



**KAUNO TECHNOLOGIJOS UNIVERSITETAS  
ELEKTROS IR ELEKTRONIKOS FAKULTETAS**

Dainius Pocius

**DAUGIAFUNKCINĖS VALDYMO BALSU SISTEMOS SUKŪRIMAS  
IR TYRIMAS**

Baigiamasis magistro projektas

**Vadovas**

Doc. dr. Kastytis Ratkevičius

**KAUNAS, 2017**

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**ELEKTROS IR ELEKTRONIKOS FAKULTETAS**  
**AUTOMATIKOS KATEDRA**

**Daugiafunkcinės valdymo balsu sistemos sukūrimas ir tyrimas**  
Baigiamasis magistro projektas  
Valdymo technologijos (kodas 621H66001)

**Vadovas**

Doc. dr. Kastytis Ratkevičius

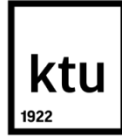
**Recenzentas**

Doc. dr. Romas Rutkauskas

**Projektą atliko**

Dainius Pocius

**KAUNAS, 2017**



KAUNO TECHNOLOGIJOS UNIVERSITETAS

Elektros ir elektronikos fakultetas

(Fakultetas)

Dainius Pocius

(Studento vardas, pavardė)

**Valdymo technologijos (621H66001)**

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Daugiafunkcinės valdymo balsu sistemos sukūrimas ir tyrimas“

**AKADEMINIO SĄŽININGUMO DEKLARACIJA**

20 17 m. 06 05 d.

Kaunas

Patvirtinu, kad mano **Dainiaus Pocius** baigiamasis projektas tema „Daugiafunkcinės valdymo balsu sistemos kūrimas ir tyrimas“ yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

\_\_\_\_\_  
(vardą ir pavardę įrašyti ranka)

\_\_\_\_\_  
(parašas)

Pocius, Dainius. Daugiafunkcinės valdymo balsu sistemos sukūrimas ir tyrimas. Valdymo sistemų magistro baigiamasis projektas / vadovas doc. dr. Kastytis Ratkevičius; Kauno technologijos universitetas, Elektros ir elektronikos fakultetas, Automatikos katedra.

Mokslo kryptis ir sritis: Elektros ir elektronikos inžinerija, Technologiniai mokslai

Reikšminiai žodžiai: Paslėptieji Markovo modeliai, WEKA, daugiafunkcinė balsu valdoma sistema, HTK.

Kaunas, 2017. 65 p.

## **SANTRAUKA**

Viena pirmųjų ir ilgiausiai vartojamų žmonių tarpusavio komunikavimo formų yra kalba. Sparčiai vystantis technologijoms žmonės kalbą pradėjo naudoti ne tik komunikacijai tarpusavyje, bet ir pradėjo komunikuoti su technologijomis. Pagrindinis to tikslas yra įrenginių valdymas nebe fiziniu darbu, o išstartomis balsu komandomis. Prietaisai, kurie valdomi balsu, orientuoti į žmones turintčius regėjimo ar judėjimo negalias. Iki šiol kuriamos balsu atpažinimo technologijos yra skirtos valdyti vienai sistemai, tai sukelia nepatogumų vartotojui norinčiam valdyti kelias skirtingas sistemas vienu metu. Šito darbo vienas iš tikslų yra daugiafunkcinės balsu valdymo sistemos sukūrimas, skirtos balsu valdyti šešias skirtingas sistemas.

Atlikus išsamią literatūros analizę, apimančią kalbos atpažinimo technologijas, atpažinimo sistemos veikimo principą ir taikomus metodus kalbos atpažinimo modeliavime, atliekami balsu komandų atpažinimo tyrimai, naudojant paslėptųjų Markovo modelių lietuvišką ir kitakalbį, atpažintuvus. Išstartų balsu komandų tyrimai atliekami naudojant paslėptiesiems Markovo modeliams taikomus metodus, kitakalbio atpažintuvo panaudojimą kitai kalbai atpažinti metodus, duomenų analizės sistemą WEKA, Auto-WEKA sistemoje esančių duomenų klasifikatorius ir kt.

Šiame darbe pateikiama „Daugiafunkcinė balsu valdoma sistema“, atpažįstanti išstartas, užprogramuotiems įrenginiams skirtas valdyti, balsu komandas. Programa parašyta C# programavimo kalba. Atlikti balsu komandų atpažinimo tikslumo tyrimai. Gauti tyrimų rezultatai išanalizuoti ir pateikiamos išvados.

Pocius Dainius. Creation and investigation of a multifunctional voice controlled system. Master's thesis in control systems / supervisor doc. dr. Kastytis Ratkevičius; Kauno University of Technology, Faculty of Electrical and Electronical Engineering, department of Automation.

Research area and field: Electrical and Electronics Engineering, Technological Sciences

Keywords : Hidden Markov Models, WEKA, multifunctional voice controlled system, HTK.

Kaunas, 2017. 65 p.

## **SUMMARY**

One of the first and longest-used forms of communication between people is language. With the rapid development of technologies, humans began to use language not only for communication with each other, but they also began to communicate with technology. The main purpose of this is that devices would be controlled not by physical work, but by spoken voice commands. The devices, which are controlled by voice, are aimed at people having visual or mobility disabilities. Until now, developing voice-recognition technology is designed to manage a single system, but it causes inconvenience for the consumer who wants to manage several different systems at the same time. One of the main purposes of this thesis is the creation of a multifunctional voice controlled system, for voice control of six different systems.

After a detailed analysis of the literature, covering speech recognition technologies, the identification of system operation principle and methods used in speech recognition modeling. Research of speech recognition was accomplished by using other language recognition system and Lithuanian recognition system. The studies of spoken voice commands are conducted using the hidden Markov models applicable methods, non-native identifiers of speech recognition is used for other language recognition, WEKA data analysis system, the classifiers used in Auto-WEKA and others.

A Multifunctional voice controlled system is handed in this thesis, recognising voice commands which are preprogrammed to control specific devices. The program is written in C # programming language. The voice command recognition accuracy tests are carried then out. The obtained results were analyzed and conclusions were given.

# TURINYS

LENTELIŲ SĄRAŠAS .....	7
PAVEIKSLŲ SĄRAŠAS.....	7
SANTRUMPŲ IR ŽENKLŲ AIŠKINIMO ŽODYNAS .....	8
ĮVADAS .....	9
1. KALBOS ATPAŽINIMO SISTEMOS APŽVALGA.....	10
1.1 Kalbos atpažinimo technologija .....	10
1.2 Kalbos atpažinimo technologijų taikymo sritys .....	11
1.2.1 SAPI sąsaja .....	12
1.3 Kalbos atpažinimo sistemos veikimas .....	13
1.4 Kalbos atpažinimo metodai.....	14
1.4.1 Dinaminis laiko skalės iškraipymas.....	15
1.4.2 Paslėpti Markovo modeliai.....	17
1.4.3 Dirbtiniai neuroniniai tinklai .....	19
1.5 Kalbos atpažinimo įrankiai .....	20
1.5.1 HTK.....	20
1.5.2 ATK.....	22
1.5.3 CMU Sphinx įrankių rinkinys .....	22
1.5.4 Julius įrankis.....	22
1.6 Garsynas .....	23
1.7 Duomenų analizės sistemų apžvalga .....	23
1.7.1 Auto-WEKA .....	24
2. METODINĖ DALIS .....	26
2.1 Daugiafunkcinės valdymo balsu sistemos sukūrimas .....	26
2.2 Garsyno ir gramatikų paruošimas atpažintuvo tikslumo tyrimams.....	34
2.2.1 Garsyno paruošimas atpažintuvo tikslumo tyrimui.....	34
2.2.2 Gramatikų paruošimas atpažinimo tikslumo tyrimams .....	35
2.3 Tyrimo taikant HTK modelius metodika .....	36
2.4 Komandų atpažinimo tyrimo naudojant žodžiais grįstus modelius metodika .....	43
2.4.1 Papildomų būsenų įterpimas į žodžių modelius.....	43
2.4.2 Papildomų Gauso mišinių įterpimas į žodžių modelius .....	44
2.5 Kitakalbis atpažintuvas.....	44
2.6 Dviejų balso atpažintuvų sujungimo metodika .....	45
2.7 Duomenų požymių tyrimo metodika .....	47
3. ATLIKTŲ TYRIMŲ REZULTATŲ DALIS .....	49
3.1 Komandų atpažinimo tyrimas naudojant HTK metodą, pridedant tik papildomas būsenas .....	49
3.2 Komandų atpažinimo tyrimas naudojant HTK metodą, pridedant papildomas būsenas ir Gauso mišinius .....	50
3.3 Tyrimas naudojant kitakalbį atpažintuvą .....	54
3.4 Hibridinio atpažintuvo galimybių tyrimas .....	56
3.4.1 Hibridinio atpažintuvo galimybių tyrimas naudojant WEKA .....	57
IŠVADOS ir REZULTATAI.....	62
LITERATŪROS ŠALTINIAI .....	63

PRIEDAI.....	66
Priedas 1. Komandos Antras PMM.....	66
Priedas 2. Lipto valdymo komandų gramatikos failas .....	72
Priedas 3. Lipto komandų sąrašo failas.....	74
Priedas 4. Daugiafunkcinės sistemos valdymo balsu programos kodas .....	75
Priedas 5. PMM modelių atpažinimo tikslumas keičiant būsenas.....	87
Priedas 6. PMM modelių atpažinimo tikslumas, keičiant busenas ir Gauso mišinius .....	88
Priedas 7. Hibridinio atpažintuvo klasifikatorių atpažinimo tikslumai.....	89
Priedas 8. Hibridinio atpažintuvo atpažinimo tikslumo požymių failas .....	91
Priedas 9. Auto-WEKA rezultatas .....	101

## LENTELIŲ SĄRAŠAS

2.1 lentelė. Lipto modelio valdymo komandų vidinės būsenos .....	44
2.2 lentelė. ATP_HB atpažinimo tikslumo papildomumo lentelė.....	47
3.1 lentelė. Komandų atpažinimo tikslumo priklausomybė nuo būsenų skaičiaus.....	49
3.2 lentelė. Komandų atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus, nepridėjus būsenų .....	51
3.3 lentelė. Komandų atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus, pridėjus 2 būsenas .....	52
3.4 lentelė. Komandų atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus, pridėjus 4 būsenas .....	53
3.5 lentelė. Komandų atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus, pridėjus 6 būsenas .....	54
3.6 lentelė. Atpažintuvų ATP_SP ir ATP_LT rezultatų papildomumas .....	56
3.7 lentelė. Visų failų ir atskirų 11 aplankų vidurkio atpažinimo tikslumai taikant skirtingus klasifikatorius.....	57
3.8 lentelė. Klasifikavimo tikslumo priklausomybė nuo „RF“ klasifikatoriaus kintančių parametru.....	59
3.9 lentelė. Klasifikavimo tikslumo priklausomybė nuo „SMO“ klasifikatoriaus kintančių parametru.....	60
3.10 lentelė. Lietuviško ir Hibridinio atpažintuvų atpažinimo tikslumų pokyčiai, esant žinomam kalbėtojui.....	61
3.11 lentelė. Lietuviško ir Hibridinio atpažintuvų atpažinimo tikslumų pokyčiai, esant nežinomam kalbėtojui .....	61

## PAVEIKSLŲ SĄRAŠAS

1.1 pav. SAPI terpė [1].....	13
1.2 pav. Automatinio šnekos atpažinimo sistemos struktūra [16].....	14
1.3 pav. a) Tiesinis laiko skalės iškraipymas b) Dinaminis laiko skalės iškraipymas [4].....	16
1.4 pav. Trajektorijos matricoje paieškos veikimas naudojant dinaminio programavimo principą [4] .....	16
1.5 pav. Struktūrinis paslėptojo Markovo modelio su neemituojančiomis įėjimo ir išėjimo .....	18
būsenomis vaizdas [13].....	18
1.6 pav. Neurono branduolys [11] .....	19
1.7 pav. Daugiasluoksnis perceptronas su dviem paslėptaisiais sluoksniais [11] .....	19
1.8 pav. HTK paketu šnekos apdorojimo schema [6].....	21
1.9 pav. WEKA programinio paketo, kairėje a) „WEKA GUI“ darbo langas, o dešinėje b) „WEKA Explorer“ darbo langas.....	24
2.1 pav. Balsu valdomos daugiafunkcinės programos darbo langas skirtas, kairėje a) pozicionavimui valdyti ir dešinėje b) pasirinktam įrenginiui ar sąsajai valdyti.....	26
2.2 pav. Daugiafunkcinės balsu valdomos sistemos atidarytas COM sąsajos langas .....	28
2.3 pav. Daugiafunkcinės balsu valdymo sistemos atidarytas USB sąsajos langas.....	29

2.4 pav. Daugiafunkcinės balsu valdomos sistemos atidarytas Interneto naršyklės langas .....	29
2.5 pav. Daugiafunkcinės balsu valdomos sistemos atidarytas PC darbalaukio langas.....	30
2.6 pav. Daugiafunkcinės balsu valdomos sistemos atidarytas PC grotuvo langas.....	30
2.7 pav. Daugiafunkcinės balsu valdymo sistemos atidarytas Bevielės sąsajos langas .....	31
2.8 pav. Daugiafunkcinės sistemos darbo algoritmas.....	33
2.3 pav. Daugiafunkcinė valdymo balsu sistema .....	34
2.4 pav. Balso komandos „hmm_antras” modelių failo turinys.....	39
2.5 pav. Komandos „antras“ transkripcijos pavyzdys .....	45
2.6 pav. Hibridinio atpažintuvo veikimo schema.....	46
3.1 pav. Komandų atpažinimo tikslumo priklausomybė nuo būsenų skaičiaus .....	50
3.2 pav. Komandų atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus, ne pridėjus papildomų būsenų .....	51
3.3 pav. Komandų atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus, pridėjus dvi papildomas būsenas..	52
3.4 pav. Komandų atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus, pridėjus keturias papildomas būsenas.....	53
3.5 pav. Komandų atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus, pridėjus šešias papildomas būsenas	54
3.6 pav. Kitakalbio atpažintuvo 11 diktorių išstartų komandų atpažinimo tikslumas.....	55
3.7 pav. Kitakalbio atpažintuvo 11 diktorių išstartų 9 komandų atpažinimo tikslumai .....	55
3.8 pav. Efektyviausio klasifikatoriaus radimas tiriant visų komandų įrašų atpažinimo tikslumą .....	58
3.9 pav. Efektyviausio klasifikatoriaus radimas tiriant vienolikos kartų kryžmintų 11 aplankų komandų įrašų atpažinimo tikslumą.....	58

