo dydį, kas realioje finansinių rinkoje pasitvirtina. „Rytoj - kaip šiandien“ [20].
- Nustatėme, kad  $AR(p)$  ir  $AR - ABS(p)$  modeliams prognozės buvo tikslesnės su virtualiai sumodeliuotais duomenimis, kadangi galimų ateities reikšmių dydžiai buvo apibrėžti Gauso pasiskirstymo dėsnium.
- Patobulinti prognozių autoregresiniai  $AR(p)$  ir  $AR - ABS(p)$  modeliai, kai atsimenamų į praeitį reikšmių skaičius yra didesnis nei turimų duomenų kiekis. Įgyvendintas duomenų generavimas atbuline tvarka. Tyrime paaiškėjo, kad toks sprendimas pasitvirtino ir yra geras.
- USEGM modelio tobulinimo darbai jau yra numatyti. Tyrimai tęsiami.

# LITERATŪRA

- [1] J. MOCKUS; A. RAUDYS. *On the Efficient-Market Hypothesis and Stock Exchange Game Model*. 2010. v, vi
- [2] J. MOCKUS; E. BERNOTAS; I ZLIOBAITE. Segm - virtualios akcijų biržos žaidimo modelis. aplikacija, [ žiūrėta 2011-04-15 ]. <http://soften.ktu.lt/~mockus/stockmatulis/indre.html>. v, vi, 6, 7, 13, 14, 16, 18
- [3] J. MOCKUS. On simulation of the nash equilibrium in the stock exchange contest. *Journal of Global Optimization. Informatica*, 2003, 2009, 2010, 2011. v, vi, 5, 6, 7, 12, 13, 18, 25, 45, 49
- [4] T.S. YU. A model used to explain financial crisis. 2009. 5
- [5] D. REAGLE; D. SALVATORE. *Robustness of Forecasting Financial Crises*. Springer, 2005. 5
- [6] E. BERNOTAS. Finansinių ir fizinių procesų prognozavimo modelių tyrimas ir taikymas. Master's thesis, KTU, 2009. 5, 9, 10, 12, 14, 15, 16, 17, 21
- [7] Stock exchange model. risks analysis, [ žiūrėta 2009-10-15 ]. <http://www.stockexchangesecrets.com/risk-analysis.html>. 5, 6, 17
- [8] A. HAIDARI. Literatūros apžvalga. virtualus finansų rinkos modelis. finansinių prognozavimų modelių programinis realizavimas, tyrimas ir taikymas naudojant dirbtinius neuroninius tinklus (ann). 2009. 6
- [9] Stock Market. Historical quotes. finansinių operacijų akcijų biržų istoriniai duomenys. duomenų šaltinis., [ žiūrėta 2011-05-21 ]. <http://finance.yahoo.com>. 6, 7, 33, 58
- [10] T. RAMANAUSKAS; A.V. RUTKAUSKAS. Building an artificial stock market populated by reinforcement-learning agents. *LB (Lietuvos Bankas)*, No 6, 2009. 6, 9, 17
- [11] A. HAIDARI. Projekto paraiška. virtualus finansų rinkos modelis. finansinių prognozavimų modelių programinis realizavimas, tyrimas ir taikymas naudojant dirbtinius neuroninius tinklus (ann). 2009. 7

- [12] P. BOURKE. Autoregression analysis (ar). Alex Sergejew, Nick Hawthorn, Rainer Hegger., [ žiūrėta 2011-05-01 ]. <http://paulbourke.net/miscellaneous/ar/>. 9, 13, 28
- [13] Autoregressive moving average model, [ žiūrėta 2011-01-09 ]. <http://en.wikipedia.org/wiki/ARMA>. 11
- [14] J. MOCKUS. *Examples of Global and Discrete Optimisation 2*. Kluwer Academic Publishers, 2009. 11, 15
- [15] Random Author. *ARMA Theory*. Springer US, 2007. 11
- [16] Normal distribution, [ žiūrėta 2010-02-14 ]. [http://en.wikipedia.org/wiki/Normal\\_distribution](http://en.wikipedia.org/wiki/Normal_distribution). 11, 38, 45
- [17] T. MITCHELL. *Machine Learning*. McGraw-Hill, 1997. 12, 46
- [18] T.D. NIELSEN F.V. JENSEN. *Bayesian Networks and Decision Graphs*. Springer, 2007. 12, 16, 28, 46
- [19] Gaussian elimination, [ žiūrėta 2010-11-21 ]. [http://en.wikipedia.org/wiki/Gaussian\\_elimination](http://en.wikipedia.org/wiki/Gaussian_elimination). 13
- [20] E.F. FAMA. Random walks in stock market price. *Financial Analysts Journal*, 1995. 14, 15, 45, 50
- [21] J. MOCKUS. Konsultacijos ruošiamo patobulinto virtualaus akcijų biržos modelio klausimais. 2009-2011. 15, 17, 28
- [22] A. HAIDARI. Projekto analizė. virtualus finansų rinkos modelis. finansinių prognozavimų modelių programinis realizavimas, tyrimas ir taikymas naudojant dirbtinius neuroninius tinklus (ann). 2009. 16
- [23] Root mean square deviation, [ žiūrėta 2009-11-21 ]. <http://en.wikipedia.org/wiki/RMSD>. 16, 46
- [24] Mean squared error, [ žiūrėta 2011-01-09 ]. <http://en.wikipedia.org/wiki/RMSE>. 16
- [25] Approximation error, [ žiūrėta 2011-01-09 ]. [http://en.wikipedia.org/wiki/Absolute\\_error](http://en.wikipedia.org/wiki/Absolute_error). 16
- [26] Oracle. Java concurrency, [ žiūrėta 2010-02-14 ]. <http://download.oracle.com/javase/tutorial/essential/concurrency/>. 19
- [27] Openmp, [ žiūrėta 2010-02-14 ]. <http://en.wikipedia.org/wiki/OpenMP>. 20

- 
- [28] Message passing interface, [ žiūrėta 2010-02-14 ]. [http://en.wikipedia.org/wiki/Message\\_Passing\\_Interface](http://en.wikipedia.org/wiki/Message_Passing_Interface). 20
- [29] A. HAIDARI. Detalioji programinės įrangos architektūros specifikacija. virtualus finansų rinkos modelis. finansinių prognozavimų modelių programinis realizavimas, tyrimas ir taikymas naudojant dirbtinius neuroninius tinklus (ann). 2009. 25
- [30] A. HAIDARI. Reikalavimų specifikavimas. virtualus finansų rinkos modelis. finansinių prognozavimų modelių programinis realizavimas, tyrimas ir taikymas naudojant dirbtinius neuroninius tinklus (ann). 2009. 25
- [31] Design patterns, [ žiūrėta 2009-10-15 ]. [http://sourcemaking.com/design\\_patterns](http://sourcemaking.com/design_patterns). 26
- [32] Design patterns, [ žiūrėta 2009-10-15 ]. <http://www.dofactory.com/Patterns/Patterns.aspx>. 26
- [33] J. HEATON. Encog java and dotnet neural network framework, [ žiūrėta 2011-04-27 ]. <http://www.heatonresearch.com/encog>. 28
- [34] J. SANDOVAL. *RESTfull Java Web Services*. Packt, 2009. 29
- [35] Happened-before, [ žiūrėta 2010-02-15 ]. <http://en.wikipedia.org/wiki/Happened-before>. 30
- [36] J. HEATON. Programming your graphics card (gpu) with java & opencl, [ žiūrėta 2011-04-27 ]. <http://www.codeproject.com/KB/recipes/opencl-part1.aspx>. 30
- [37] Yahoo finance. Microsoft corporation (msft), [ žiūrėta 2011-05-21 ]. <http://finance.yahoo.com/q/hp?s=MSFT+Historical+Prices>.
- [38] J. MOCKUS. A small tour of optimization models, [ žiūrėta 2010-02-21 ]. <http://pilis.if.ktu.lt/~jmockus/docj/sliden.pdf>.
- [39] J. MOCKUS. On simulation of optimal strategies and nash equilibrium in the financial market context. *Journal of Global Optimization, to appear*.
- [40] J. MOCKUS. Stock exchange game model as an example for graduate level distance studies. *Computer Applications in Engineering Education*, 2003.
- [41] Standard deviation, [ žiūrėta 2011-05-20 ]. [http://en.wikipedia.org/wiki/Standard\\_deviation](http://en.wikipedia.org/wiki/Standard_deviation).
- [42] M. SNIR. Parallel programming models in comparison, [ žiūrėta 2011-02-14 ]. [http://pheattarchive.emporia.edu/projects/UPCRC/lectures/June26/062609\\_Taxonomy\\_Snir.pdf](http://pheattarchive.emporia.edu/projects/UPCRC/lectures/June26/062609_Taxonomy_Snir.pdf).





# **PADEKA**

Noriu padėkoti Gerb. prof. habil. dr. Jonui Mockui, baigiamojo magistrinio darbo vadovui, už organizacinę veiklą, siekiant įgyvendinti USEGM projekto pradinę realizaciją, už suteiktas žinias, organizuotas konsultacijas bei diskusijas įvairiais projekto klausimais.



# Skyrius 6

## PRIEDAI

### 6.1 Priedas 1. Projekto pasiekiamumas

**USEGM projektą galima pasiekti šiais adresais:**

- <http://usegm.alisher.net> arba
- <http://soften.ktu.lt/~mockus/>.

**Projekto SVN (*Subversion*):**

- <http://svn.alisher.net/repos/usegm>

**Dėl prisijungimo vardų ir slaptažodžių kreiptis adresais:**

- aliser /at/ ymail /./ com – (Ališer Haidari, šio darbo autorius) arba;
- jmockus /at/ gmail /./ com – (prof. habil. dr. Jonas Mockus, darbo kuratorius).