## SANTRUMPŲ IR ŽENKLŲ AIŠKINIMO ŽODYNAS

API	Programinės įrangos kūrimo sąsaja (plg. angl. Application programming interface)
AŠA	Automatinis šnekos atpažinimas (angl. Automatic Speech Recognition)
ATP_HB	Hibridinis atpažintuvas
ATP_LT	Lietuviškas atpažintuvas
ATP_SP	Ispaniškas atpažintuvas
COM	Objektinis komponentų modelis (plg. angl. Component Object Model)
DLK	Dinaminis laiko skalės kraipymas
DNT	dirbtiniai Neuroniniai tinklai
DTW	Dinaminis laiko skalės iškraipymas (angl. Dynamic time warping)
HMM	Hidden Markov Model Toolkit
HTK	angl. (Hidden Markov Toolkit)
MAC	Media access control protocol
PMM	Paslėptieji Markovo modeliai
SAPI	Kalbos taikomoji programavimo sąsaja (plg. angl. Microsoft Speech Application Programming Interface)
SRGS	(angl. Speech Recognition Grammar Specification)
SSML	(angl. Speech Synthesis Markup Language)
USB	Universal serial bus
XML	Išplėstinė ženklinimo kalba (plg. angl. eXtensible Markup Language)



## IVADAS

Viena pirmųjų ir ilgiausiai vartojamų žmonių tarpusavio komunikavimo formų yra kalba. Sparčiai vystantis technologijoms, žmonės kalbą pradėjo naudoti ne tik komunikacijai tarpusavyje, bet ir pradėjo komunikuoti su technologijomis. Automatinės šnekos atpažinimo (*angl. Automatic Speech Recognition*) technologijos leidžia žmonėms komunikuoti su kompiuterine technika ir kitais elektroniniais įtaisais. Pagrindinis to tikslas tai įrenginių valdymas ne fiziniu darbu, o balsu. Prietaisai, kurie valdomi balsu, orientuoti į žmones turintčius regėjimo ar judėjimo negalias. Toks būdas įgalina ir padeda jiems valdyti prietaisus balsu bei klausyti kompiuteryje surinktą ar renkamą tekstą. Pakolkas kuriamos balso atpažinimo technologijos skirtos valdyti vienai sistemai, o tai gali sukelti nepatogumų vartotojui norinčiam valdyti kelias skirtingas sistemas vienu metu. Šito darbo vienas iš tikslų yra daugiafunkcinės balso valdymo sistemos sukūrimas, skirtos balsu valdyti šešias skirtingas sistemas.

Valdymo balsu sistemos veikimas nėra sudėtingas ir susideda iš kelių etapų. Automatinės šnekos atpažinimo sistemos iš gauto kalbos signalo, signalinės analizės būdu, gauna kalbos signalo parametrus, taikant atpažinimo metodus (HTK, Julius, Sphinx ir t.t), suskirsto tuos parametrus ir lygindama su turimu garsynu, identifikuoja žodžius, kuriuos sako žmogus ir konvertuoja juos į tekstą, kuris atitiktų išstartą valdymo komandą, užkuodotas atlikti tam tikrą darbą.

Balso atpažinimo programų kūrimui dažniausiai naudojama „*Microsoft Visual Studio*“ programinė įranga. Buvusioje KTU laboratorijose naudojamos įrangos programinė bazė yra parašyta C++ kalba. Tai yra gana sena ir plačiai naudojama programavimo kalba, tačiau atsirandant naujoms technologijoms, keičiasi ir programavimo reikalavimai, priimta sistemas programuoti C# programavimo kalba, nes ji turi panašumų su C++ programavimo kalba ir yra paprastesnė.

Viena iš problemų yra ta, kad kalbos atpažinimo sistemos pritaikytos tik plačiai vartojamoms kalboms, tokioms kaip: anglų, japonų, prancūzų, vokiečių ir kt. Lietuvių kalbos struktūra yra gana sudėtinga, jai netinka kitoms kalboms sukurti akustiniai modeliai. Todėl yra svarbu kurti lietuvių kalbos atpažinimo sistemas, atlikti tyrimus skirtingais metodais ir pasitelkiant įvairius kalbos atpažinimo įrankius pvz., Julius, Sphinx, HTK, kad šios kalbos vartojimas nebūtų išstumtas iš modernių technologijų.

# 1. KALBOS ATPAŽINIMO SISTEMOS APŽVALGA

Šiame skyriuje yra aptariami kalbos atpažinimo veikimo principai, kalbos atpažinimo technologijos bei jos aktualumas pasaulyje. Trumpai aprašomi kalbos atpažinimo metodai, jų veikimas, atsižvelgiama į kiekvieno metodo trūkumus ir privalumus, pagal kuriuos matome, kad geriausiai yra naudoti paslėptųjų Markovo modelius (PMM). Supažindinama su naujais metodais, kurie nors ir yra tik dabar vystomi pasaulyje, bet jau rodo gerus balso atpažinimo sistemų rezultatus. Paminimi įvairūs kalbos atpažinimo įrankiai HTK, ATK, Julius ir Sphinx. Aprašomi garsynų tipai, akcentuojama, kad garsyno kūrimas turi didelę įtaką kalbos atpažinimo sistemoje. Aprašoma duomenų analizės sistemų apžvalga.

## 1.1 Kalbos atpažinimo technologija

Vystantis pramonei, žmonių techninė veikla tapo orientuota į įrenginių ir sistemų, gebančių imituoti žmogaus judesius, su tikslu išlaisvinti žmones nuo fizinio krūvio ir monotonijos, kūrimą. Viena iš žmogaus fizinį darbą lengvinančių vystomų technologijų yra balso atpažinimo sistemos. Mes gyvename technologijų, informacijos ir visuotinės kompiuterizacijos amžiuje, tobulėjant technologijoms ir pingant senoms technologijoms, naujo ar seno kompiuterio turėjimas namuose jau tapo norma. Atsižvelgiant į technikos orientavimosi kryptis, šiuolaikiniuose bei senesniuose kompiuteriuose yra įdiegiamos balso atpažinimo technologijos, kurios palengvina žmonių darbą. Kadangi labiausiai paplitusios atpažinimo sistemos kuriamos ne lietuvių kalba, mažai tikėtina, kad lietuviškos programos bus kuriamos ne Lietuvoje. Vienas iš šio baigiamojo magistrinio darbo tikslų sukurti daugiafunkcinę balsu valdomą sistemą, kuri bus valdoma Lietuvių kalba.

Išskiriamos dvi pagrindinės kalbos atpažinimo technologijų grupės:

1. Balsu išstartų žodžių, jų sekos, frazių automatinis nustatymas (kalbos atpažinimas);
2. Teksto skaitymas balsu (teksto sintezė);

Kalbos atpažinimo paskirtis yra automatiškai nustatyti, kas balsu sakoma informacijos priėmimo sistemai. Tai gali būti vienas atskiras žodis (balso komanda valdyti prietaisus prijungtus per COM ar USB sąsajas ir pan), kelių žodžių seka (pašto adresas), net pilnų rišlių sakinių skaitymas. Informacijos priėmimo sistema, nustačiusi, kas jai buvo pasakyta, atlieka atitinkamus veiksmus.

Kalbos atpažinimas vystomas beveik šešis dešimtmečius, sukurta nemažai atpažinimo metodų, bet dar kyla sunkumų realizuojant visiškai nuo kalbančiojo nepriklausomas, triukšmui atsparias, tikslias atpažinimo sistemas. Pagrindinės problemos kylančios kalbos atpažinime :

Pirmoji problema – tai kalbos signalų kintamumas, kadangi du to paties žodžio ištarimai tarpusavyje gali skirtis tempu, energijos lygiu ir kitomis laikinėmis ar spektrinėmis savybėmis.

Antra problema – natūralios kalbos savybės (pvz., tariant žodžius daromos skirtingo ilgio pauzės, abejojimo garsas „mmm“ ir t.t).

Trečioji problema – dideli žodynai. Juose gaunasi daug akustiškai panašių pavyzdžių ir to rezultatas - painumas auga kartu su žodyno dydžiu.

Ketvirtoji problema – signalo akustinės ir sklidimo terpės ar aplinkos įtaka signalui.

## **1.2 Kalbos atpažinimo technologijų taikymo sritys**

Pirmoji kalbos atpažinimo sistema atsirado maždaug praėjusio amžiaus trečiajame dešimtmetyje. Nuo to laiko technologijoms kylant vis į naujas aukštumas, kalbos atpažinimo sistemos pradėtos naudoti vis įvairesnėse srityse [1]. Kalbos atpažinimo technologijų taikymo pavyzdžiai : Kalbinis dialogas su kompiuteriu; Balsui interaktyvi interneto naršyklė; Balsu valdoma automatizuota gyvenamo namo aplinka; Asmens atpažinimas pagal balsą; Kalbinis dialogas su telefonu.

Kalbinis dialogas su kompiuteriu – tai tiesioginis bendravimas su kompiuteriu balsu. Kaip jau minėta praeitame skyriuje, mūsų kasdienybėje sunku įsivaizduoti gyvenimą be kompiuterio, esančio namų aplinkoje. Dėl tos priežasties dauguma asmenų turi prieigą prie kalbos atpažinimo programų. Tuose programose yra paruoštas komandų sąrašas. Vartotojui ištarus vieną iš jų, kompiuteris atitinkamai reaguoja pagal atpažintos komandos numerį. Nepavykus atpažinti balso komandos, išvedamas pranešimas arba klaida ir prašoma dar kartą pakartoti komandą, jei po to ji atpažįstama, programa baigia darbą.

Interneto naršymas balsu dvidešimt pirmame amžiuje labai naudingas. Interneto programos gali atlikti daugybę komandų, uždaryti langą, sugrįžti atgal ir t.t. Viena didžiausių įmonių „Google“ savo tinklalapyje pateikia galimybę paieškos lauke duomenis įvesti balso komandomis („Google Voice Search“). Interneto naršymas balsu augant vartotojų skaičiui, skatina naujų internetinių programų vystymui ir kūrimui.

Viena didžiausių naudų balso atpažinimo taikyme - taikymas namų butyje ir viešuose aplinkose. Tobulėjant ir mažėjant mikroprocesoriams, namų butyje vis daugėja prietaisų, kuriuos galime valdyti balsu. Lietuvos ir užsienio mokslininkai ir studentai bando pritaikyti įvairias technologijas namų buičiai valdyti balsu. Vienas iš pavyzdžių būtų studentas Stepan Lebedev, kuris panaudojo mikrovaldiklį „Arduino MEGA 2560“ bei modulį „EasyVR“ namo įrenginių valdymui balsu [26].

Asmens atpažinimas pagal balsą svarbus kriminalistikoje. Kalbėtojo atpažinimas pagal jo balsą yra suprantamas, kaip procesas, kurio metu iš asmens balso signalų išskiriami identifikaciniai požymiai, pagal kuriuos ir atpažįstamas konkretus asmuo. Kalbėtojo atpažinimas savo ruožtu skirstomas į identifikavimą ir verifikavimą, kurie gali būti priklausomi arba nepriklausomi nuo teksto. Teismo ekspertizėje asmens atpažinimas pagal balsą vadinamas asmens identifikavimu ir dažniausiai yra nepriklausomas nuo teksto. Prieš tai išvardintos kalbos atpažinimo sistemos problemos apsunkina asmenų atpažinimą pagal balsą.

Kalbinis dialogas su telefonu – tai tiesioginis bendravimas su telefonu. Per pastarąjį dešimtmetį vystantis technologijoms ir atsiradus išmaniesiems telefonams, tam tikro lygio balso atpažinimo programos gali turėti kiekvienas turintis išmanųjį telefoną. Labiausiai išpopuliarėjusi „Apple“ įmonės balso atpažinimo išmanaus telefono interaktyvi programėlė „Siri“. Kadangi šiuolaikiniai išmanieji telefonai prilygsta kompiuteriams, telefone dar gali būti apjungiamos kompiuterio ir interneto naršyklės, kalbos atpažinimo funkcijos.

Nors Lietuvoje kalbos atpažinimo darbai pradėti prieš gerus tris dešimtmečius, pasaulio mastu Lietuva vis dėlto atsilieka kalbos atpažinimo technologijų vystime, mažiausiai vienu dešimtmečiu. Todėl Lietuvių mokslininkams iškilo dvi problemos: beveik visos kalbos atpažinimo technologijos kuriamos anglų arba kitomis kalbomis ir kad visos lietuviškos kalbos atpažinimo sistemos plėtojamos tik Lietuvoje. [12].

Lietuvos mokslininkų ir studentų tikslas, kurti sistemas galinčias sumažinti fizinį krūvį žmonėms, padėti jiems kasdieniuose darbuose, padėti neįgaliesiems. Priemonės neįgaliesiems - kompiuterizuotos darbo vietos akliems, kurtiems, turintiems motorinių funkcijų sutrikimus, neprigirdintiems. Balso technologijos taip pat naudingos medicinoje, pooperacinėje reabilitacijoje, ką rodo nuolat augantis dėmesys kitose šalyse [12].

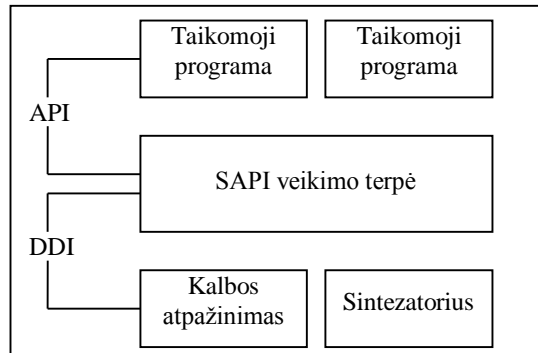
Vienas didesnių lietuviškos kalbos atpažinimo programos projektų „LIEPA“ apima kelias prieš tai išvardintas taikymo sritis ir teikia tokias paslaugas vartotojams [28]:

- tartuvas (leidžia išgirsti lietuvių kalbos naują žodžių tarimą);
- naršytuvas (leidžia naršyti internete naudojant kompiuteriui nurodomas komandas balsu);
- pažintuvas (padeda moksleiviams mokytis ląstelės sandarą naudojant garsines komandas);
- valdytuvas (leidžia valgyti kompiuterį balsu);
- ieškotuvai (balsu galima ieškoti informacijos apie UNESCO kultūros paveldą pasaulyje);
- pagalbininkas (padeda mokytis socialinių gebėjimų negalias turintiems žmonėms);
- lietuvių šnekos sintezatorius akliems (kompiuterio ekrane rodomos informacijos skaitymas).