Šiuo metu projektas yra aktyviai vystomas. USEGM - tai mokslinė tiriamoji programa, kuri yra atviro kodo ir vėlesniame etape bus perkelta į GIT versijų sistemą, kas leis kiekvienam norinčiam vartotojui tirti ar plėtoti projektą pagal asmenines reikmes.

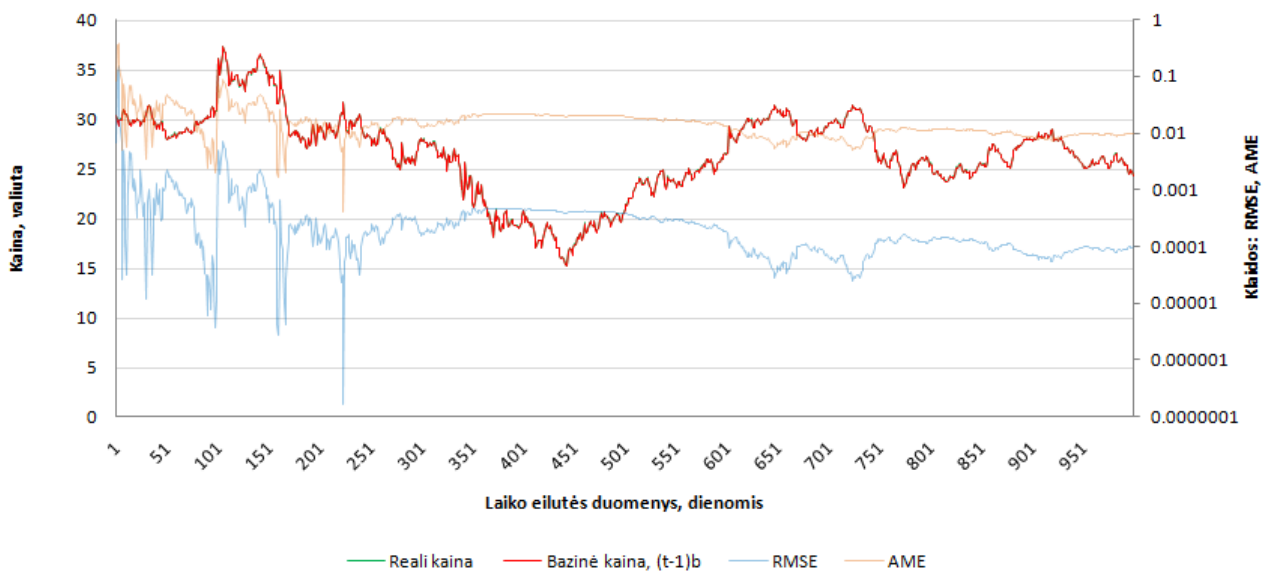
Laukiami pasiūlymai, pastebėjimai bei pageidavimai siekiant tobulinti įgyvendinamą USEGM modelį. Daugiau informacijos rasite adresu <http://usegm.alisher.net>.

**SEGM projektas:** Šio projekto realizaciją galite pasiekti adresu <http://soften.ktu.lt/~mockus/stockedvinas/contstockedvinas.html>.

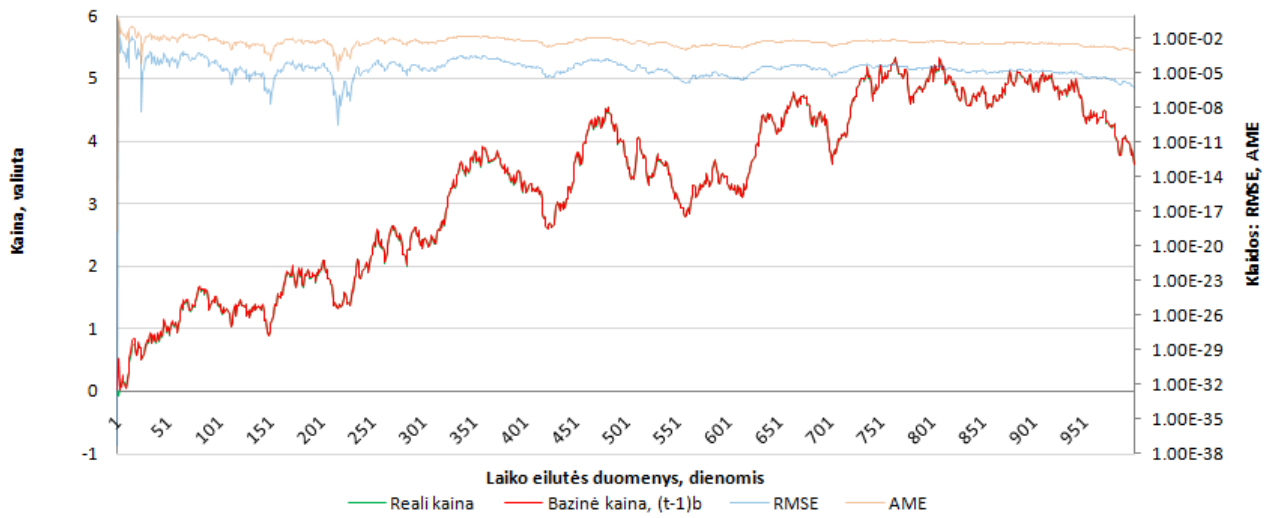
## 6.2 Priedas 2. Papildomi prognozių tyrimo grafiniai rezultatai

Papildomus prognozių tyrimo rezultatus rasite USEGM svetainėje *results* skyrelyje <http://usegm.alisher.net/results/>. Duomenų ir rezultatų failai pateikiami CSV formate. Testavimo atvejų šiuo metu yra ~300. Įvairius sudarytų prognozių skaičiavimus galima rasti adresu <http://usegm.alisher.net/predictions.xls>.

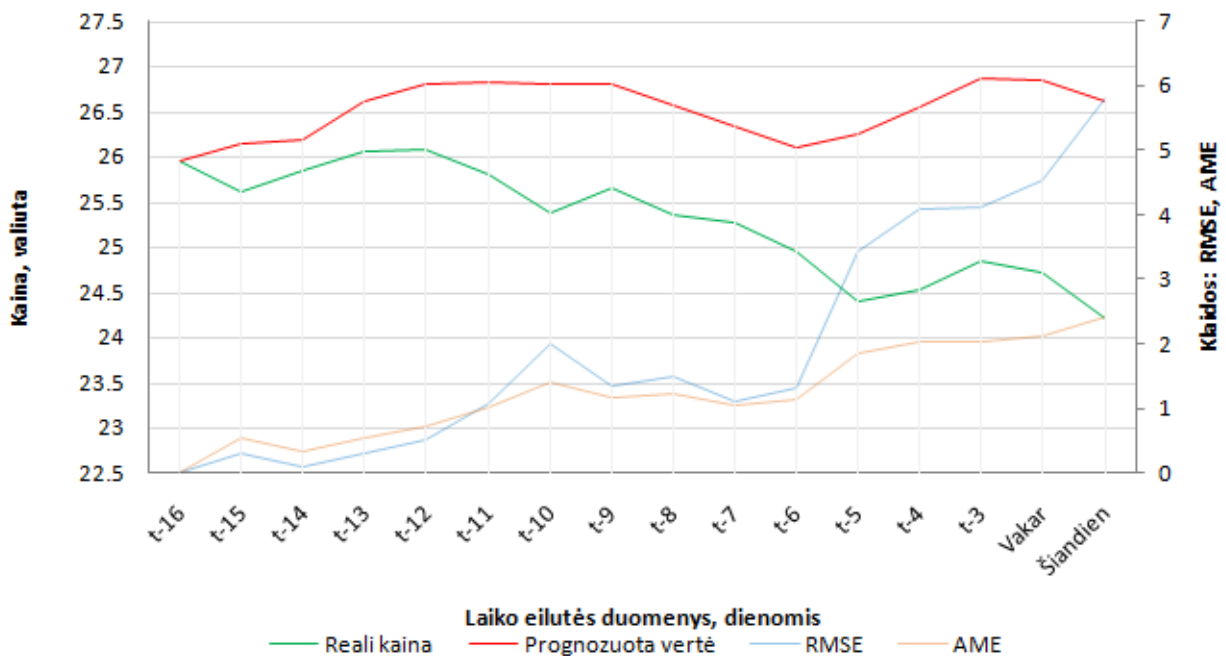
Pateikiami 8 papildomo tyrimo rezultatai, kurie parodo atitinkamo prognozavimo algoritmo charakteristikas naudojant 1000 laiko eilutės reikšmių optimizavimui, gautų iš *finance.yahoo.com* portalo [9] ir tiriant *Microsoft Corporation (MSFT)* akcijų kainų vertės koeficientų sudarymo tikslumą bei prognozavimą (1-ąjį bei 2-ąjį etapus).



6.1 pav.:  $AR(1)$ , parametrų optimizavimas, *MSFT*,  $RMSE = 0.025502938$

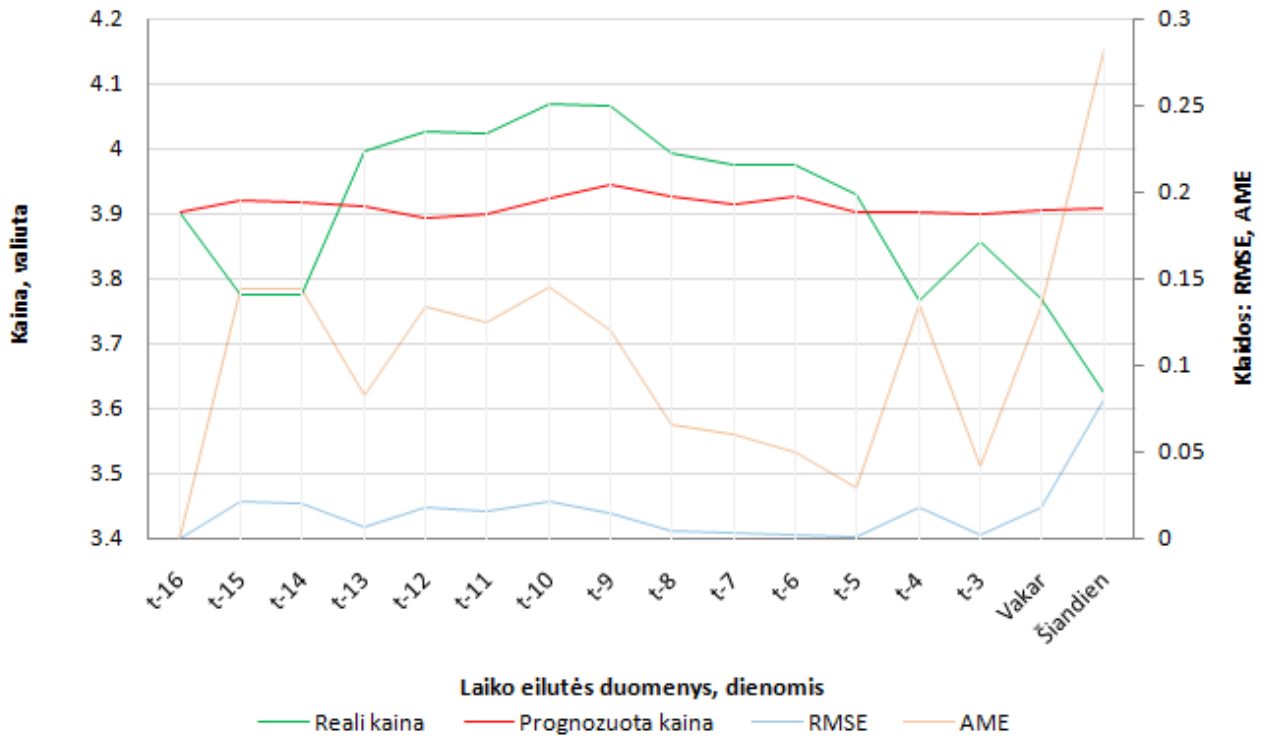


6.2 pav.:  $AR(1)$ , parametų optimizavimas, sugeneruoti duom., RMSE = 0.109073974

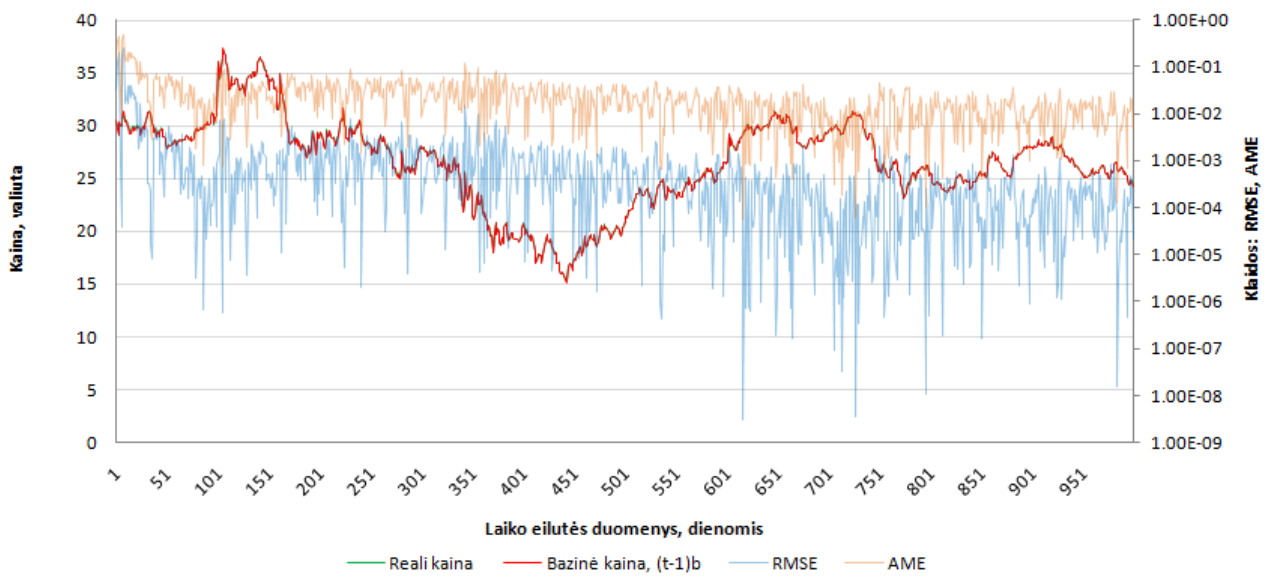


6.3 pav.:  $AR(1)$ , prognozė, MSFT, RMSE = 5.615557442

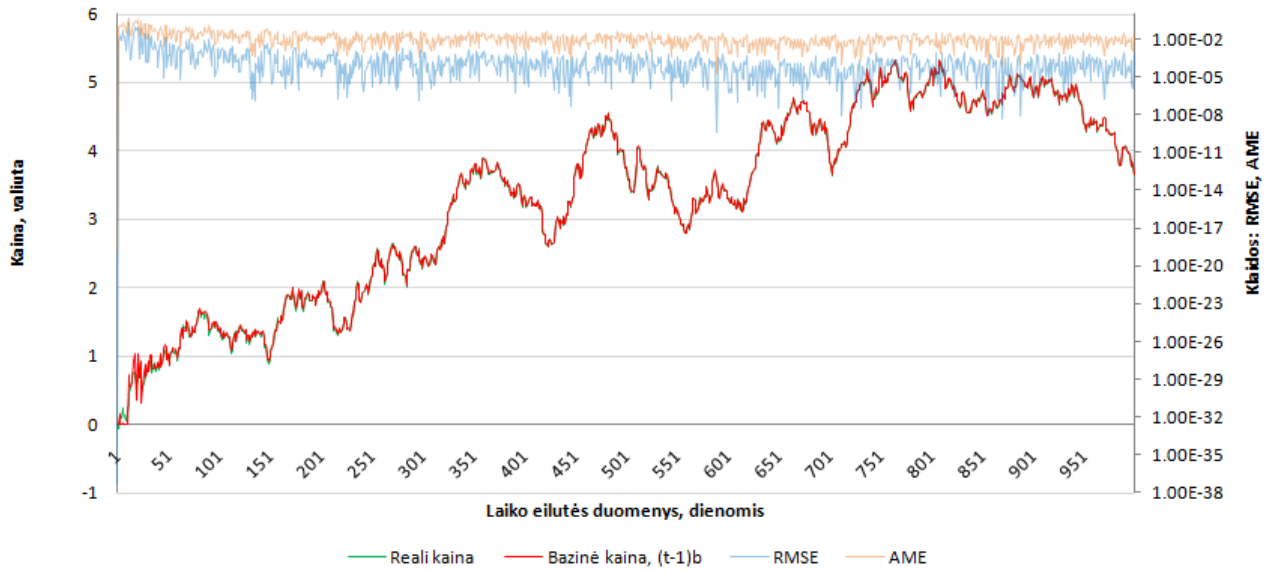
6.2. Priedas 2. Papildomi prognozių tyrimo grafiniai rezultatai



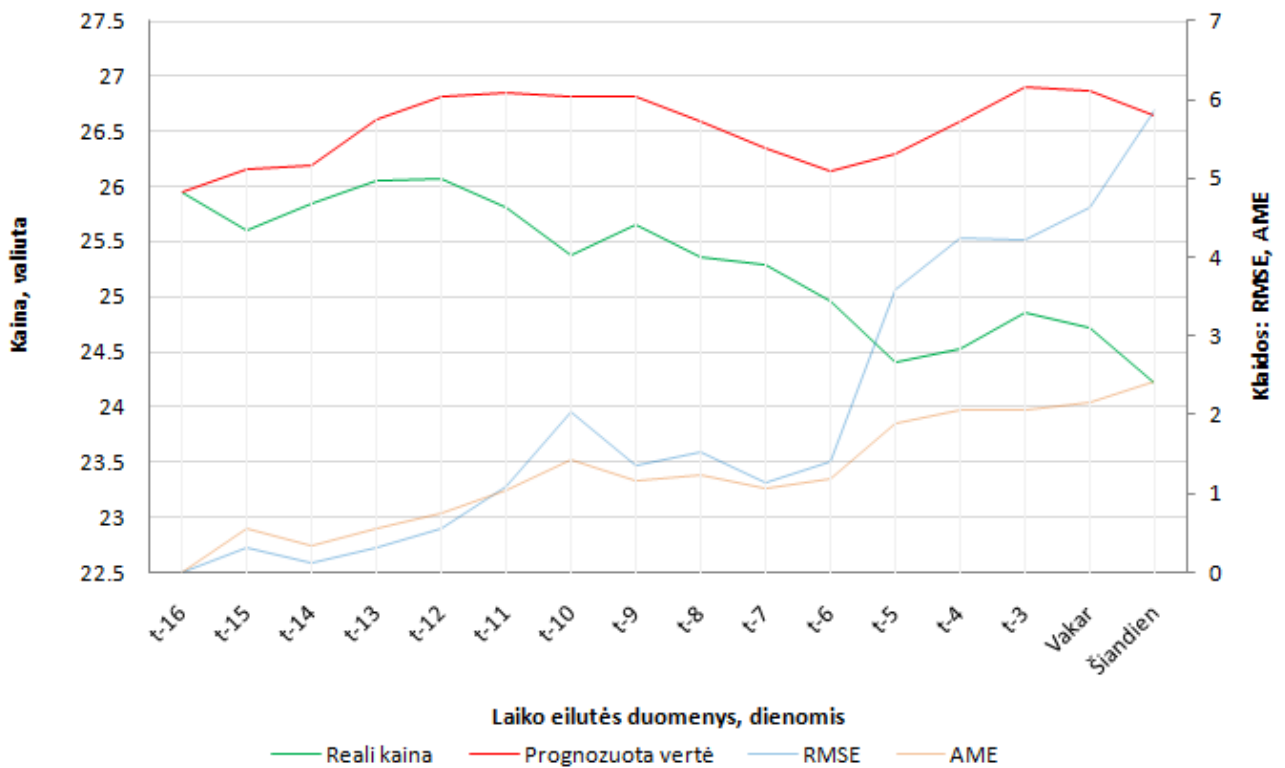
6.4 pav.:  $AR(1)$ , prognozė, sugeneruoti duom.,  $RMSE = 0.496580846$