Viena didžiausių projekto „Liepa“ klaidų yra kalbos atpažinimo tikslumo tyrimo neatlikimas.

### 1.2.1 SAPI sąsaja

*Speech API* (SAPI) tai programa tarp taikomosios programos ir balso atpažinimo arba sintezės modulio (angl. *SR arba TTS engine*), veikianti su Windows operacine sistema. SAPI dažnai vaizduojama kaip sąsaja tarp taikomųjų programų ir kalbos atpažinimo bei teksto vertimo į kalbą įrankių. SAPI su taikomosiomis programomis sąveikauja per API (*Application Programming Interface*) programavimo terpę, o su kalbos atpažinimo ir teksto vertimo į kalbą įrankiais per DDI (*Device Driver Interface*) įrankių tvarkyklės. SAPI terpė pavaizduota 1.1 paveiksle.



**1.1 pav. SAPI terpė [1]**

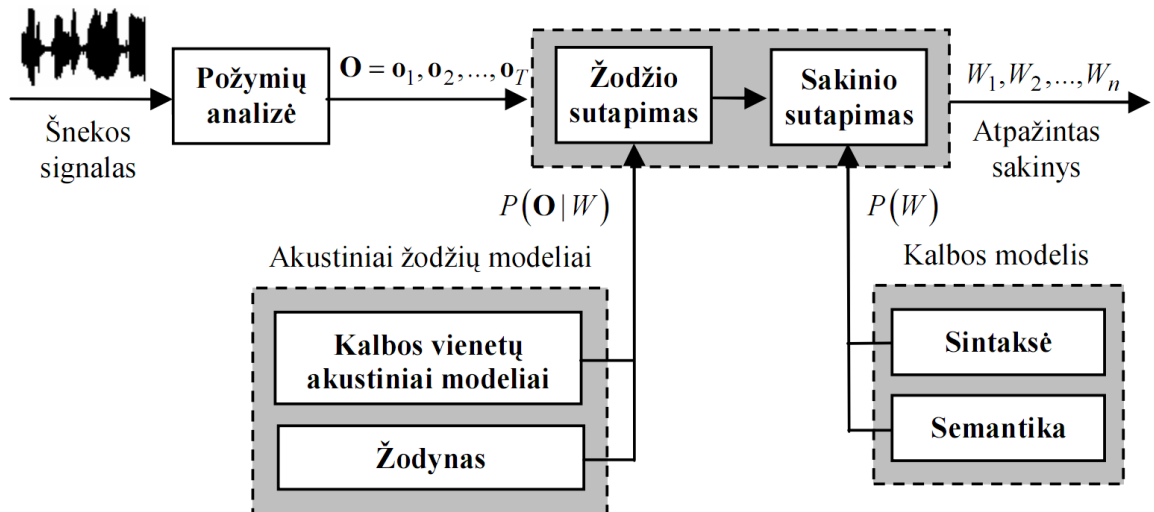
2000 metais išleista SAPI5 (naujausiai išleista yra SAPI5.4 versija) , ją jau naudoja lietuviška programa, sukurta projekte „LIEPA“. SAPI5, lyginant su prieš tai buvusia versija turi daugiau papildomų galimybių: yra paprastesnė, atlieka sintezatoriui reikalingų audio formatų konvertavimą, palaiko tarimo žodynus, išanalizuoja tekste sutiktas XML žymes, atlieka parametrų tikrinimą, palaiko SSML (angl. Speech Synthesis Markup Language) ir SRGS (angl. Speech Recognition Grammar Specification) [2].

SAPI naudojimas standartizuoja taikomosios programos ir balso sintezatoriaus sąveiką, kūrėjams leidžia paprastai naudoti balso technologijas, išsprendžiamos kai kurios sintezatorių kūrėjams išskylančios problemos ir taip sintezatorių kūrėjai gali susikaupti ties balso atpažinimu.

SAPI palengvina atpažinimo modulių kūrėjų darbą atlikdama visą aibę funkcijų: Priima balso duomenis iš mikrofono, failo ar kito įrenginio ir paverčia į atpažinimo moduliui suprantamą formatą; Pakrauna atpažinimo gramatikas; Leidžia kelioms taikomosioms programoms dalintis tą patį atpažinimo modulį; Gražina taikomajai programai atpažinimo rezultatus; užtikrina, kad atpažinimo moduliui būtų perduodami tik leistini parametrai.

### **1.3 Kalbos atpažinimo sistemos veikimas**

Preituose skyriuose aprašomas balso atpažinimo technologijų panaudojimas, šitame skyriuje apžvelgama, kaip minėtos balso atpažinimo technologijos veikia. Kalbos atpažinimas dar žinomas kaip automatinis šnekos atpažinimas (AŠA) – tai procesas, atpažįstantis ir interpretuojantis žmogaus kalbą, ją paversdamas kompiuteriui suprantama signalų kalba. Pagrindinis AŠA sistemų kūrimo ir tobulinimo tikslas yra sukurti mašinas, kurios imituotų žmogaus veiklą t. y. galėtų girdėti, suprasti, kalbėti ir veikti pagal balsu gautą informaciją. Kalbos atpažinime kalbos signalas pasiekia atpažinimo sistemą, kurioje etapais atliekami veiksmai, kurių galutinė užduotis yra, kad sistema atpažintų kalbos signalą, tokios sistemos pavyzdys pateiktas 1.2 paveiksle.



1.2 pav. Automatinio ųnekos atpaųinimo sistemos struktūra [16]

Pagrindiniai AųA sistemos elementai yra: poųymių analize, akustinių ir kalbos modelių paruoųimas, akustinių ųdųių komandų-garsyno sudarymas, ųdųio ir sakinio atpaųinimas,

Ių 1.2 paveikslo matome, kad AųA sistemos struktūra susideda ių kelių dalių, kurios yra:

1. Poųymių iųskyrimas. Ŵioje dalyje ių garso signalo iųskiriami poųymiai  $O$ , kurie bus naudojami tolesniame atpaųinime.
2. Akustinis modelis. Tai modelis, kuris turi įvertinti tikimybę  $P(O/W)$ .
3. Kalbos modelis. Jo paskirtis yra įvertinti tikimybę  $P(W)$ .
4. Paieųka hipotezių erdvėje. Apima akustinio ir kalbos modelių sujungimą bei paieųką ųdųių hipotezių erdvėje.

Kalbos atpaųinime mus domina tik lingvistinė informacija, nepriklausomai nuo diktorius ir kokioje aplinkoje jis kalbėjo. Todėl pirmasis etapas kalbos atpaųinimo procese kaip matome ių 1.2 pav. yra kalbos signalo analizė [17]. Kalbos signalo analizės tikslas – iųtirti poųymius – signalo charakteristikas, atspindinčias lingvistinį turinį. Jos metu gaunami ar iųskiriami ųnekos signalo parametrai, kurie bus naudojami tolesniuose kalbos atpaųinimo procesuose. Kalbos analizė turi tris tikslus: duomenų apimties sumaųinimas; charakteringų duomenų iųskyrimas; diskretinės galios didinimas. Pavienių ųdųių atpaųinimo sistemose remiamasi tik akustiniais modeliais, todėl akustinės analizės metodai yra ųie : spektro analizė; tiesinė prognozė; kepstro analizė.

#### 1.4 Kalbos atpaųinimo metodai

Turint kalbos signalo poųymius sekančiame etape reikia atlikti turimų poųymių klasifikavimą, t. y. pagal atrinktus poųymius nuspręsti, kokiam objektui priklauso ųie poųymiai. Kadangi kiekvienas

žmogus turi savo individualų kalbos stilių, vieni žmonės kalba greičiau kiti lėčiau, todėl žodžiai gali būti sudaryti iš skirtingo duomenų kiekio, to pasakoje gaunamas skirtingų požymių skaičius. Siekiant išspręsti kalbos atpažinimo problemas yra sukurta keletas metodų: akustinis fonetinis, struktūrų atpažinimo, šabloninis, žiniomis grįstas, neuroniniais tinklais grįstas, dinaminiu laiko skalės iškraipimu bei paslėptųjų Markovo modeliais grįsti metodai. Iš visų paminėtų metodų apžvelgsime tris labiausiai pasiteisinančius ir rodančius geriausias rezultatus tai : Dinaminis laiko skalės iškraipymas (Dynamic time warping-based DTW), paslėpti Markovo modeliai PMM (Hidden Markov Model-based HMM) bei dirbtiniai Neuroniniai tinklai DNT.

### 1.4.1 Dinaminis laiko skalės iškraipymas

DLS metodas leidžia rasti dviejų sekų panašumą, kurios gali skirtis trukme ir greičiu. Dviejų sekų palyginimas vyksta pagal tokį eiliškumą [4]:

1. Iš įvesto žodžio išgaunami kalbos požymiai;
2. Kalbos požymiai lyginami su visais turimais etalonais;
3. Labiausiai panašus etalonas, į nagrinėjamą žodį, priimamas kaip teisingas.

Vienas labiausiai kelemų klausimų, lyginant žodžių panašumus, tai kaip nustatomas atstumas tarp etalonų ir turimo žodžio. Apskaičiuotiems atstumams pritaikius minimumo kriterijų, galima priimti sprendimą apie geriausiai atitinkantį etaloną [4]. Atstumams apskaičiuoti yra pasiūlyta nemažai metodų: tikėtinumų santykio matas, tikėtinumų santykio logaritmo matas, Čebyševio atstumas, tačiau dažniausiai naudojamas Euklido atstumas [4]. Euklido atstumas tarp vektorių apskaičiuojamas pagal formulę:

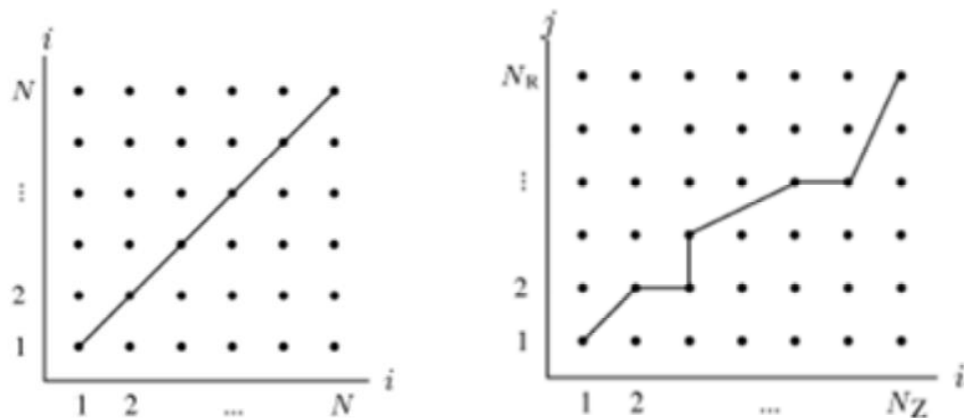
$$d(r_i, z_i) = \sqrt{\sum_{k=1}^L (r_i^k - z_i^k)^2} . \quad (1)$$

Formulėje  $r_i$  ir  $z_i$  – etalono ir nežinomojo  $i$  – ieji požymių vektoriai,  $L$  – nagrinėjamų vektorių eilė,  $k$  – vektoriaus elemento indeksas.

Galimi du laiko skalės iškraipymo atvejai:

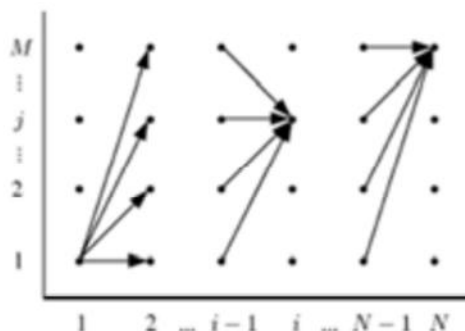
- Tiesinis laiko skalės iškraipymas;
- Dinaminis laiko skalės iškraipymas.

Tiesinio laiko skalės kraipymo metode teigiama, kad greitis tariant garsus pastovus. Kadangi realiame gyvenime kalbėjimo greitis nėra pastovus, tai tokia prielaida yra klaidinga, todėl dažniausiai yra taikomas dinaminis laiko skalės iškraipymas [4]. Abu skalės iškraipymo atvejai yra pateikiami grafiškai apačioje.



1.3 pav. a) Tiesinis laiko skalės iškraipymas b) Dinaminis laiko skalės iškraipymas [4]

Kiekvienas matricos taškas atitinka signalo požymių vektorių dalinį atstumą, kurių indeksą nurodo taško koordinatės. Pagrindinis DLK uždavinys yra optimalios trajektorijos nustatymas. Trajektorijai rasti yra taikomi dinaminio programavimo metodo principai. Šio metodo principas – kiekviename optimizavimo etape priimamas sprendimas garantuoja globalaus kriterijaus optimalumą. Apačioje pateikiamas trajektorijos matricoje paieškos veikimas naudojant dinaminio programavimo principą [4].



1.4 pav. Trajektorijos matricoje paieškos veikimas naudojant dinaminio programavimo principą [4]

Atliekant kalbos pavyzdžių palyginimą labai svarbu išlaikyti nuoseklumą laike. Analizuojami pavyzdžiai turi būti nagrinėjami nuosekliai, nuo pradžios iki galo, lyginant tas pačias pavyzdžių dalis.

Dinaminės laiko skalės iškraipymo metodo privalumai ir trūkumai:

- Vienas iš privalumų yra algoritmo paprastumas.
- DLK metodas apsiriboja akustine pavyzdžių analize, gali būti taikomas žodžiams, skiemenims, garsams atpažinti;
- Nesudėtingas lingvistinio apdorojimo įdiegimas;
- Atpažinimo tikslumas priklauso nuo etalonų skaičiaus;
- Pavyzdžių analizės trukmė priklauso nuo pavyzdžio ilgio;
- Atpažinimo proceso trukmė priklauso nuo žodyno ilgio.