6.5 pav.:  $AR-ABS(8)$ , parametų optimizavimas,  $MSFT$ ,  $RMSE = 0.043231444$

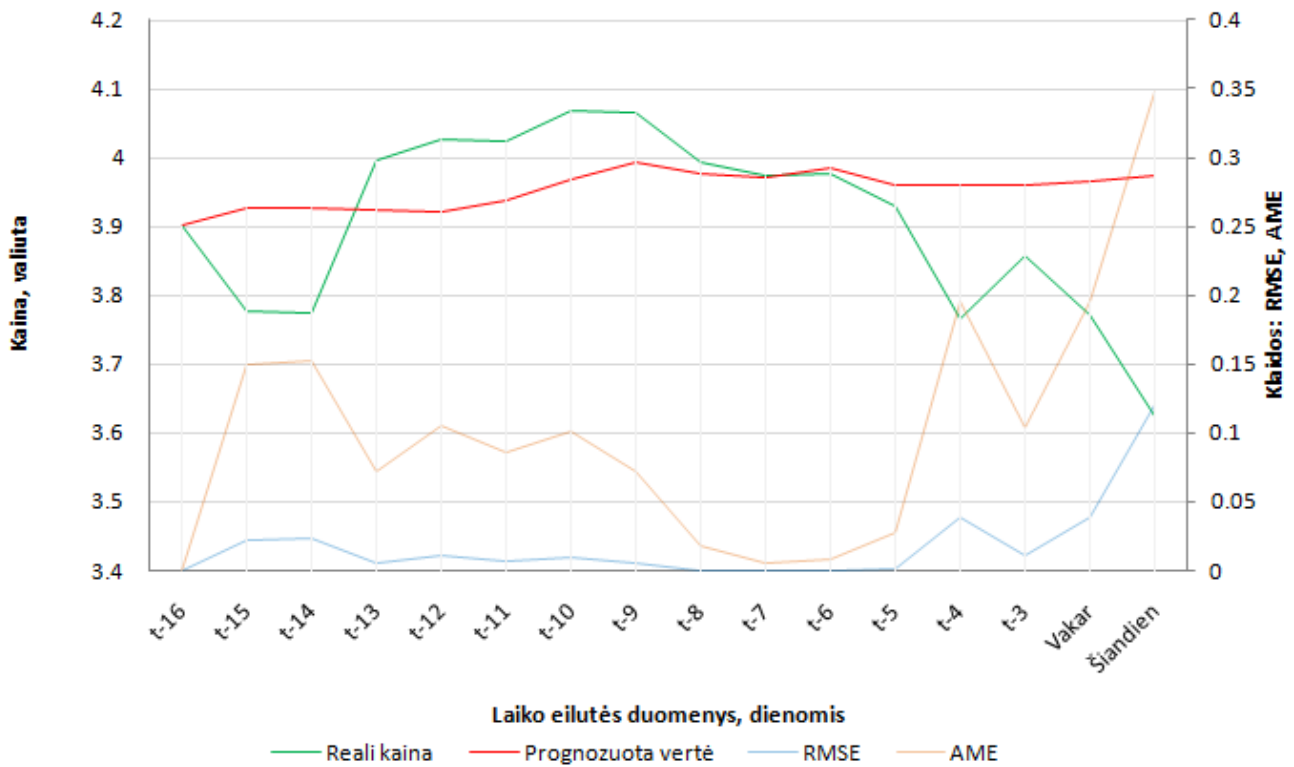


6.6 pav.: AR-ABS(8), parametų optimizavimas, sugeneruoti duom., RMSE = 0.038376583



6.7 pav.: AR-ABS(8), prognozė, MSFT, RMSE = 0.038376583





6.8 pav.: AR-ABS(8), prognozė, sugeneruoti duom., RMSE = 0.038376583

### 6.3 Priedas 3. USEGM generuojamos skaičiavimų išvestys

Ryšium su tuo, kad įgyvendinti matematiniai algoritmai yra sudėtingi, USEGM aplikacija generuoja tarpines prognozės skaičiavimų išvestis, kas palengvina nustatyti, ar teisingas reikšmes generuoja algoritmas prie tam tikrų duomenų imčių.

Kaip matoma iš žemiau pateiktų išvesčių pav. 6.3 ir 6.3, prie tam tikro algoritmo reikšmių išvedimo matomi tam tikri žingsniai, kuriuos aplikacija vykdo kiekvienoje prognozavimo iteracijoje. Tai - architektūrinis sprendimas ir, taipogi, karkasas vėlesniems darbams kuriant USEGM akcijų pirkimo-pardavimo ir kitas logikas. Taipogi aiškiai matomi koeficientų sudarymo žingsniai bei gauti koeficientai. Tai itin svarbūs modeliavimo parametrai.

```
----- Iteration: #6 -----
Step 1: initialization
  Epsilon [error rate, random]: -0.01136965371820997

Step 2: Opening price evaluation
  Yesterday closing price: [ t - 1 ]: 5.184493209155996
  Today opening price: [ t ]: 5.184493209155996

Step 3: Tomorrow [ t + 1 ] opening price prediction
  Customer #0 -> strategy: AR; prediction: 5.486459178150971; memory [ p ]: 3.
    Historical prices (given data for prediction):
      [ 5.0 ],
      [ 5.0 ],
      [ 5.018979035316955 ],
      [ 5.115244866430133 ],
      [ 5.303743796593958 ],
      [ 5.223126454896565 ],
      [ 5.184493209155996 ].

    Calculated matrix (AR):
      [ Coeffs. matrix ] | [ Free variables ]
      [ 26.6917, 26.4001, 26.0020. ] | [ 26.89616078586778 ]
      [ 26.4001, 26.1214, 25.7245. ] | [ 26.602599434404443 ]
      [ 26.0020, 25.7245, 25.3390. ] | [ 26.207414441058194 ]

    Solved matrix (using Gauss method):
      [ Coeffs. matrix ] | [ Free variables ]
      [ 26.6917, 26.4001, 26.0020. ] | [ 1.238409201033436E-4 ]
      [ 0.0000, 52.2331, 51.4425. ] | [ 3.443245768912905E-5 ]
      [ 0.0000, 0.0000, 101.333. ] | [ 1.0342952998943633 ]

Step 6: Closing price evaluation
  Has transactions: no
  Today closing price [ t ]: 5.173123555437786
```

6.9 pav.: AR modelio skaičiavimo išvestys

6.3. Priedas 3. USEGM generuojamos skaičiavimų išvestys

```
----- Iteration: #9 -----
Step 1: initialization
  Epsilon [error rate, random]: 0.024290561553593545

Step 2: Opening price evaluation
  Yesterday closing price: [ t - 1 ]: 4.938265162924155
  Today opening price: [ t ]: 4.938265162924155

Step 3: Tomorrow [ t + 1 ] opening price prediction
  Customer #0 -> strategy: ARABS; prediction: 4.958934547138592; memory [ p ]: 2.
  Historical prices (given data for prediction):
    [ 5.0 ],
    [ 5.0 ],
    [ 4.89003895635174 ],
    [ 4.974071878766442 ],
    [ 4.920629151618812 ],
    [ 5.003690493107057 ],
    [ 4.983817099169372 ],
    [ 5.019361589712707 ],
    [ 4.917166198802457 ],
    [ 4.938265162924155 ].
  lprec(ncolumns) length: 14
  minf, rib1, rib2 lengths: 15
  rib1 values:
    [ 0.0 ],
    < ... >,
    [ 1.0 ],
    [ 4.917166198802457 ],
    [ 5.019361589712707 ],
    [ -4.917166198802457 ],
    [ -5.019361589712707 ].
  rib2 values:
    [ 0.0 ],
    < ... >,
    [ -1.0 ],
    [ 4.917166198802457 ],
    [ 5.019361589712707 ],
    [ -4.917166198802457 ],
    [ -5.019361589712707 ].
  Solution: 0 (optimal, coeffs. found)
  Rows: 20
  minf values (best solution):
    [ 0.0 ],
    [ 5.0 ],
    [ -2.133709875451472E-16 ],
    [ 0.13097862102796165 ],
    [ 0.0630153450350005 ],
    [ 0.0739980815786059 ],
    [ 0.06215275336741103 ],
    [ 0.0405573347278771 ],
    [ 0.014511400118742931 ],
    [ 0.12314494321586462 ],
    [ -2.2724877535296173E-16 ],
    [ 0.99 ],
    [ -0.005796484524059691 ],
    [ -0.01 ],
    [ -0.01 ].
  coeffs:
    [ 1 ],
    [ 0.0042035154759403094149039503690801211632788181304931640625 ],
    [ 0 ].

Step 6: Closing price evaluation
  Has transactions: no
  Today closing price [ t ]: 4.962555724477749
```

6.10 pav.: AR-ABS modelio skaičiavimo išvestys

## 6.4 Priedas 4. Papildomi optimizavimo rezultatai

Pateikiami apibendrinantys tiriamus modelius rezultatai. Šie rezultatai buvo gauti vykdant testinius atvejus prognozavimo posistemėje, optimizuojant laiko eilutę pasirinktu modeliu su tikslu rasti  $a$  parametrus ir rodo bendrą tendenciją, kuri priklauso nuo pasirinkto  $p$  parametro dydžio bei naudojamo modelio. Optimizuojamose laiko eilutėse naudojami *Microsoft Corporation (MSFT)*, *Apple Inc. (AAPL)* ir sugeneruoti USEGM duomenys -  $T2$ .

Daugiau rezultatų rasite priede 6.2 puslapyje 58 bei jo nuorodose.

6.4. Priedas 4. Papildomi optimizavimo rezultatai

Raudonas tekstas - didžiausia reikšmė  
 Žalias tekstas - mažiausia reikšmė

Strategija	Metodas, AR	p	Duomenys (online)	Rezultatai (online)	RMSE	AME
2	AR	1	data_MSFT_1k.csv	res_AR_p1_MSFT_1k.csv	0.025502938	0.016720022
3	AR	2	data_MSFT_1k.csv	res_AR_p2_MSFT_1k.csv	0.506020592	0.35291967
4	AR	3	data_MSFT_1k.csv	res_AR_p3_MSFT_1k.csv	0.707749997	0.507593106
5	AR	4	data_MSFT_1k.csv	res_AR_p4_MSFT_1k.csv	0.844083311	0.62293036
6	AR	5	data_MSFT_1k.csv	res_AR_p5_MSFT_1k.csv	0.953069115	0.707640176
7	AR	6	data_MSFT_1k.csv	res_AR_p6_MSFT_1k.csv	1.054080106	0.786827052
8	AR	7	data_MSFT_1k.csv	res_AR_p7_MSFT_1k.csv	1.148062863	0.868307492
9	AR	8	data_MSFT_1k.csv	res_AR_p8_MSFT_1k.csv	1.2326772	0.931246933
10	AR	9	data_MSFT_1k.csv	res_AR_p9_MSFT_1k.csv	1.305003767	0.987625184
11	AR	10	data_MSFT_1k.csv	res_AR_p10_MSFT_1k.csv	1.358142385	1.031472258
12	AR	11	data_MSFT_1k.csv	res_AR_p11_MSFT_1k.csv	1.417260047	1.084727442
13	AR	12	data_MSFT_1k.csv	res_AR_p12_MSFT_1k.csv	1.47943156	1.138333847
14	AR	13	data_MSFT_1k.csv	res_AR_p13_MSFT_1k.csv	1.544295769	1.190770072
15	AR	14	data_MSFT_1k.csv	res_AR_p14_MSFT_1k.csv	1.610188665	1.242306847
16	AR	15	data_MSFT_1k.csv	res_AR_p15_MSFT_1k.csv	1.669292662	1.292307534
					<b>4.105467206</b>	<b>0.850781866</b>
17	AR	1	data_MSFT_115.csv	res_AR_p1_MSFT_115.csv	0.142328157	0.053238372
18	AR	2	data_MSFT_115.csv	res_AR_p2_MSFT_115.csv	0.283597597	0.218709413
19	AR	3	data_MSFT_115.csv	res_AR_p3_MSFT_115.csv	0.393386203	0.308563799
20	AR	4	data_MSFT_115.csv	res_AR_p4_MSFT_115.csv	0.45376615	0.368182963
21	AR	5	data_MSFT_115.csv	res_AR_p5_MSFT_115.csv	0.486489951	0.399691465
22	AR	6	data_MSFT_115.csv	res_AR_p6_MSFT_115.csv	0.538162845	0.441423235
23	AR	7	data_MSFT_115.csv	res_AR_p7_MSFT_115.csv	0.589348292	0.480909131
24	AR	8	data_MSFT_115.csv	res_AR_p8_MSFT_115.csv	0.632451085	0.510755349
25	AR	9	data_MSFT_115.csv	res_AR_p9_MSFT_115.csv	0.654522522	0.537321788
26	AR	10	data_MSFT_115.csv	res_AR_p10_MSFT_115.csv	0.668134712	0.573686638
27	AR	11	data_MSFT_115.csv	res_AR_p11_MSFT_115.csv	0.689558017	0.587245584
28	AR	12	data_MSFT_115.csv	res_AR_p12_MSFT_115.csv	0.726763451	0.600071375
29	AR	13	data_MSFT_115.csv	res_AR_p13_MSFT_115.csv	0.752849916	0.61011975
30	AR	14	data_MSFT_115.csv	res_AR_p14_MSFT_115.csv	0.779589462	0.619707617
31	AR	15	data_MSFT_115.csv	res_AR_p15_MSFT_115.csv	0.800591481	0.635683885
					<b>2.93113286</b>	<b>0.463020691</b>
32	AR	1	data_AAPL_1k.csv	res_AR_p1_AAPL_1k.csv	0.289599932	0.206414363
33	AR	2	data_AAPL_1k.csv	res_AR_p2_AAPL_1k.csv	4.432129714	3.269480498
34	AR	3	data_AAPL_1k.csv	res_AR_p3_AAPL_1k.csv	5.852756533	4.396571137
35	AR	4	data_AAPL_1k.csv	res_AR_p4_AAPL_1k.csv	7.103706255	5.439266963
36	AR	5	data_AAPL_1k.csv	res_AR_p5_AAPL_1k.csv	8.157298684	6.306614179
37	AR	6	data_AAPL_1k.csv	res_AR_p6_AAPL_1k.csv	9.258168884	7.184670493
38	AR	7	data_AAPL_1k.csv	res_AR_p7_AAPL_1k.csv	10.21774352	7.96783148
39	AR	8	data_AAPL_1k.csv	res_AR_p8_AAPL_1k.csv	11.04276718	8.653313722
40	AR	9	data_AAPL_1k.csv	res_AR_p9_AAPL_1k.csv	11.74432152	9.193118517
41	AR	10	data_AAPL_1k.csv	res_AR_p10_AAPL_1k.csv	12.36812625	9.668390381

6.11 pav.: Parametų optimizavimo statistika. 1-5

42	AR	11	data_AAPL_1k.csv	res_AR_p11_AAPL_1k.csv	12.9735523	10.04374206
43	AR	12	data_AAPL_1k.csv	res_AR_p12_AAPL_1k.csv	13.57249826	10.40619267
44	AR	13	data_AAPL_1k.csv	res_AR_p13_AAPL_1k.csv	14.19279152	10.81303691
45	AR	14	data_AAPL_1k.csv	res_AR_p14_AAPL_1k.csv	14.891744	11.29885849
46	AR	15	data_AAPL_1k.csv	res_AR_p15_AAPL_1k.csv	15.52885523	11.72369728
					12.31365339	7.771413277
47	AR	1	data_AAPL_115.csv	res_AR_p1_AAPL_115.csv	0.672977092	0.547118367
48	AR	2	data_AAPL_115.csv	res_AR_p2_AAPL_115.csv	5.342750204	3.917306103
49	AR	3	data_AAPL_115.csv	res_AR_p3_AAPL_115.csv	6.763483448	4.785696771
50	AR	4	data_AAPL_115.csv	res_AR_p4_AAPL_115.csv	7.955630298	5.846147984
51	AR	5	data_AAPL_115.csv	res_AR_p5_AAPL_115.csv	8.74037543	6.486696271
52	AR	6	data_AAPL_115.csv	res_AR_p6_AAPL_115.csv	9.533150536	7.433135746
53	AR	7	data_AAPL_115.csv	res_AR_p7_AAPL_115.csv	10.42114677	8.307799842
54	AR	8	data_AAPL_115.csv	res_AR_p8_AAPL_115.csv	11.11364098	8.992275516
55	AR	9	data_AAPL_115.csv	res_AR_p9_AAPL_115.csv	11.47824609	9.158286615
56	AR	10	data_AAPL_115.csv	res_AR_p10_AAPL_115.csv	11.44259663	9.270190126
57	AR	11	data_AAPL_115.csv	res_AR_p11_AAPL_115.csv	11.47673993	9.103459272
58	AR	12	data_AAPL_115.csv	res_AR_p12_AAPL_115.csv	11.65399017	9.326241949
59	AR	13	data_AAPL_115.csv	res_AR_p13_AAPL_115.csv	11.53001254	9.231389613
60	AR	14	data_AAPL_115.csv	res_AR_p14_AAPL_115.csv	11.52637225	9.350867137
61	AR	15	data_AAPL_115.csv	res_AR_p15_AAPL_115.csv	11.42932657	9.130582546
					11.87772869	7.39247959
62	AR	1	data_T2.csv	res_AR_p1_T2.csv	0.109073974	0.016556785
63	AR	2	data_T2.csv	res_AR_p2_T2.csv	0.092216933	0.073106204
64	AR	3	data_T2.csv	res_AR_p3_T2.csv	0.129279182	0.101635798
65	AR	4	data_T2.csv	res_AR_p4_T2.csv	0.159467736	0.12469321
66	AR	5	data_T2.csv	res_AR_p5_T2.csv	0.19239211	0.150525375
67	AR	6	data_T2.csv	res_AR_p6_T2.csv	0.218514122	0.169201736
68	AR	7	data_T2.csv	res_AR_p7_T2.csv	0.73350391	0.210299424
69	AR	8	data_T2.csv	res_AR_p8_T2.csv	0.266106533	0.207112163
70	AR	9	data_T2.csv	res_AR_p9_T2.csv	0.309948895	0.226758916
71	AR	10	data_T2.csv	res_AR_p10_T2.csv	0.304625148	0.23872141
72	AR	11	data_T2.csv	res_AR_p11_T2.csv	0.325083654	0.254490169
73	AR	12	data_T2.csv	res_AR_p12_T2.csv	0.344766642	0.269900988
74	AR	13	data_T2.csv	res_AR_p13_T2.csv	0.369159737	0.286662578
75	AR	14	data_T2.csv	res_AR_p14_T2.csv	0.387692934	0.299599853
76	AR	15	data_T2.csv	res_AR_p15_T2.csv	0.407308903	0.313925355
					2.085459281	0.196212664
77	AR	1	data_T2_115.csv	res_AR_p1_T2_115.csv	0.061414359	0.011986846
78	AR	2	data_T2_115.csv	res_AR_p2_T2_115.csv	0.091793486	0.069749007
79	AR	3	data_T2_115.csv	res_AR_p3_T2_115.csv	0.121171728	0.089220785
80	AR	4	data_T2_115.csv	res_AR_p4_T2_115.csv	0.129379473	0.101547299
81	AR	5	data_T2_115.csv	res_AR_p5_T2_115.csv	0.148419835	0.111982344
82	AR	6	data_T2_115.csv	res_AR_p6_T2_115.csv	0.159835643	0.119144002
83	AR	7	data_T2_115.csv	res_AR_p7_T2_115.csv	0.168532014	0.127408066
84	AR	8	data_T2_115.csv	res_AR_p8_T2_115.csv	0.178798064	0.137256939
85	AR	9	data_T2_115.csv	res_AR_p9_T2_115.csv	0.1832935	0.141881491