### 1.4.2 Paslėpti Markovo modeliai

Paslėptieji Markovo modeliai (PMM) – tai yra populiarus statistinis Markovo modelis, kuriame sistema yra modeliuojama su paslėptais nariais, arba kitaip – būsenų kintamaisiais. Šis metodas yra plačiausiai naudojamas kalbai atpažinti. Šis metodas remiasi prielaida, kad kalbos signalas yra atsitiktinis procesas, kurio parametrus galima nustatyti [9][29]. Paslėptųjų Markovo modelių (PMM) taikymo metodika remiasi tikimybinio modelių sudarymu. Tai statistiniais pavyzdžiais pagrįstas atpažinimo metodas. Šiuo būdu atliekama spektrinė kalbos signalo analizė, kurios metu gaunami požymio vektoriai, kurie aprašo įvairius kalbos garsus.

Markovo modeliai yra skirstomi [9][29]:

- Stebimas Markovo modelis. Viena būsena atitinka vieną įvykį, todėl stebint įvykių seką galima surasti kokios buvo būsenų seka.
- Paslėptasis Markovo modelis. Būsenų negalime stebėti, nes vienas įvykis yra tikimybinė būsenos funkcija. Viena būsena gali atitikti kelis įvykius.

Formalus PMM aprašymas:

- $A$  – būsenų perėjimo tikimybių pasiskirstymas. Jos parodo, su kokia tikimybe galima pereiti į sekančią sistemos būseną, arba likti toje pačioje;
- $B$  – stebimų simbolių tikimybės pasiskirstymas būsenoje;
- $\pi$  – pradinio buvimo būsenoje tikimybių pasiskirstymas.

Taigi, pilnai paslėptų Markovo modelių specifikacijai aprašyti reikia trijų tikimybinių dydžių  $A$ ,  $B$  ir  $\pi$ . Kad būtų patogiau yra naudojamas trumpesnis aprašymo būdas:

$$\lambda = (A, B, \pi) . \quad (2)$$

Turint stebėjimų aibę  $O$ , kur  $O = o_1, o_2, o_3 \dots o_t$ , tam tikru laiko momentu  $t$ , atpažinimo problema gali būti suvedama į tikimybės skaičiavimą:

$$\operatorname{argmax} \{P(\lambda_i|O)\} . \quad (3)$$

Kur  $\lambda_i$  yra  $i$  – tasis modelis. Kadangi galimų požymių sekų  $O$  yra be galo daug naudojama Bajeso formulė, tada išraiška  $P(\lambda_i|O)$  gali būti užrašoma taip:

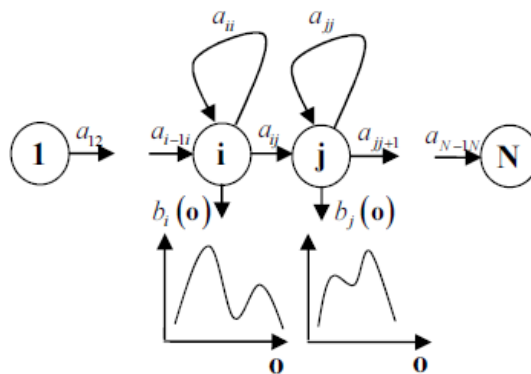
$$P(\lambda_i|O) = \frac{P(O|\lambda_i)P(\lambda_i)}{P(O)} . \quad (4)$$

Čia  $P(O|\lambda_i)$  yra tikimybė, jog tariant žodžių seką  $\lambda_i$ , bus stebima požymių seka  $O$ ,  $P(\lambda_i)$  – apriori tikimybė, jog ištarta žodžių seka  $\lambda_i$ ,  $P(O)$  – tikimybė, kad bus stebima požymių seka  $O$ .

Pritaikant paslėptus Markovo modelius kalbai atpažinti reikia išspręsti tris pagrindinius uždavinius [4], [9], [29]. Pirmas uždavinys - tikimybių skaičiavimas. Tai kai apskaičiuojame stebėjimų sekos tikimybę, duotai stebėjimų sekai ir modeliui (Forward-Backward algoritmas). Uždavinys nurodo, jog, turint stebėjimų seką  $O=O_1, O_2, \dots O_T$  ir modelį  $\lambda$  reikia apskaičiuoti stebėjimo sekos

tikimybę  $P(O|\lambda)$ . Antras uždavinys - atitikimas. Kaip duotai stebėjimų sekai parinkti būsenų seką, kuri būtų optimali (Viterbi algoritmas). Uždavinys nurodo, jog turint stebėjimų seką  $O$ , reikia rasti optimaliausią būsenų seką. Trečias uždavinys - apmokymas. Kaip adaptuoti modelio parametrus, kad maksimizuoti  $P(O|\lambda)$  (Baum-Welch algoritmas). Uždavinys atsakingas už paslėpto Markovo modelio parametrų pakoregavimą taip, kad būtų maksimizuotas stebėjimų sekos tikimybė šiam modeliui.

Paslėptųjų Markovo modelių struktūra, naudojama kalbos atpažinimo sistemose, pavaizduota 1.5 paveiksle.



**1.5 pav. Struktūrinis paslėptojo Markovo modelio su neemituojančiomis įėjimo ir išėjimo būsenomis vaizdas [13]**

Iš paveikslo pastebime kad modelio pradžioje ir pabaigoje pridedama po vieną papildomą neemituojančią (angl. non-emitting) būseną, kurios sujungia kelis modelius. PMM tampa akustiniu modeliu tik po mokymo, kurio metu pagal stebėjimų duomenis, įvertinami modelio parametrai, t. y. perėjimo iš būsenos į būseną tikimybių matrica ir stebėjimų tikimybės tankio funkcijos kiekvienai modelio būsenai.

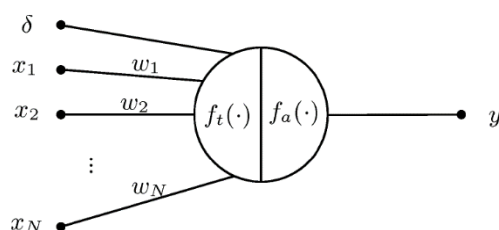
Paslėptųjų Markovo modelių metodo privalumai ir trūkumai:

- Pasiekiamas didesnis atpažinimo tikslumas ir nepriklausomybė nuo kalbėtojo.
- Kalbos pavyzdžių modeliavimas susietomis būsenomis leidžia nesunkiai į akustinį apdorojimą įjungti lingvistinio apdorojimo elementus.
- Sudarant etaloninių kalbos pavyzdžių modelius, dažnai į juos įtraukiama papildoma lingvistinė informacija.
- PMM atveju minimalus reikalingas, duomenų kiekis – bent kelios dešimtys kalbos pavyzdžio versijų (plataus garsyno).
- Egzistuoja pavyzdžių atpažinimo, naudojant PMM, tikslumo priklausomybė nuo duomenų kiekio (garsyno), kuris yra naudojamas modelių apmokymui. Jei duomenų kiekis yra nepakankamas, t. y. garsynas yra mažas, galimas netikslus apmokytų modelių parametrų

įvertinimas. Jei garsynas per didelis, paruoštiems kalbos atpažinimo modeliams gresia permokymas, tada apmokytų modelių parametrai atspindys ne pavyzdžių klases, o pavyzdžių savybes. Abiem paminėtais atvejais mažėja atpažinimo tikslumas.

### 1.4.3 Dirbtiniai neuroniniai tinklai

Neuronas yra sudarytas iš branduolio, įėjimo ir išėjimo taškų. Supaprastintas neurono modelis pateiktas žemiau.



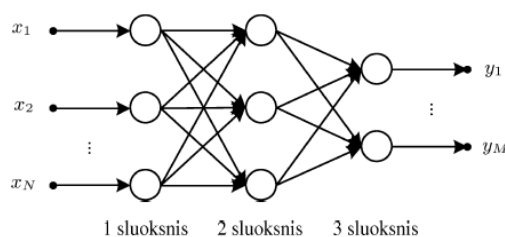
1.6 pav. Neurono branduolys [11]

Neurono branduolį galime aprašyti dviem funkcijomis: sumavimo ir aktyvavimo. Sumavimo funkcija nulemia įėjimo duomenų apdorojimą, o aktyvavimo funkcija formuoja neurono išėjimo signalą, tiesiškai arba netiesiškai transformuodama įėjimo signalus [8][10]. Viršuje pateikto neurono modelio išėjimo signalas apibrėžiamas kaip:

$$y = f_a \left( \sum_{i=1}^N w_i x_i - \delta \right). \quad (5)$$

Kur  $x_i$  – yra  $i$ -asis neurono įėjimas,  $w_i$  – įėjimo svorio koeficientas,  $\delta$  – slenksčio riba (jei šis dydis viršijamas, neuronas yra aktyvuojamas, kitu atveju jis yra neaktyvuojamas),  $f_a$  – aktyvavimo funkcija,  $y$  – neurono išėjimas. Kaip matyti iš 1.6 paveikslo ir (5) formulės neurono reakciją į įėjimo signalą nulemia svorio koeficientai bei tinklo ir aktyvavimo funkcijos [8][10].

Atskirų neuronų branduolių jungimas tarpusavyje lemia neuronų tinklo sukūrimą. Informacija neuronų tinkluose, sklinda iš vieno galo į kita per mazgus. Daugiasluoksniame perceptrone neuronai yra organizuojami sluoksniais (1.7 pav.).



1.7 pav. Daugiasluoksnis perceptronas su dviem paslėptaisiais sluoksniais [11]

Kalbai atpažinti naudojamas 1.7 pav. pavaizduotas daugiasluoksnis perceptronas. Tinklui mokinti reikia turėti segmentuotus ir sugrupuotus kalbos pavydžius. Pats neuroninio tinklo apmokymas yra pakankamai ilgas procesas.

Kaip ir ankstesniuose aprašytuose metoduose (žr. 1.4.1 ir 1.4.2 sk), pateikiami Neuronų tinklo metodo kalbos atpažinimui privalumai ir trūkumai:

- Neuronų tinklų sugebėjimas mokytis.
- Neuronų tinklų skaičiavimų lygiagretumas ir netiesiškumas.
- Literatūroje neaptikta metodikos, kurios pagalba būtų galima nustatyti, kokia tinklo topologija, kokios tinklo charakteristikos.
- Tarp neuronų tinklo topologijos, apmokymo duomenų aibės dydžio, apmokymo laiko ir klasifikacijos tikslumo egzistuoja tarpusavio ryšys.
- Griežtas ryšys tarp analizės duomenų, klasių skaičiaus ir tinklo topologijos.

### **1.5 Kalbos atpažinimo įrankiai**

Vystantis technologijoms bei augant paklausai, sukurta nemažai atviro kodo balso atpažinimo įrankių. Plačiausiai naudojami yra šie: CMU Sphinx, HTK, Julius, ISIP, Spharchcore, NICO, Intel AVSR. Iš šių minėtų įrankių labiausiai pritaikyti ištisinei šnekai ir pavieniams žodžiams (komandoms) atpažinti yra HTK, Julius ir CMU Sphinx.

#### **1.5.1 HTK**

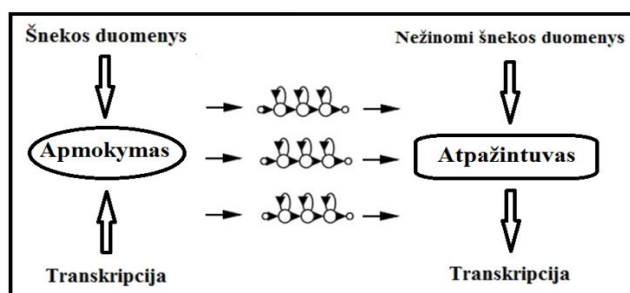
Paslėptojo Markovo modelio programinė įranga - HTK (angl. Hidden Markov Model Toolkit) – programinių priemonių rinkinys, sukurtas Kembridžo universitete. HTK yra vienas iš populiariausių ir plačiausiai paplitusių įrankių, naudojamų paslėptaisiais Markovo modeliais grįsto kalbos atpažinimo tyrimams atlikti. Taip pat HTK naudojamas tyrimams ir kitose srityse - kalbos sintezėi, simbolių atpažinimui ir DNR sekų tyrimuose. HTK tai įrankis akustiniams, kalbos modeliams mokytis, šitie apmokyti modeliai naudojami, PMM grįstu šnekos atpažinimo metodu, tyrimams atlikti.

HTK sudaro bibliotekų modelių rinkinys ir programiniai įrankiai, parašyti C programine kalba. Šiame programiniame rinkinyje galime rasti priemones, skirtas duomenims apdoroti, mokytis ir testuoti akustinius ir šnekos modelius ir priemones akustiniams modeliams adaptuoti konkrečiam kalbėtojui, bei rezultatų analizės priemones.

Naudojantis „*Hidden Markov Model Toolkit*“ programine įranga galima, modeliuoti atskirus žodžius, kurie aktualūs mano tiriamajam darbe ir ištisinės kalbos atpažinimo sistemas, pasirinkti skirtingas paslėptųjų Markovo modelių topologijas ir parametrus, signalą reprezentuojančių požymių tipą.

Pažymimi du pagrindiniai šnekamosios kalbos signalo apdorojimo etapai, esantys HTK programinėje įrangoje. Pirmas - HTK programiniame pakete, šnekamosios kalbos atpažinimui, apmokymo įrankiai yra naudojami siekiant apytiksliai apskaičiuoti paslėptųjų Markovo modelių parametrus, naudojant apmokymui pateiktus komandų įrašus ir parinktas komandų transkripcijas. Antras - Nežinomi komandų įrašai yra transkribuojami naudojant HTK atpažinimo įrankius. [6]

1.8 paveiksle pavaizduota schema, apibūdinanti šiuos du svarbiausius apdorojimo etapus.



1.8 pav. HTK paketu šnekos apdorojimo schema [6]

Naudojant HTK programinį paketą kalbos atpažinimui, kiekviename etape naudojamas specialus programinis įrankis, naudojamas sukurti, apmokyti ir testuoti PMM. Visos programos yra valdomos iš komandinės eilutės per papildomus failus ir eilutės komandas [6]. 2.3 Skyriuje plačiai aprašomos visos programos, kurios yra naudojamos iš HTK programinio įrankių paketo.