6.12 pav.: Parametrų optimizavimo statistika. 2-5

6.4. Priedas 4. Papildomi optimizavimo rezultatai

86	AR	10	data_T2_115.csv	res_AR_p10_T2_115.csv	0.189802447	0.150156852
87	AR	11	data_T2_115.csv	res_AR_p11_T2_115.csv	0.194691878	0.152343952
88	AR	12	data_T2_115.csv	res_AR_p12_T2_115.csv	0.194872387	0.153489465
89	AR	13	data_T2_115.csv	res_AR_p13_T2_115.csv	0.20952242	0.16176848
90	AR	14	data_T2_115.csv	res_AR_p14_T2_115.csv	0.203680878	0.156087332
91	AR	15	data_T2_115.csv	res_AR_p15_T2_115.csv	0.219252935	0.171278594
					1.566671965	0.123686764
Vykdymo laikas: 00:06:21 h						
Strategija	Metodas, AR-ABS	p	Duomenys (online)	Rezultatai (online)	RMSE	AME
2	AR-ABS	1	data_MSFT_1k.csv	res_AR-ABS_p1_MSFT_1k.csv	0.960453833	0.045282395
3	AR-ABS	2	data_MSFT_1k.csv	res_AR-ABS_p2_MSFT_1k.csv	0.043660588	0.01501201
4	AR-ABS	3	data_MSFT_1k.csv	res_AR-ABS_p3_MSFT_1k.csv	0.035610751	0.015994466
5	AR-ABS	4	data_MSFT_1k.csv	res_AR-ABS_p4_MSFT_1k.csv	0.028691054	0.015603356
6	AR-ABS	5	data_MSFT_1k.csv	res_AR-ABS_p5_MSFT_1k.csv	0.031792974	0.016783659
7	AR-ABS	6	data_MSFT_1k.csv	res_AR-ABS_p6_MSFT_1k.csv	0.038224685	0.018565695
8	AR-ABS	7	data_MSFT_1k.csv	res_AR-ABS_p7_MSFT_1k.csv	0.043231444	0.022583387
9	AR-ABS	8	data_MSFT_1k.csv	res_AR-ABS_p8_MSFT_1k.csv	0.0455992	0.026195992
10	AR-ABS	9	data_MSFT_1k.csv	res_AR-ABS_p9_MSFT_1k.csv	0.042770475	0.023859772
11	AR-ABS	10	data_MSFT_1k.csv	res_AR-ABS_p10_MSFT_1k.csv	0.045545548	0.023400871
12	AR-ABS	11	data_MSFT_1k.csv	res_AR-ABS_p11_MSFT_1k.csv	0.047693194	0.023117848
13	AR-ABS	12	data_MSFT_1k.csv	res_AR-ABS_p12_MSFT_1k.csv	0.043358173	0.022860869
14	AR-ABS	13	data_MSFT_1k.csv	res_AR-ABS_p13_MSFT_1k.csv	0.051812709	0.026015703
15	AR-ABS	14	data_MSFT_1k.csv	res_AR-ABS_p14_MSFT_1k.csv	0.048783915	0.025768625
16	AR-ABS	15	data_MSFT_1k.csv	res_AR-ABS_p15_MSFT_1k.csv	0.050126011	0.026880635
					1.247940124	0.023195019
17	AR-ABS	1	data_MSFT_115.csv	res_AR-ABS_p1_MSFT_115.csv	2.501697903	0.263304069
18	AR-ABS	2	data_MSFT_115.csv	res_AR-ABS_p2_MSFT_115.csv	0.157367361	0.031003515
19	AR-ABS	3	data_MSFT_115.csv	res_AR-ABS_p3_MSFT_115.csv	0.07515027	0.036014898
20	AR-ABS	4	data_MSFT_115.csv	res_AR-ABS_p4_MSFT_115.csv	0.073160804	0.035763417
21	AR-ABS	5	data_MSFT_115.csv	res_AR-ABS_p5_MSFT_115.csv	0.077802295	0.037967317
22	AR-ABS	6	data_MSFT_115.csv	res_AR-ABS_p6_MSFT_115.csv	0.072355405	0.035848107
23	AR-ABS	7	data_MSFT_115.csv	res_AR-ABS_p7_MSFT_115.csv	0.072119591	0.034050837
24	AR-ABS	8	data_MSFT_115.csv	res_AR-ABS_p8_MSFT_115.csv	0.067658551	0.035920752
25	AR-ABS	9	data_MSFT_115.csv	res_AR-ABS_p9_MSFT_115.csv	0.071188217	0.040957162
26	AR-ABS	10	data_MSFT_115.csv	res_AR-ABS_p10_MSFT_115.csv	0.07161676	0.03936033
27	AR-ABS	11	data_MSFT_115.csv	res_AR-ABS_p11_MSFT_115.csv	0.070942216	0.041645305
28	AR-ABS	12	data_MSFT_115.csv	res_AR-ABS_p12_MSFT_115.csv	0.075249465	0.04214052
29	AR-ABS	13	data_MSFT_115.csv	res_AR-ABS_p13_MSFT_115.csv	0.070883332	0.040940948
30	AR-ABS	14	data_MSFT_115.csv	res_AR-ABS_p14_MSFT_115.csv	0.07157106	0.039230217
31	AR-ABS	15	data_MSFT_115.csv	res_AR-ABS_p15_MSFT_115.csv	0.074329407	0.042164251
					1.898181403	0.053087443
32	AR-ABS	1	data_AAPL_1k.csv	res_AR-ABS_p1_AAPL_1k.csv	3.901134618	0.582164437

6.13 pav.: Parametų optimizavimo statistika. 3-5



33	AR-ABS	2	data_AAPL_1k.csv	res_AR-ABS_p2_AAPL_1k.csv	0.528084099	0.460228279
34	AR-ABS	3	data_AAPL_1k.csv	res_AR-ABS_p3_AAPL_1k.csv	0.582910202	0.470593196
35	AR-ABS	4	data_AAPL_1k.csv	res_AR-ABS_p4_AAPL_1k.csv	0.616258339	0.496135692
36	AR-ABS	5	data_AAPL_1k.csv	res_AR-ABS_p5_AAPL_1k.csv	0.648024589	0.513686867
37	AR-ABS	6	data_AAPL_1k.csv	res_AR-ABS_p6_AAPL_1k.csv	0.643053374	0.504909817
38	AR-ABS	7	data_AAPL_1k.csv	res_AR-ABS_p7_AAPL_1k.csv	0.708557381	0.557924838
39	AR-ABS	8	data_AAPL_1k.csv	res_AR-ABS_p8_AAPL_1k.csv	0.689644584	0.545776004
40	AR-ABS	9	data_AAPL_1k.csv	res_AR-ABS_p9_AAPL_1k.csv	0.667512025	0.520234311
41	AR-ABS	10	data_AAPL_1k.csv	res_AR-ABS_p10_AAPL_1k.csv	0.674473043	0.521320652
42	AR-ABS	11	data_AAPL_1k.csv	res_AR-ABS_p11_AAPL_1k.csv	0.677193065	0.525815298
43	AR-ABS	12	data_AAPL_1k.csv	res_AR-ABS_p12_AAPL_1k.csv	0.701454414	0.548850414
44	AR-ABS	13	data_AAPL_1k.csv	res_AR-ABS_p13_AAPL_1k.csv	0.678227972	0.520741298
45	AR-ABS	14	data_AAPL_1k.csv	res_AR-ABS_p14_AAPL_1k.csv	0.682273981	0.518133538
46	AR-ABS	15	data_AAPL_1k.csv	res_AR-ABS_p15_AAPL_1k.csv	0.696859156	0.532337313
					<b>3.618792733</b>	<b>0.521256797</b>
2	AR-ABS	1	data_AAPL_115.csv	res_AR-ABS_p1_AAPL_115.csv	29.8181008	3.565426993
3	AR-ABS	2	data_AAPL_115.csv	res_AR-ABS_p2_AAPL_115.csv	1.073207783	0.802523785
4	AR-ABS	3	data_AAPL_115.csv	res_AR-ABS_p3_AAPL_115.csv	0.997018903	0.837549307
5	AR-ABS	4	data_AAPL_115.csv	res_AR-ABS_p4_AAPL_115.csv	0.968878411	0.782844854
6	AR-ABS	5	data_AAPL_115.csv	res_AR-ABS_p5_AAPL_115.csv	1.006695009	0.851497542
7	AR-ABS	6	data_AAPL_115.csv	res_AR-ABS_p6_AAPL_115.csv	1.03108507	0.885832413
8	AR-ABS	7	data_AAPL_115.csv	res_AR-ABS_p7_AAPL_115.csv	1.032595851	0.862098638
9	AR-ABS	8	data_AAPL_115.csv	res_AR-ABS_p8_AAPL_115.csv	1.104567955	0.935128769
10	AR-ABS	9	data_AAPL_115.csv	res_AR-ABS_p9_AAPL_115.csv	1.218163994	1.069366981
11	AR-ABS	10	data_AAPL_115.csv	res_AR-ABS_p10_AAPL_115.csv	1.124404414	0.965642711
12	AR-ABS	11	data_AAPL_115.csv	res_AR-ABS_p11_AAPL_115.csv	1.041680514	0.84499857
13	AR-ABS	12	data_AAPL_115.csv	res_AR-ABS_p12_AAPL_115.csv	1.184734793	0.972831036
14	AR-ABS	13	data_AAPL_115.csv	res_AR-ABS_p13_AAPL_115.csv	1.240915865	1.018615059
15	AR-ABS	14	data_AAPL_115.csv	res_AR-ABS_p14_AAPL_115.csv	1.180737304	0.94967958
16	AR-ABS	15	data_AAPL_115.csv	res_AR-ABS_p15_AAPL_115.csv	1.356488278	1.135877519
					<b>6.736414102</b>	<b>1.098660917</b>
17	AR-ABS	1	data_T2.csv	res_AR-ABS_p1_T2.csv	0.043015193	0.009174549
18	AR-ABS	2	data_T2.csv	res_AR-ABS_p2_T2.csv	0.102519944	0.012567743
19	AR-ABS	3	data_T2.csv	res_AR-ABS_p3_T2.csv	0.609397278	0.033581918
20	AR-ABS	4	data_T2.csv	res_AR-ABS_p4_T2.csv	3.790474586	0.149345947
21	AR-ABS	5	data_T2.csv	res_AR-ABS_p5_T2.csv	23.44554981	0.864269652
22	AR-ABS	6	data_T2.csv	res_AR-ABS_p6_T2.csv	23.14535527	0.747686898
23	AR-ABS	7	data_T2.csv	res_AR-ABS_p7_T2.csv	0.161339177	0.018069044
24	AR-ABS	8	data_T2.csv	res_AR-ABS_p8_T2.csv	0.038376583	0.017217734
25	AR-ABS	9	data_T2.csv	res_AR-ABS_p9_T2.csv	0.104277354	0.022054597
26	AR-ABS	10	data_T2.csv	res_AR-ABS_p10_T2.csv	10.60129633	0.357905841
27	AR-ABS	11	data_T2.csv	res_AR-ABS_p11_T2.csv	0.235748616	0.030343067
28	AR-ABS	12	data_T2.csv	res_AR-ABS_p12_T2.csv	0.065759276	0.024294714
29	AR-ABS	13	data_T2.csv	res_AR-ABS_p13_T2.csv	1000.356611	100.1030678
30	AR-ABS	14	data_T2.csv	res_AR-ABS_p14_T2.csv	1517.071168	230.2050536
31	AR-ABS	15	data_T2.csv	res_AR-ABS_p15_T2.csv	3317.898095	1101.758705
					<b>76.79628236</b>	<b>95.62355584</b>