Toliau pateikiami keturi pagrindiniai darbo su HTK programiniu paketu etapai žodžių atpažinimo modeliavimui:

1. **Žodyno sudarymas.** Žodyno sudarymas yra reikalingas tam, kad projektuojama sistema žinotų, kokios žodžių (akustinių vienetų) kombinacijos yra naudojamos atpažinimui, todėl darbo eigoje svarbu paruošti tinkamo dydžio garsyną;
2. **Požymių apskaičiavimas.** Atpažinimo sistemose naudojami iš balso komandų (žodžių) įrašų išskirti požymiai. Požymiams apskaičiuoti naudojama programa „Hcopy“;
3. **Modelių apmokymas.** Apmokymas vyksta naudojant „HRest“ įrankį, kuris savo ruožtu apmokymo proceso vykdymui naudoja Baum-Welch algoritmą. Svarbus apmokymo etapas – modelių failų kūrimas, kuriuose nurodomi pagrindiniai PMM modelio parametrai. Modelių failai – tai specialaus formato failai, kuriuose yra surašoma atitinkamo modelio parametrai (žr. 1 PRIEDAS): perėjimų tikimybės ir išėjimų tikimybės. HTK operuoja tik tolydiniais PMM modeliais, todėl išėjimų tikimybės modeliuojamos Gauso skirstiniais. Vadinasi išėjimų tikimybės aprašomos vidurkiais ir standartiniais nuokrypiais;
4. **Atpažinimas.** Atpažinimo testavimas vykdomas naudojant „HVite“ įrankį. Po apmokymo etapo, naudojant testavimui skirtus įrašus (naudojami kiti įrašai nei apmokymui) galima atlikti

atpažinimo testavimą ir patikrinti atpažinimo kokybę su esamais modeliais. Mano darbo atveju sistema apmokoma 10 diktorių ir testuojama 1 diktoriaus, taip sistema testuojama 11 kartų. Atpažinimo kokybės gerinimui iš naujo atliekami visi etapai nuo pirmojo keičiant transkripcijas arba modelių parametrus. Keičiant modelių vidines būsenas ir Gauso mišinius [15].

### **1.5.2 ATK**

ATK - tai įrankis, skirtas kurti eksperimentines programas, naudojant HTK sukurtus akustinius modelius. ATK turi C ++ sluoksnį, kuris komunikuoja su standartinėmis HTK bibliotekomis. ATK palaiko „Unix“ ir „Windows“ platformas. ATK įpatybės: palaiko programavimą gijomis, audio failų įvestį/išvestį, palaiko baigtinių būsenų gramatikas ir trigramų kalbos modelius, leidžia atvaizduoti atpažinimo rezultatus „žodis po žodžio“.

### **1.5.3 CMU Sphinx įrankių rinkinys**

CMU Sphinx – yra kalbos atpažinimo sistemų rinkinys susidedantis iš:

- CMUcmltk – kalbos modelių įrankių rinkinio.
- Pocketsphinx – atpažinimo biblioteka parašyta C kalba.
- Sphinxtrain – akustinių modelių apmokymo įrankis.
- Sphinx4 – atpažinimo biblioteka parašyta Java kalba.
- Sphinxbase – palaikymo bibliotekos reikalingos Pocketsphinx ir Sphinxtrain.

Šis įrankis taip pat palaiko, su HTK sukurtus, akustinius modelius.

### **1.5.4 Julius įrankis**

Julius programinė įranga, tai atviro kodo kalbos atpažinimo sistema, skirta kalbos atpažinimo tyrėjams ir programų kūrėjams. Ji veikia paslėptųjų Markovo modelių pagrindu. Gali atlikti realaus laiko šnekos atpažinimą. Su šia šnekos atpažinimo sistema galima kurti bet kokios kalbos akustinius modelius. Julius biblioteka skirta nepertraukiamos kalbos, didelio žodyno, kalbos atpažinimu užsiimantiems mokslininkams ir programuotojams. Julius optimizuotas greitam kalbos atpažinimui, gali atlikti beveik realaus laiko kalbos signalo dekodavimą šiuolaikiniame kompiuteryje, kalbai kurios žodynas siekia 60000 žodžių. Pasak Julius tinklalapio, Julius biblioteka užima iki 64 Mb atminties, o tai labai aktualu mobiliuose įrenginiuose. Julius biblioteka kompiliuojama ir naudojama iOS platformoje kalbai atpažinti.

## 1.6 Garsynas

Viena iš svarbių kalbos atpažinimo sistemos sudedamųjų dalių yra šnekamosios kalbos garsynas. Garsynas, tai surinkta struktūralizuota garso įrašų aibė konkrečiai kalbai, kur kiekvienas garso įrašas turi savo atitikmenį tekstu (žodžiais, skiemenimis ar transkripciją fenomenomis), žodyną.

Garsynai – tai didelės apimties šnekos signalų imtys, be jų nebūtų įmanomi šnekos atpažinimo sistemų modeliavimai [17]. Garsynai ne tik reprezentuoja kalbą, bet ir pasitarnauja atliekant kalbos tyrimus [16]. Garsynai skiriasi transkribavimo lygmeniu, diktorių skaičiumi, trukme, renkamų garso įrašų turiniu, žodyno dydžiu ir kt.

Didėjant taikomosios paskirties garsynų apimtims, automobilių kūrimo įmonės, bankai, policijos įstaigos ir telefonijos firmos jų kūrimui skiria dideles investicijas. Vienas naujausių garsynų pristatytas lietuvių mokslininkų sukurtos kalbos atpažinimo programos LIEPOS, kurios garsynas susideda iš 113 valandų anotuotos lietuvių kalbos [28]. Daugėja nacionalinių garsynų, kurie dokumentuoja kalbą, kaip tautos socialinės kultūrinės aplinkos paminklą. Kuriami sisteminiai gerai anotuoti garsynai, kurie skiriami išsamiesiems šnekos mokslo tyrimo darbams.

Sukūrus garsyną yra ypatingai svarbu, nuodugniai patikrinti bei užtikrinti, nepriekaištingą visų įrašų kokybę, nes įvairūs trukdžiai bei triukšmai, daro didžiulę įtaką šnekos atpažinimo tikslumui.

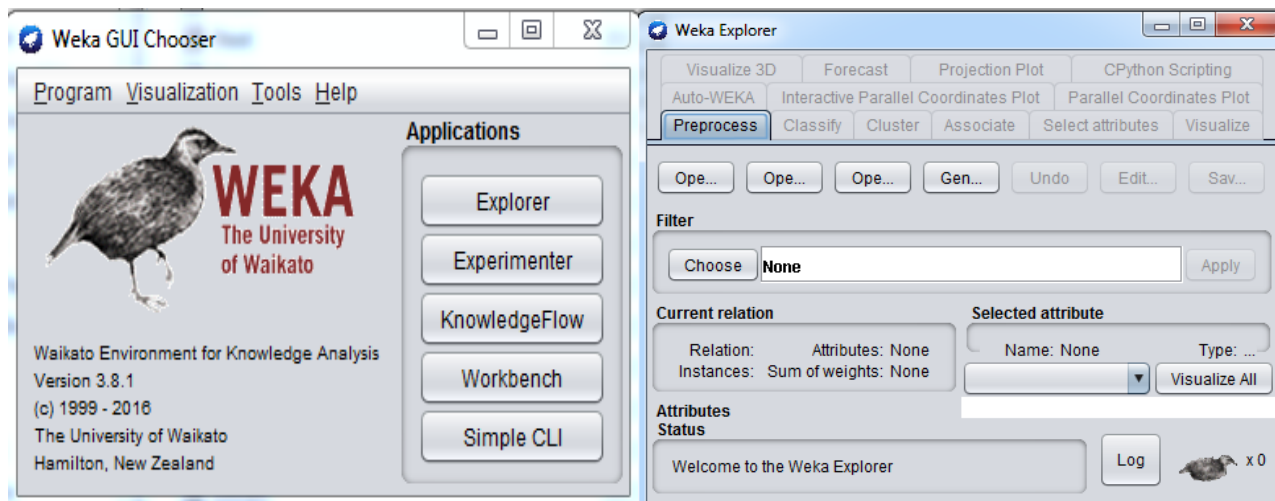
Pagal paskirtį garsynai yra sukuriami sistemoms atpažįstančioms balsą, kuris nepriklauso nuo diktoriaus ir sistemoms atpažįstančioms balsą, kuris priklauso nuo diktoriaus.

Efektyvios kalbos atpažinimo sistemos kūrimui, naudojamas garsynas, privalo: suteikti galimybes vartotojui manipuluoti dideliais duomenų kiekiais ir užtikrinti visų duomenų integralumą ir korektiškumą [14].

## 1.7 Duomenų analizės sistemų apžvalga

Kauno technologijos universitete duomenų požymių analizės tyrimams atlikti, ieškant ir tiriant įvairius klasifikatorius, buvo naudojama inžinerine programine įranga „Matlab“. Nuo 2015 metų buvusioje KTU kalbos signalų tyrimo mokslo laboratorijoje pradėta plačiau naudoti, WEKA duomenų analizės programą. Pastebėta, kad atliekant tyrimus su WEKA, tyrimai atliekami gana greitai ir kad paprasta iširti daugiau klasifikatorių nenaudojant programinio kodo, kurio, pvz., reikalauja programinis paketas „Matlab“. Kadangi „Matlab“ programine įranga atlikti tyrimą, su keliais klasifikatoriais, yra sudėtinga ir reikalauja daug laiko, dviejų atpažintuvų apjungimo tyrimui atlikti (3.4 skyrius), pasirinkta duomenų analizės sistema WEKA.

WEKA programa yra atviro kodo, ją galima parsisiųsti iš oficialaus Weka tinklalapio [30], dabartinė naujausia jos versija yra 3.8.1. (1.9 pav).



**1.9 pav. WEKA programinio paketo, kairėje a) „WEKA GUI“ darbo langas, o dešinėje b) „WEKA Explorer“ darbo langas.**

Vienas naujesnių šios versijos aspektų, tai nauja funkcija Auto-WEKA, kurios pagalba surandamas klasifikatorius, gaunantis geriausius kalbos atpažinimo tikslumo rezultatus.

### **1.7.1 Auto-WEKA**

Auto-WEKA tai įrankis, kuris atlieka sujungtų algoritmų (efektyviausio klasifikatoriaus) pasirinkimą ir hyper-parametrų optimizavimą, pasinaudojantis klasifikacijos ir regresijos algoritmais, esančiais WEKA pakete [26].

Atidarius WEKA programą, pasirenkami duomenys su kuriais bus daromas tyrimas, pasirenkama efektyviausio klasifikatoriaus radimo funkcija Auto-WEKA. Pasirinkus Auto-WEKA, kitas žingsnis yra nustatyti jos pagrindinius parametrus: timeLimit (testavimo laiko limitą) ir memLimit (kompiuterio naudosimas ram kiekio atminties limitą). Darbo metu Auto-WEKA suranda, duotiems duomenims, efektyviausią klasifikatorių ir parenka jam optimaliausius parametrus, atlikusi darbą sukuria modelį ir išveda gautus rezultatus (9 PRIEDAS). Auto-WEKA tyrinėja klasifikatorių parametrus ir vartotojui rekomenduoja metodą, kuris apibendrintai turės geriausią našumą naudojant nustatytus parametrus, pagrįstus optimizavimo metodu. Sukurtą modelį galima išsaugoti ir panaudoti tiriant kitus norimus duomenis. Optimizuojant parametrus naudojant Auto\_Weka gali kilti optimizavimo problemos, parametrai gali neatitikti realių reikšmių, tai pabrėžė vienas iš Auto-WEKA kūrėjų, bandydamas taisyti parametrų optimizavimo paklaidas [24].

Kaip buvo paminėta, Auto-WEKA darbo metu ieško efektyviausio klasifikatoriaus pasirinktiems duomenims tirti. Klasifikatoriai iš kurių išrenkamas tik vienas yra štai tokie: J48, DecisionTable, GaussianProcesses, M5P, KStar, LML, PART, SMO, BayesNet, Classifier, NaiveBayes, JRip, SimpleLogistics, LinearRegression, VotedPerceptron, SGD, Logistic, OneR, MultilayerPerceptron,



REPTree, IBk, M5Rules, RandomForest, RandomTree, SMOreg, Stacking, Bagging, RandomSubSpace, Attribute, SelectedClassifier, RandomCommittee, BestFirst, GreedyStepwise ir CfsSubsetEval.

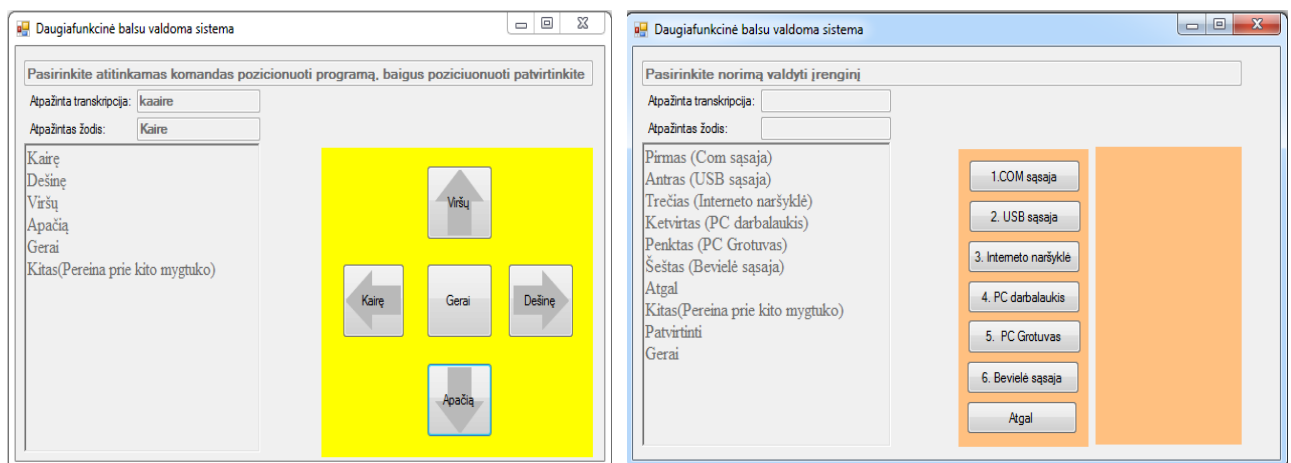
## 2. METODINĖ DALIS

Magistrinio darbo tikslas - sukurti ir ištirti daugiafunkcinę balsu valdomą sistemą. Šiame skyriuje aprašoma daugiafunkcinės sistemos kūrimo eiga, jos sudedamos dalys, veikimo principas. Toliau aprašomos garsyno ir transkripcijų kūrimo metodikos. Plačiai aprašomi, 1.5.1 skyriuje pateikto HTK programinio įrankio paketo, 7 darbo etapai reikalingi lifto modelio komandų atpažinimo tikslumo tyrimui atlikti. Aprašomas kitakalbio atpažintuvo paruošimas, atpažinimo tikslumo tyrimams. Aprašomas hibridinio atpažintuvo paruošimas, atpažinimo tikslumo tyrimams, tiriant „aklą“ atpažinimo taisyklę ir efektyviausius klasifikatorius bei pildant papildomumo lenteles pagal kurias nustatomi sistemų atpažinimo tikslumai.

### 2.1. Daugiafunkcinės valdymo balsu sistemos sukūrimas

Dirbant su „Visual Studio 2015“ programiniu paketu, sukurta programa skirta daugiafunkciniai valdymo balsu sistemai valdyti. Programa sukurta naudojant C# programavimo kalbą.

Atidarius „Visual Studio 2015“ programą, sukuriama programos darbo langas, kuriame bus kuriama, daugiafunkcinės sistemos programa: „File\New\Project\Windows forms application C#“. Naujai sukurtame tuščiajame lange, naudojant įrankių meniu „Tool box“, sudedami visi reikiami elementai, kurie bus naudojami programai valdyti. „View Code“ lange rašomas C# programavimo kalba kodas, skirtas susieti visus sudėtus elementus su operacijomis, skirtomis valdyti daugiafunkcinę sistemą. Pilnai veikiančios programos darbinis langas pavaizduotas 2.1 a) ir b) paveiksluose.



**2.1 pav. Balsu valdomos daugiafunkcinės programos darbo langas skirtas, kairėje a) pozicionavimui valdyti ir dešinėje b) pasirinktam įrenginiui ar sąsajai valdyti.**

Programa pilnai valdoma balsu bet, kaip pavaizduota 2.1 a) ir b) paveiksluose, programa sukurta taip, kad ją būtų galima valdyti ir mygtukų pagalba, taip praplečiant programos funkcionalumą.

Programoje naudojami „Tool Box“ elementai : mygtukai; informacijos atvaizdavimo laukas, skirtas nurodyti darbo eigą; atpažintos transkripcijos atvaizdavimo laukas; atpažinto žodžio atvaizdavimo laukas; apažintos komandos atvaizdavimo laukas; pasirinkto įrenginio MAC adreso langas.

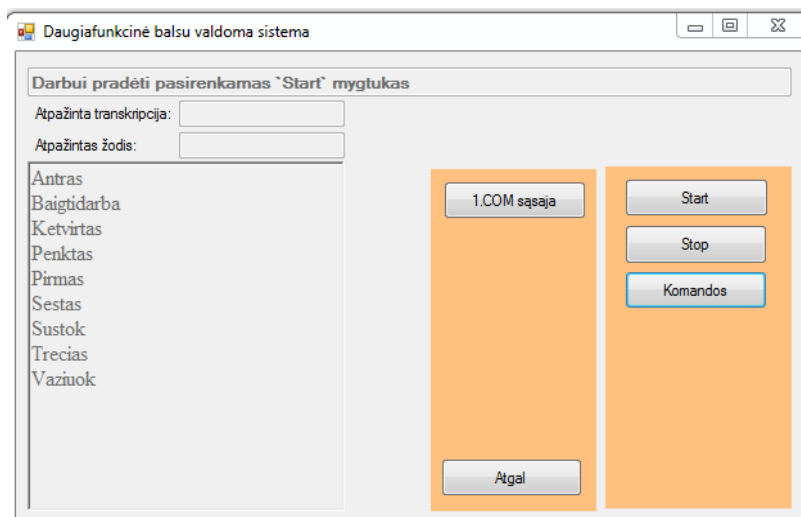
Sukuriami failai pvz., „1.liftokomandos.grxml“ ( 2 PRIEDAS) su komandų transkripcijomis, kurios darbo metu yra kalbos atpažintuvo prilyginamos su ištarta diktoriaus komanda. Sukuriamas ir „1.liftokomandos.doc“ (3.PRIEDAS) failas, su lifto komandų sąrašu, kurios yra įkeliamos į komandų atvaizdavimo lauką, ką ir matome 2.1 a) ir b) paveiksluose.

Daugiafunkcinės valdymo balsu sistemos programos peleidimo metu, įjungiamas kalbos atpažintuvas, įkeliamos pozicionavimo gramatikos „0.1Pozicionavimas.grxml“. Atpažintuvas nuo programos aktyvavimo pradžios, tikrina visas balsu komandas ir lygina jas su „\*.grxml“ failuose esančiomis komandomis. Kad atpažintuvas programos aktyvavimo metu įsijungtų, į programos kodą (4 PRIEDAS) įrašomos trys eilutės: „recognizer.SetInputToDefaultAudioDevice();“, „recognizer.SpeechRecognized += Recognizer\_SpeechRecognized;“ ir „recognizer.RecognizeAsync(RecognizeMode.Multiple);“. Atpažinus ištartą pozicionavimo komandą atliekami veiksmai, užprogramuoti atitinkamai pozicionavimo komandai, taip programa perorientuojama į norimą poziciją, esančią kompiuterio ekrane. Nustačius norimą poziciją ištariama komanda arba paspaudžiamas mygtukas „Gera!“. Atsidaro daugiafunkcinės balsu valdomos sistemos programinis langas pavaizduotas 2.1 b) paveiksle.

Kaip matome (2.1 b) pav.) Daugiafunkcinės valdymo balsu sistemos programa - užprogramuota valdyti šešis skirtingus įrenginius - sąsajas. Visų jų veikimas ir sudėtis aprašyti žemiau.

**Com sąsaja.** Per ją siunčiamos komandos į įrenginius, sujungtus per COM sąsają, vienas iš įrenginių yra lifto modelis, esantis automatikos katedroje. Kad programa siųstu ir priimtu duomenis per COM sąsają, programos kode (4 PRIEDAS) įrašoma įvesties-išvesties sąsajos (I/O Port) programavimo bibliotekos komandinė eilutė „using System.IO.Ports“. Siuntimo greitis ir pasirinkta COM sąsaja nustatomi programiškai, įrašius programos kode: „SerialPort SP = new SerialPort("COM1", 9600)“ eilutę, o siuntimas inicijuojamas įrašius „SP.Write(komanda);“ eilutę.

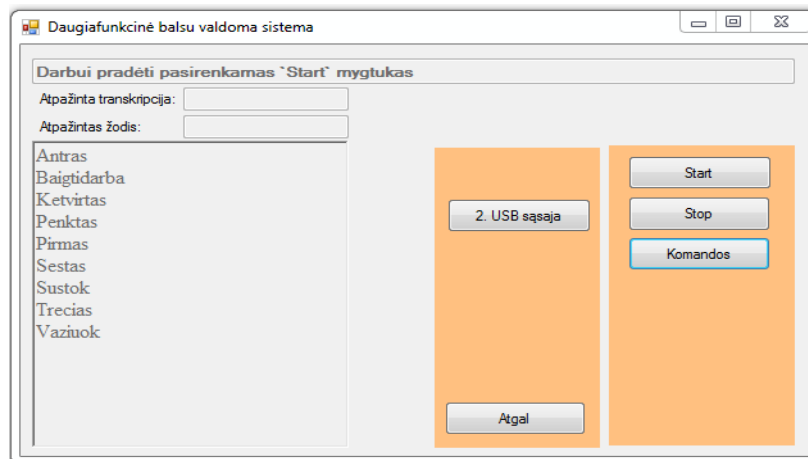
Ištarus arba paspaudus mygtuką „1.Com sąsaja“, dingsta kitų įrenginių pasirinkimo mygtukai ir atsiranda trys papildomi mygtukai „Start“, „Stop“ ir „Komandos“ (2.2 pav). Ištarus arba paspaudus mygtuką „Komandos“, įkeliamos lifto modelio garsyno gramatikos. Balsu komandos nustatytu greičiu siunčiamos per COM sąsają.



**2.2 pav. Daugiafunkcinės balsu valdomos sistemos atidarytas COM sąsajos langas**

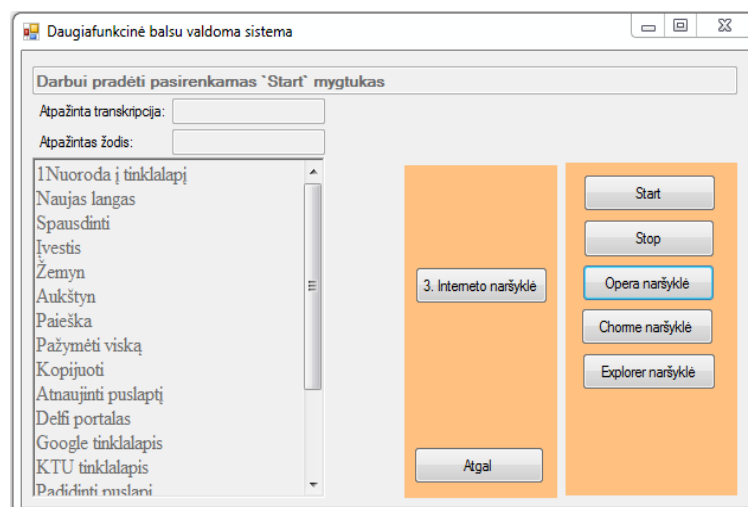
**USB sąsaja.** Pasinaudojus ja, komunikuojama su įrenginiais sujungtais per USB sąsają. Kad programa galėtų atlikti operacijas, susijusias su USB sąsaja, reikalingas atitinkamos bibliotekos įtraukimas į programą. Tai padaroma „Visual Studio 2015“ programoje atlikus šiuos veiksmus: programos kūrimo metu pasirenkame „project/add\_referece/browse/browse“; pasirenkama kompiuteryje instaliuota programavimo biblioteka „LibUsbDotNet.dll“; patvirtinamas pasirinkimas ir paspaudžiamas mygtukas „OK“. Kitas žingsnis, programos kode (4 PRIEDAS) įrašomos „using LibUsbDotNet“ ir „using LibUsbDotNet.Main;“ eilutės. Norint valdyti pasirinktą į USB sąsają įjungtą įrenginį, reikia sužinoti pasirinkto įrenginio „VID“ ir „PID“ adresus. USB įrenginio „VID“ ir „PID“ adresus, galima rasti apatinėje kompiuterio meniu juostoje paspaudus „Start/control\_panel/device\_manager“, tada paspaudus dešinį pelės klavišą ant „USB\_input\_device“ pasirenkama „properties/details/hardware\_ids“. „VID“ ir „PID“ adresai įrašomi į komandinę eilutę „public static UsbDeviceFinder MyUsbFinder = new UsbDeviceFinder(int vid, int pid);“.

Ištarus arba paspaudus mygtuką „2.USB sąsaja“, dingsta kitų pasirinkimo įrenginių mygtukai ir atsiranda trys papildomi mygtukai „Start“, „Stop“, „Komandos“ (2.3 pav). Ištarus arba paspaudus mygtuką „Komandos“ yra įkeliamos USB sąsajos garsyno gramatikos.



**2.3 pav. Daugiafunkcinės balsu valdomos sistemos atidarytas USB sąsajos langas**

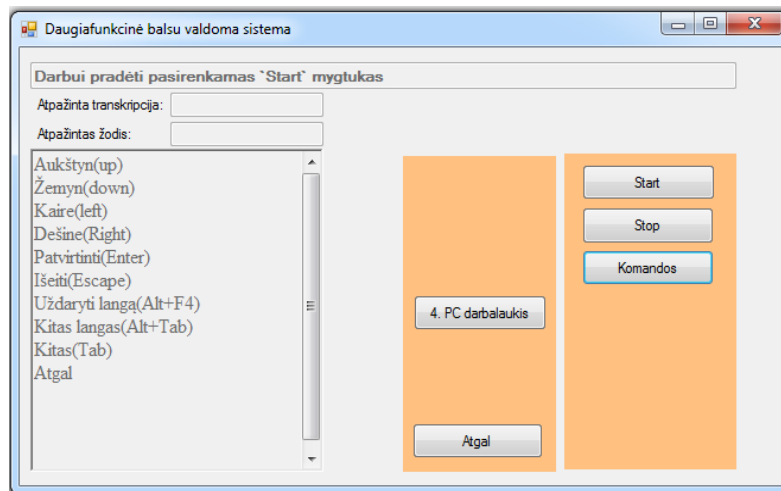
**Interneto naršyklė.** Ištarus arba paspaudus mygtuką „3. Interneto naršyklė“ dingsta kitų įrenginių pasirinkimo mygtukai ir atsiranda penki papildomi mygtukai „Start“, „Stop“, „Opera naršyklė“, „Explorer naršyklė“ ir „Chrome naršyklė“ (2.4 pav). Pasirinkus vieną iš šių interneto naršyklių: Opera; Chrome arba Internet Explorer, atsidaro pasirinkta naršyklė ir tada įkeliamos, pasirinktos naršyklės valdymo komandos ir jų gramatikos, kurių pagalba ta naršyklė yra valdoma balsu.