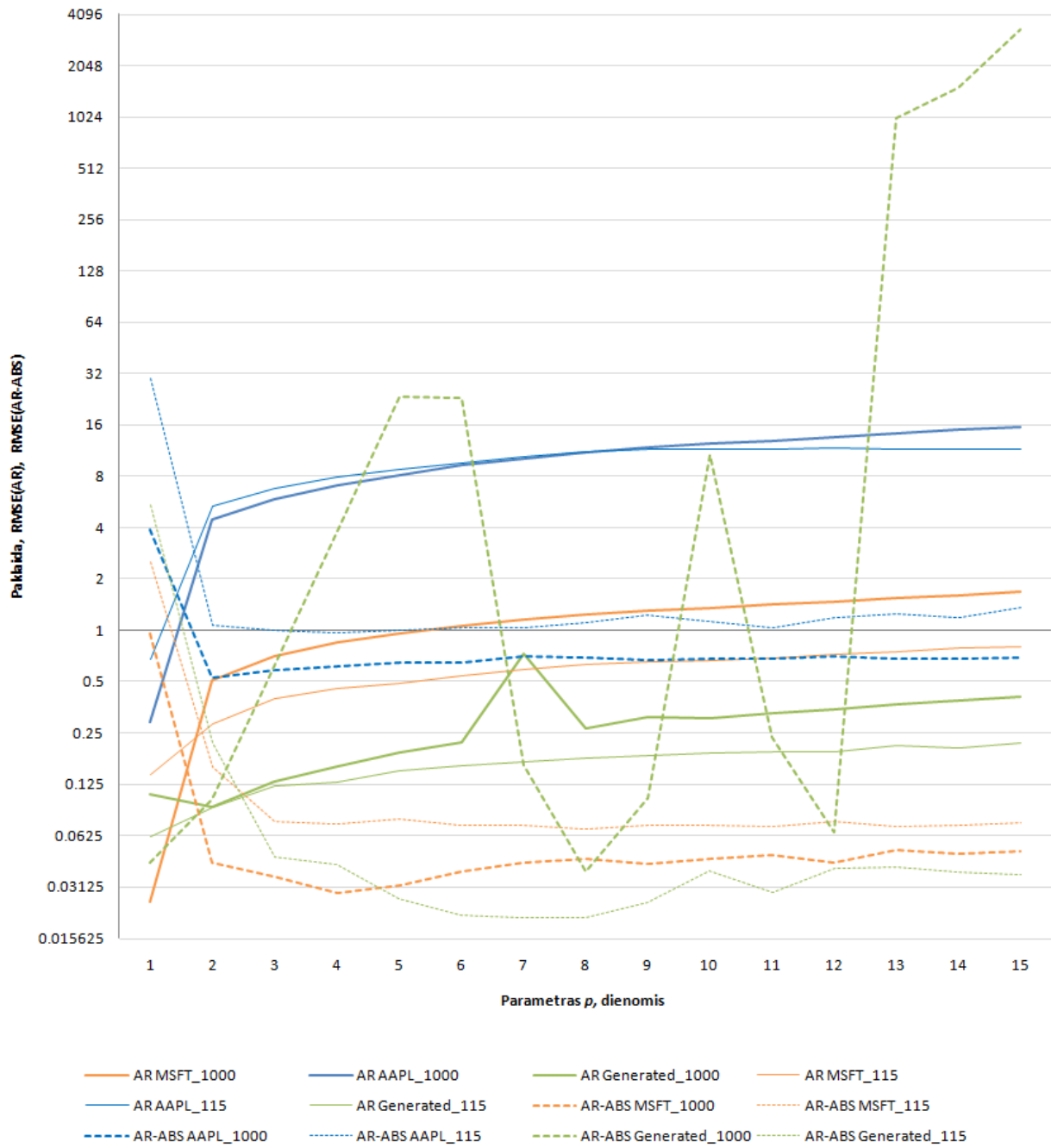
6.14 pav.: Parametrų optimizavimo statistika. 4-5



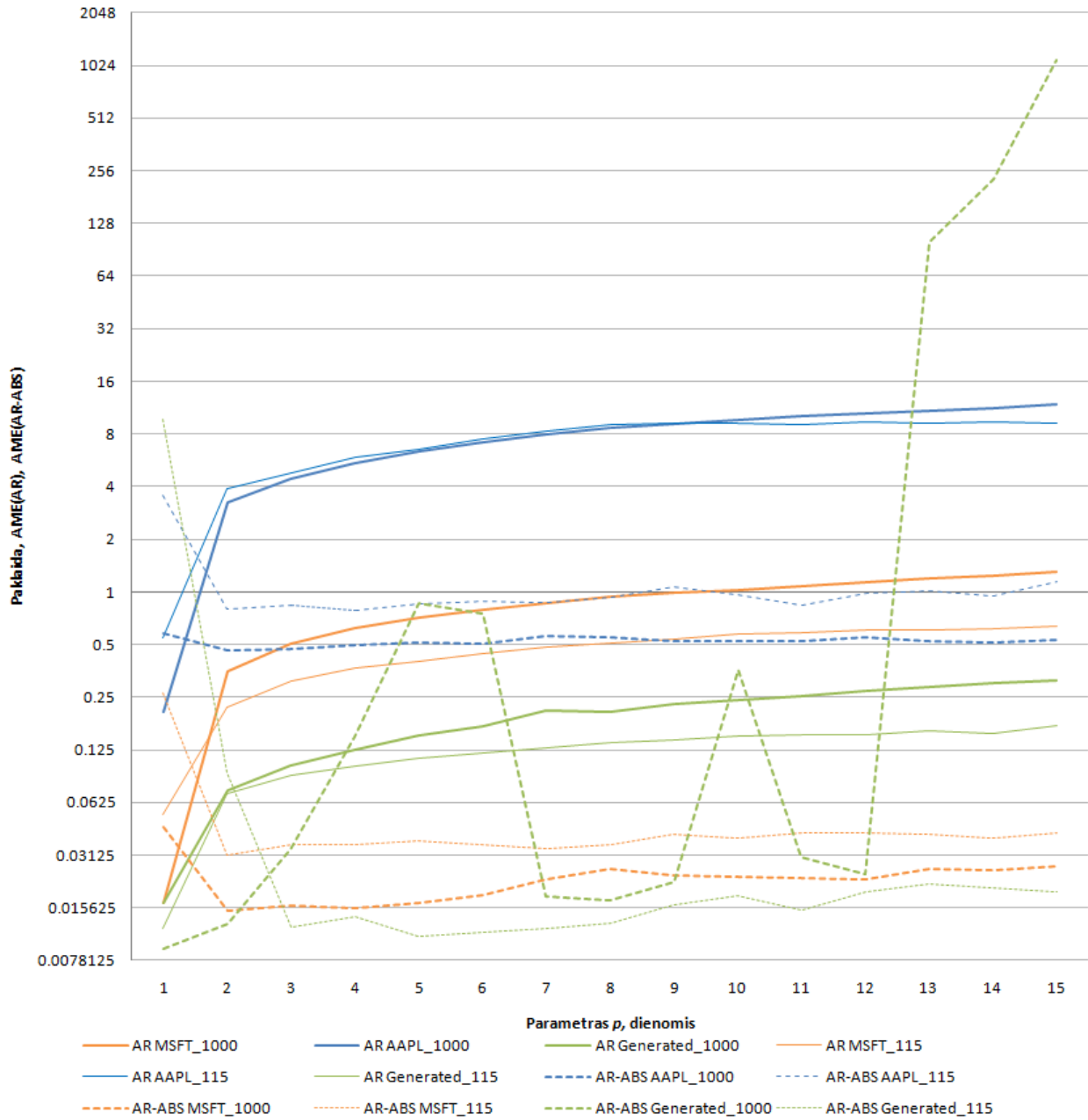
6.4. Priedas 4. Papildomi optimizavimo rezultatai