**2.4 pav. Daugiafunkcinės balsu valdomos sistemos atidarytas Interneto naršyklės langas**

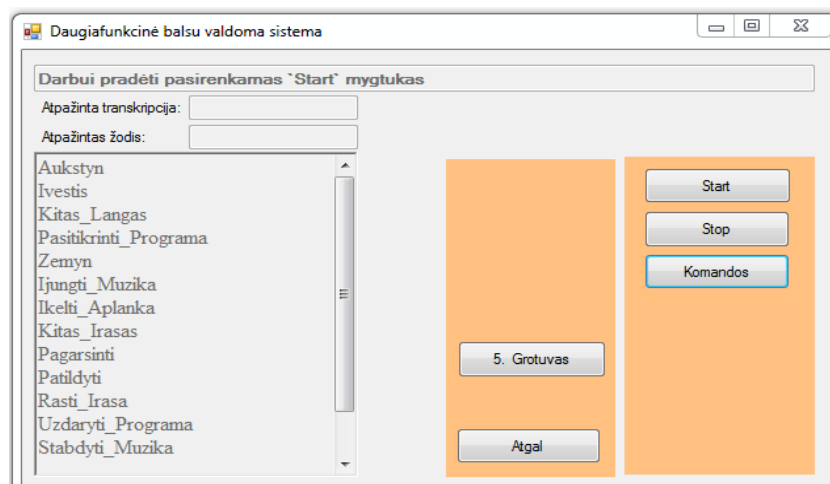
**Kompiuterio darbalaukio valdymas.** Ištarus komandą arba paspaudus mygtuką „4. PC darbalaukis“ dingsta kitų pasirinkimo įrenginių mygtukai ir atsiranda trys papildomi mygtukai „Start“, „Stop“ ir „Komandos“ (2.5 pav) ir atidaromas kompiuterio darbalaukis. Ištarus arba paspaudus mygtuką „Komandos“ yra įkeliamos PC darbalaukio valdymo gramatikos, kurias ištariant bus valdomas kompiuterio darbalaukis. Kompiuterio darbalaukio valdymas integruotas, suteikiant vartotojui galimybę balsu naviguoti tarp darbalaukio programų. 2.5 paveikslo komandų sąrašė

matomos išvestos, PC darbalaukiui, skirtos valdyti komandos, kurių pagalba vartotojas gali naviguoti tarp programų ar dokumentų esančių, kompiuterio darbalaukyje ar bet kokiame aplankale, esančiame kompiuteryje.



2.5 pav. Daugiafunkcinės balsu valdymo sistemos atidarytas PC darbalaukio langas

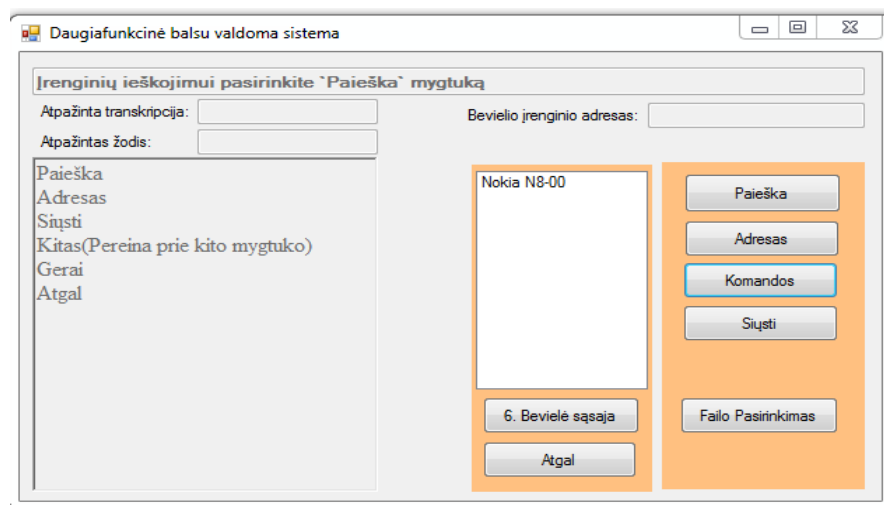
**PC grotuvas.** Ištarus arba paspaudus mygtuką „5.PC grotuvas“, dingsta kitų įrenginių pasirinkimo mygtukai ir atsiranda penki papildomi mygtukai „Start“, „Stop“, „Windows media player“, „AIMP2“ ir „Winnamp“ (2.6 pav). Pasirinkus vieną iš šių PC įrašytų grotuvo programų : Windows media player, Aimp2 ar Winnamp, atsidaro pasirinktas PC grotuvas, įkeliamos pasirinkto grotuvo valdymo komandos ir jų gramatikos, kurių pagalba bus balsu valdomas pasirinktas grotuvas.



2.6 pav. Daugiafunkcinės balsu valdomos sistemos atidarytas PC grotuvo langas

**Bevielė sąsaja.** Pasinaudojus ja yra komunikuojama su įrenginiais turinčiais įjungtą Bluetooth sąsają. Kad programa galėtų atlikti operacijas susijusias su Bluetooth sąsaja, reikalingas atitinkamų bibliotekų įkėlimas į programą. Tai atliekama „Visual Studio 2015“ programoje, atlikus šiuos

veiksmus: programos kūrimo metu pasirenkame „project“ tada „add\_referece“ toliau „browse“ ir galiausiai vėl „browse“; į kuriamą programą įkeliamos „InTheHand.Net.Forms.dll“, „InTheHand.Net.IrDA.dll“, „InTheHand.Net.ObjectExchange.dll“ ir „InTheHand.Net.Bluetooth.dll“ programavimo bibliotekos; patvirtinami pasirinkimai ir paspaudžiamas „OK“ mygtukas. Kitas žingsnis yra programos kode (4 PRIEDAS) įrašyti „using InTheHand.Net.Ports;“, „using InTheHand.Net.Bluetooth;“, „using InTheHand.Net;“ ir „using InTheHand.Windows.Forms;“ eilutes. Ištarus arba paspaudus mygtuką „6.Bevielė sąsaja“ dingsta kitų įrenginių pasirinkimo mygtukai ir atsiranda penki papildomi mygtukai „Paieška“, „Adresas“, „Komandos“, „Siųsti“ ir „Failo Pasirinkimas“ (2.7 pav). Atsiranda naujas interaktyvus langas, kuriame atvaizduojami visi rasti įrenginiai, su įjungtu „bluetooth“ moduliu. Atsiranda ir lentelė skirta pasirinkto įrenginio MAC adresui atvaizduoti. Išstartus arba paspaudus mygtuką „Komandos“ yra įkeliamos bevielės sąsajos komandų gramatikos. Kai kompiuteryje įjungtas „bluetooth“ modulis ir ištariama „Paieška“ komanda, sistema pradeda ieškoti aplinkinių įrenginių su įjungtu „bluetooth“ moduliu. Radus tuos įrenginius, iš sąrašo pasirenkamas įrenginys, kuriam bus atliekama duomenų persiuntimo operacija. Pasirinktam įrenginiui su komandos „Siųsti“ pagalba, išsiunčiamas programiškai pasirinktas dokumentas ( txt, pdf ir kitais formatais).

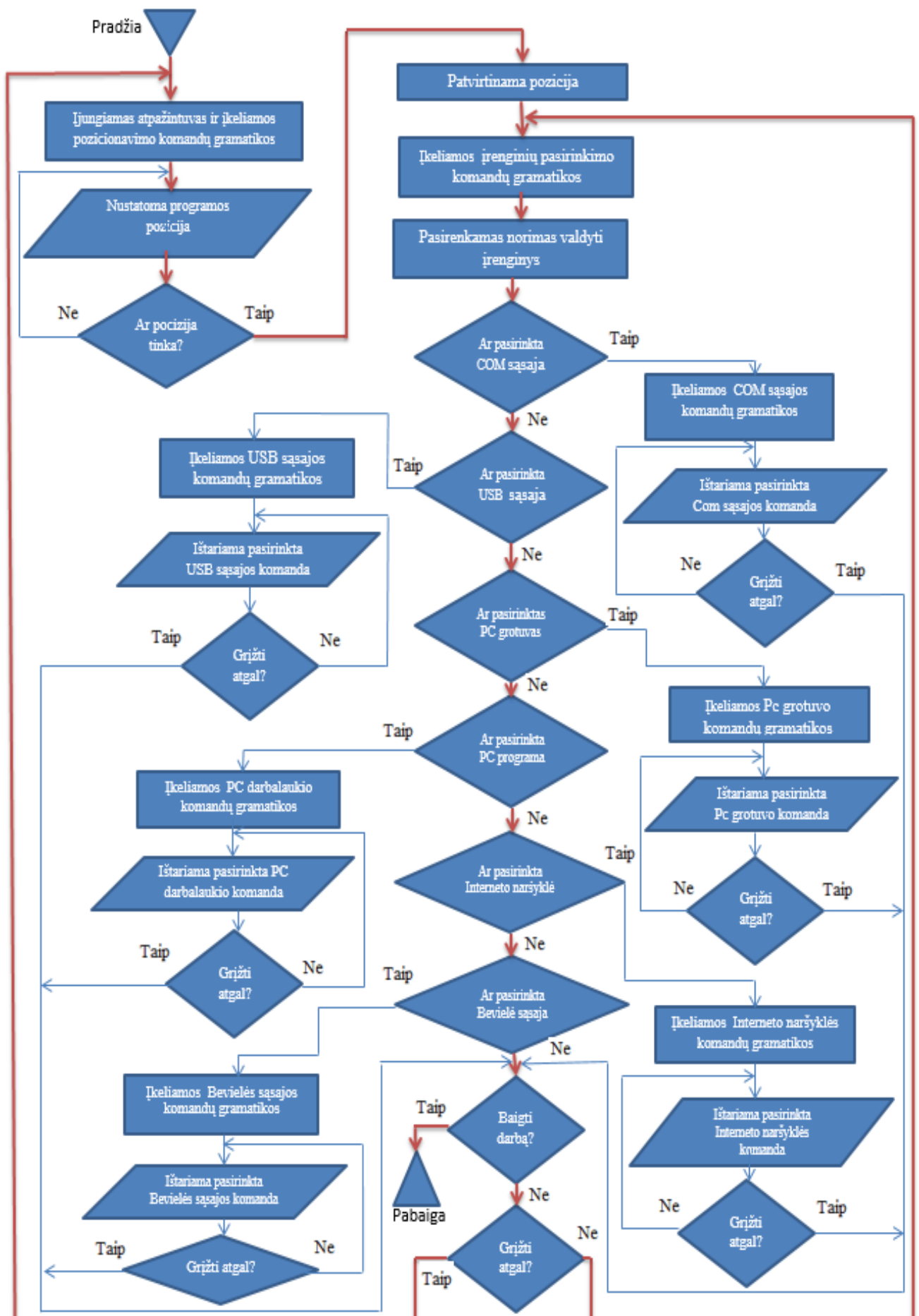


**2.7 pav. Daugiafunkcinės balsu valdymo sistemos atidarytas Bevielės sąsajos langas**

Daugiafunkcinės valdymo balsu sistemos algoritmas, iš kurio matoma programos darbo eiga, pateiktas 2.8 paveiksle. Aktyvavus programą yra įjungiamas balsu valdomas atpažintuvas ir yra įkeliamos pozicionavimo gramatikų komandos. Naudojantis tomis komandomis, nustatoma daugiafunkcinės sistemos programos pozicija, tai atlikus pereinama į įrenginių pasirinkimo etapą. Pasirinkus atitinkamą įrenginį yra įkeliamos to įrenginio ar sąsajos komandos, kurias ištariant valdomas pasirinktas įrenginys ar sąsaja. Bet kuriuo metu, kaip parodyta algoritme, galima grįžti į

prieš tai esantį etapą, iki pat ciklo pradžios t. y. pozicijos gramatikų užkrovimo. Pilnas programos kodas, kurį paveiksle pavaizduotas algoritmas atvaizduoja yra 4 PRIEDE.



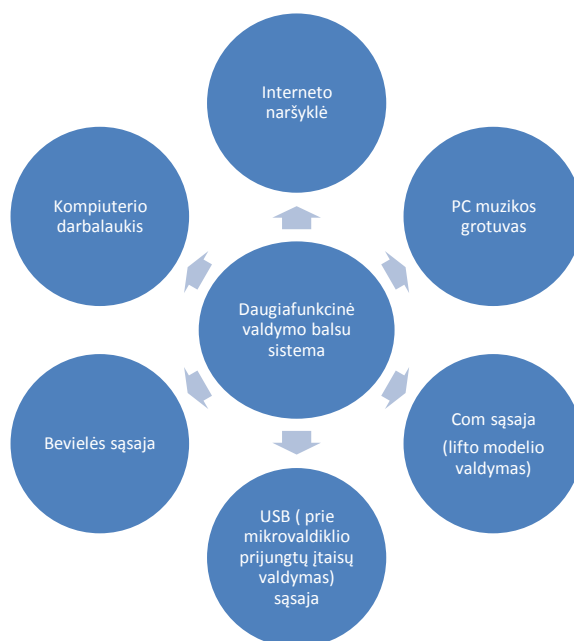


2.8 pav. Daugiafunkcinės sistemos darbo algoritmas.

## 2.2 Garsyno ir gramatikų paruošimas atpažintuvo tikslumo tyrimams

### 2.2.1 Garsyno paruošimas atpažintuvo tikslumo tyrimui

Kalbos signalų duomenų bazės, arba garsynai, sudaromi siekiant spręsti mokslo uždavinius ir įvertinti sukurtų sistemų kokybę. Kadangi daugiafunkcinė valdymo balsu sistemos programa bus valdomi skirtingi įrenginiai, su jiems atitinkamai valdyti parinktomis balso komandomis, kiekviena valdymo sistema turės savo atskirą gramatikų dokumentą, kurio viduje yra sukurtos komandų transkripcijos (žr 2.2.2 skyrių). 2.3 paveiksle pateikta daugiafunkcinė valdymo balsu sistema, skirta valdyti 6 modulius.



2.3 pav. Daugiafunkcinė valdymo balsu sistema

Kaip minima 1.1 skyriuje, balsu valdomos sistemos kuriamos valdyti atskirus įrenginius. Šitame darbe yra kuriama daugiafunkcinė balsu valdoma sistema gebanti valdyti kelis įrenginius ar sąsajas vienu metu. Buvusioje KTU kalbos signalų tyrimo mokslo laboratorijoje, studentų sukurti garsynai:

- Interneto naršyklės valdymo garsynas - 17 diktorių, 18 komandų po 20 ištarimų;
- Elektros prietaisų valdymo garsynas - 2 diktoriai, 11 komandų po 20 ištarimų;
- Lifto modelio valdymo garsynas - 7 diktoriai, 9 komandos po 20 ištarimų;
- PC muzikos grotuvo valdymo garsynas - 17 diktorių, 8 komandos po 20 ištarimų.