32	AR-ABS	1	data_T2_115.csv	res_AR-ABS_p1_T2_115.csv	5.392136931	9.631693616
33	AR-ABS	2	data_T2_115.csv	res_AR-ABS_p2_T2_115.csv	0.216910959	0.093046448
34	AR-ABS	3	data_T2_115.csv	res_AR-ABS_p3_T2_115.csv	0.046932616	0.012060904
35	AR-ABS	4	data_T2_115.csv	res_AR-ABS_p4_T2_115.csv	0.042431379	0.013947773
36	AR-ABS	5	data_T2_115.csv	res_AR-ABS_p5_T2_115.csv	0.026590514	0.010717589
37	AR-ABS	6	data_T2_115.csv	res_AR-ABS_p6_T2_115.csv	0.021363737	0.011243102
38	AR-ABS	7	data_T2_115.csv	res_AR-ABS_p7_T2_115.csv	0.02073159	0.01189486
39	AR-ABS	8	data_T2_115.csv	res_AR-ABS_p8_T2_115.csv	0.020501314	0.012811171
40	AR-ABS	9	data_T2_115.csv	res_AR-ABS_p9_T2_115.csv	0.025115726	0.016188789
41	AR-ABS	10	data_T2_115.csv	res_AR-ABS_p10_T2_115.csv	0.039015618	0.018190781
42	AR-ABS	11	data_T2_115.csv	res_AR-ABS_p11_T2_115.csv	0.028950862	0.015278469
43	AR-ABS	12	data_T2_115.csv	res_AR-ABS_p12_T2_115.csv	0.040053615	0.019264317
44	AR-ABS	13	data_T2_115.csv	res_AR-ABS_p13_T2_115.csv	0.040898746	0.021368851
45	AR-ABS	14	data_T2_115.csv	res_AR-ABS_p14_T2_115.csv	0.038140046	0.020246769
46	AR-ABS	15	data_T2_115.csv	res_AR-ABS_p15_T2_115.csv	0.036553964	0.019311139
					2.456893896	0.661817639
Vykdymo laikas: 06:22:41 h						

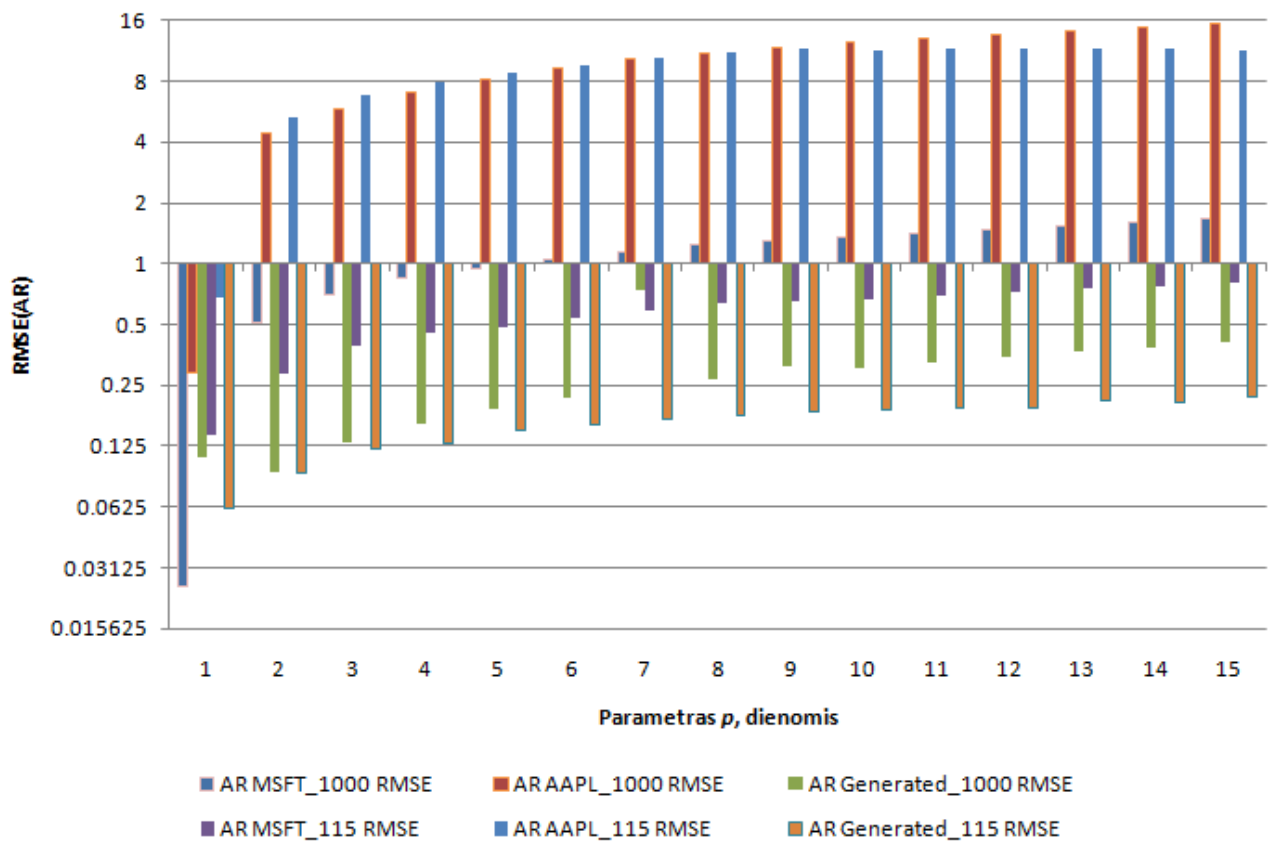
6.15 pav.: Parametrų optimizavimo statistika. 5-5



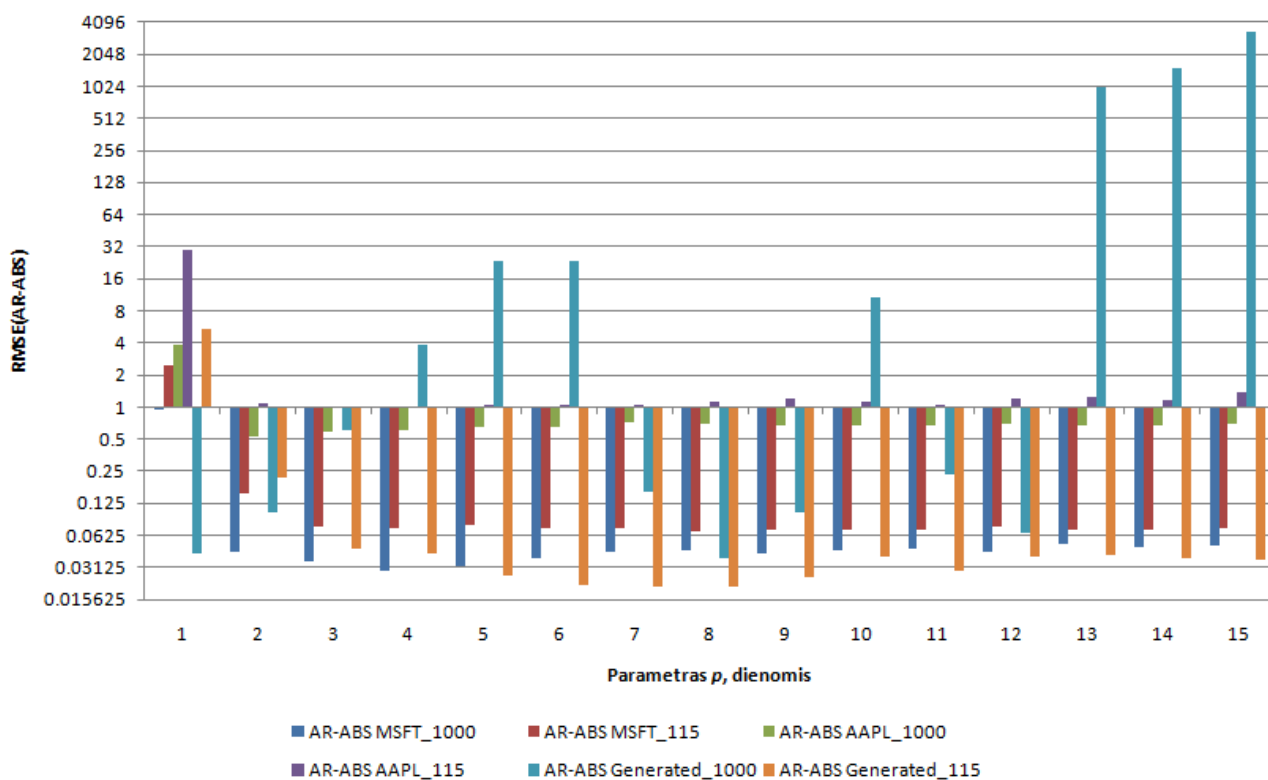
6.16 pav.: RMSE priklausomybė nuo parametro  $p$



6.17 pav.: AME priklausomybė nuo parametro  $p$



6.18 pav.: RMSE pasiskirstymas  $AR(p)$  atžvilgiu



6.19 pav.: RMSE pasiskirstymas  $AR - ABS(p)$  atžvilgiu