Kadangi su lifto modelio valdymo komandų garsynu, nebuvo atlikti HTK programiniu įrankiu paketu atpažnimo tikslumo tyrimai, naudojant PMM metodą ir kad galima panaudoti lifto modelio valdymo komandas ne tik lifto valdyme, tyrimui atlikti pasirinktas lifto modelio valdymo garsynas, turintis: 7 diktorių įrašus, kuriuose įrašytos 9 komandos: *antras, baigti darbą, ketvirtas, penktas, pirmas, šeštąs, sustok, trečias, važiuok*, ištartos po 20 kartų.

Atsižvelgus, kad esant mažam garsynui t. y. mažam diktorių kiekiui, arba mažam ištartų komandų kiekiui, sistemos balso komandų atpažinimo tikslumo tyrimas neparodys adekvačių rezultatų, tyrimui atlikti pasirinktas lifto garsynas buvo praplėstas keturių papildomų diktorių įrašais. Įrašai padaryti buvusioje KTU kalbos signalų tyrimo mokslo laboratorijoje.

Įrašai kuriami, ištariant 9 komandas po 20 kartų, perklausant ar įrašuose nėra pašalinių trikdžių, nekokybiškus įrašus perrašant. Visi įrašai išsaugomi „\*.voc“ formate, kartu sukuriamas „\*.zgl“ failas, kuriame yra informacija apie žodžių galus. „voc\_wav.exe“ programa, failai konvertuojami į „\*.wav“ tipą. Tokiu būdu buvo gauti 1980 wav ištartų balso komandų failai, kurie naudojami tyrimui Paslėptųjų Markovo Modelių metodu aprašytu 2.3 skyriuje ir su kitakalbiu atpažintuvu aprašytu 2.4 skyriuje. Visi vienuolikos diktorių įrašai perkelti į kompiuterį, veikiančią su *Windows XP* operacine sistema.

### 2.2.2 Gramatikų paruošimas atpažinimo tikslumo tyrimams

Išanalizavus literatūrą pastebėta, kad geriausi atpažinimo rezultatai gaunami su ispanišku kitakalbiu atpažintuvu, todėl ištartų komandų atpažinimo tikslumo nustatymo tyrimui su kitakalbiu atpažintuvu, pasirinktas ispaniškas atpažintuvas.

Garsyno paruošimas, tai gramatikų ir transkripcijų kūrimas, kuris siejamas su įdiktuoju garsynu ir jo atpažinimo gerinimu. Gramatikų kūrimas yra svarbus darbo eigos elementas, be gramatikų nebūtu galima atlikti tyrimo su kitakalbiu atpažintuvu ir nebūtu galima valdyti balsu sukurtos daugiafunkcinės sistemos. Gramatika – paruoštas „\*.grxml“ failas, kuriame nurodomos komandų transkripcijos nurodančios, ką reikia išvesti į ekraną įvykus balso komandos atpažinimui (žr. 2.1 skyrių). Pirminės gramatikos sudarymui, komandų transkripcijos buvo rašomos taip, kaip ir tariamos, tik nenaudojant lietuviškų simbolių. Nosinės ir ilgosios raidės virto paprastosiomis trumposiomis, pavyzdžiui., raidė Š virto į SH, raidė J → HY, Č → CH, C → TS ir kt. Žemiau pateikiama ištrauka iš pirminės lifto komandų gramatikų failo:

---

```
<item>
  <one-of>
    <item>sestas</item>
    <item>sheshtas</item>
  </one-of>
  <tag>$. _value = "sestas"</tag>
</item>
```

---

–

Atliekant tyrimą su kitakalbiu atpažintuvu, siekiant pagerinti ištartų komandų atpažinimo tikslumo rezultatus, savo nuožiūra buvo pakoreguotos transkripcijos. Pakoreguotos gramatikos pateikiamos 2 PRIEDE. Žemiau pateikiama ištrauka iš pakoreguoto lifto komandų gramatikų failo:

---

```
<item>
  <one-of>
    <item>sheshtas</item>
    <item>cheeshtas</item>
    <item>cheshtas</item>
    <item>sestas</item>
  </one-of>
  <tag>$.value = "sestas"</tag>
</item>
```

---

Be lifto komandų gramatikų failo „1.liftokomandos.grmx“ daugiafunkciniai sistemai valdyti balsu, sukurti šie gramatikų failai: „0.1\_Pozicionavimo.grmxl“; „0.2\_Sasaju\_Pasirinkimas.grmxl“; „2\_USB\_komandos.grmxl“; „3.1\_Opera\_Narsykle.grmxl“; „3.2\_Chrome\_Narsykle.grmxl“; „3.3\_Explorer\_Narsykle.grmxl“; „4\_PC\_komandos.grmxl“; „5\_Grotuvo\_komandos.grmxl“; „6\_Bevieles\_komandos.grmxl“.

### 2.3 Tyrimo taikant HTK modelius metodika

Šnekamajai kalbai atpažinti, atliekamas tyrimas siekiant gauti kalbos atpažinimo tikslumą naudojant HTK programinių įrankių paketą. HTK žodžių atpažinime grįstais paslėptaisiais Markovo modeliais (PMM), sistema pirmiausiai apmokoma, o paskui naudojama balso komandoms atpažinti. PMM apmokymo metu visiems žodyno žodžiams, pagal parinktas žodžių transkripcijas, sudaromi tikimybiniai akustiniai modeliai ir apskaičiuojami kiekvieno akustinio modelio tikėtumo įvertčiai. Sistema iš žodyno pateikia, kaip atpažinti tą žodį, kurį reprezentuoja modelis, kurio tikėtumo įvertis yra didžiausias. Šnekamosios kalbos atpažinimo sistemos, kurioje kalbos atpažinimui taikomas PMM metodas, apmokymui rekomenduojama naudoti kitų diktorių garsyno įrašus nei atpažinimo kokybės testavimui.

HTK programinio paketo programų aktyvavimui naudojami „batch“ (trumpinys „\*.bat“) failai. Šių failų pagrindinė paskirtis yra automatizuoti tam tikrą procesą. Failų viduje parašoma tam tikra veiksmų seka, pvz. „HCOPY -C CONFIG -S failai.scp“, kuri suveiks, kai bus aktyvuotas „batch“ failas. HTK paketo programos yra naudojamos nuosekliai, veiksmai atliekami vienas po kito.

Taigi šiame skyriuje aprašomas HTK programinių įrankių naudojimas balso komandų atpažinimo modeliavimui. Naudojantis HTK programiniu įrankių paketu, darbo eiga susideda iš 7 nuoseklių etapų [6][9]. Pirmi šeši etapai apima komandų akustinių modelių apmokymą, o septintasis etapas apima modelių testavimą.

**1 ETAPAS** - *gramatikos failo kūrimas*. Sukuriamas, HTK programų paketui reikalingas, gramatikų failas „gram.dikt“. Jame aprašomos norimos atpažinti balso komandos ir jų tvarka atpažinimo procese. Šiame tyrime naudojamų lifto valdymo komandų aprašymas „gram.dikt“ faile atrodo taip:

```
„$word=antras|baigtidarba|ketvirtas|penktas|pirmas|sestas|sustok|treacias|vaziuok;  
($word)“
```

Faile „*gram.dikt*“ komandos pavadinimai yra atskiriami specialiu simboliu (/). Taip yra nurodoma, kad bus tariami atskiri žodžiai: *antras, baigtidarba, ketvirtas, penktas, pirmas, šeštas ir t.t.*

Taip pat „*gram.dikt*“ faile gramatikos gali būti pateikiamos ir kitaip. Jeigu vieno ištarimo metu norima pasakyti iškarto du žodžius iš pateikto sąrašo - „penktas, šeštas“, gramatikos failo turinys atrodo taip:

```
„$word=antras|baigtidarba|ketvirtas|penktas|pirmas|sestas|sustok|treacias|vaziuok;  
$word $word“
```

„*gram.dikt*“ faile gramatikos gali būti pateikiamos ir kitaip. Jeigu vieno ištarimo metu norima pasakyti iškarto du žodžius iš pateikto sąrašo ir juos sujungti jungtuku „penktas ir šeštas“, gramatikos failo turinys atrodo taip :

```
„$word=antras|baigtidarba|ketvirtas|penktas|pirmas|sestas|sustok|treacias|vaziuok;  
$word1 = ir  
$word $word $word“
```

Šiame darbe balso komandų atpažinimo tyrimuose naudojamas pirmasis gramatikos variantas.

**2 ETAPAS** - *žodžių modelių sudarymui reikalingų failų paruošimas*. HTK programiniai įrankiai automatiškai sukuria tekstinį failą „*wordnet.txt*“. Šiam failui sukurti naudojama programa „*Hparse.exe*“, kurios *batch* faile įrašyta komandinė eilutė: *HParse.exe gram.dict wordnet.txt*. Pirmoji komanda eilutėje - „*Hparse.exe*“ nurodo, kad bus naudojama ši programa, taip pat, bus naudojamas failas „*gram.dict*“ – prieš tai jau sukurtas žodžių sąrašas, o „*wordnet.txt*“ - programos automatiškai sukuriamas failas.

**3 ETAPAS** - *požymių kūrimas*. Turint „*gram.dikt*“ ir „*wordnet.txt*“ failus toliau sukuriamas, požymiams sukurti ir apskaičiuoti reikalingas failas - „*failai.scf*“. Tai failas, kuriame bus nurodomos kompiuterio direktorijos, kur bus sukuriami ir talpinami požymių failai. Toliau pateikiama lifto valdymo komandų garsynams naudojamų „*failai.scf*“ failų turinių dalis:

---

```
D:\-Dokumentai-\Desktop\komandosA\irasai\0A00.wav D:\-Dokumentai-\Desktop\komandosA\irasai\0A00.mfc  
D:\-Dokumentai-\Desktop\komandosA\irasai\0A01.wav D:\-Dokumentai-\Desktop\komandosA\irasai\0A01.mfc  
D:\-Dokumentai-\Desktop\komandosA\irasai\0A02.wav D:\-Dokumentai-\Desktop\komandosA\irasai\0A02.mfc  
.....  
D:\-Dokumentai-\Desktop\komandosA\irasai\0A17.wav D:\-Dokumentai-\Desktop\komandosA\irasai\0A17.mfc  
D:\-Dokumentai-\Desktop\komandosA\irasai\0A18.wav D:\-Dokumentai-\Desktop\komandosA\irasai\0A18.mfc  
D:\-Dokumentai-\Desktop\komandosA\irasai\0A19.wav D:\-Dokumentai-\Desktop\komandosA\irasai\0A19.mfc
```

---

Pavyzdyje matoma, kad kiekviena komandinė eilutė susideda iš direktorijos, kurioje yra garsyno diktorių balso įrašas (D:\-Dokumentai\Desktop\komandosA\irasai\0A00.wav) ir direktorijos, kurioje bus sukuriamas to įrašo \*.mfc požymių failas (D:\-Dokumentai\Desktop\komandosA\irasai\0A00.mfc), komandosA simbolizuoja pirmąjį diktorių, šitaip kiekvieno diktoriaus atveju daromas naujas katalogas ir taip sukuriama 11 katalogų. Kiekvienas požymių failas yra sukuriamas pagal to paties diktoriaus, tą patį, garsinį failą. Požymių failai yra reikalingi, nes atpažinimo sistema naudoja ne pačius \*.wav formato garso įrašus, o tik jų požymius. Sukurti požymių failai tada perkeliama į atskirą katalogą pavadinimu „pozymiai“. Požymių failams sukurti naudojama programa „Hcopy.exe“. Jos vykdymui reikia turėti paruošus tokius failus - failai.scf, failas kuriame nurodomos direktorijos ir „CONFIG“, kuris yra informacinis failas su pateikta informacija apie požymių skaičiavimo būdus. „CONFIG“ failo turinys atrodo taip:

---

```
SOURCEKIND=WAVEFORM
SOURCEFORMAT=WAVE
TARGETKIND=MFCC_D_A_E
SAVEWITHCRC=F
SOURCERATE=625
TARGETRATE=100000.0
WINDOWSIZE=250000.0
USEHAMMING=T
PREEMCOEF=0.96
NUMCEPS=12
NUMCHANS=20
```

---

**4 ETAPAS** – *modelių failų parengimas*. Turint požymius, tolimesniame etape yra parengiami ir pildomi modelių failai. Šie modelių failai yra reikalingi atlikti šnekamai kalbai atpažinti po sistemos apmokymo. Modelių failų sukuriama tiek kiek yra komandų. Devynių komandų modelių failai:

---

```
hmm_antras
hmm_baigtidarba
hmm_ketvirtas
hmm_penktas
hmm_pirmas
hmm_sestas
hmm_sustok
hmm_trecias
hmm_vaziuok
```

---

Failų pavadinimai turi būti tikslūs ir be klaidų atitiki naudojamas valdymo komanda, nes juose yra griežtos struktūros informacija, apie tam tikrą komandos pavadinimą. Kiekviename modelių faile

nurodoma, kiek būsenų ar Gauso mišinių bus naudojama modeliuojant atitinkamą žodį. Aprašomi vidurkių bei dispersijų vektoriai, perėjimų tikimybių reikšmės ir kiti parametrai. 2.4 paveiksle pateikiamas komandos pavadinimo „*antras*“ modelių failo, turinys.

```

~o
<STREAMINFO> 1 39
<VECSIZE> 39<NULLD><MFCC_E_D_A>
~h "hmm_antras"
<BEGINHMM>
<NUMSTATES> 6
<STATE> 2
<MEAN> 39
-1.635420e+001 -7.481701e+000 -3.241516e+000 -6.746332e+000 -1.905021e+000 -2.098491e-001 -3.556307e+000 -1.653795e+000 -3.160717e+000 -7.614004e-002 -1.860593e+000 -2.659124e+000
5.936241e-001 1.045115e-001 -9.938418e-002 1.852256e-003 -1.078802e-001 -7.139887e-002 -5.955098e-002 -1.166002e-001 -1.363539e-002 5.293483e-002 -1.531312e-002 1.978809e-002 1.269146e-003
1.283406e-002 5.581629e-002 -1.794550e-002 1.215951e-003 -2.029217e-002 -3.292636e-002 -1.788949e-002 -6.102315e-002 -6.571918e-003 2.086528e-002 -6.610165e-003 -8.879054e-003 -7.258497e-
003 1.663400e-003
<VARIANCE> 39
6.120396e+001 4.128733e+001 3.451609e+001 3.580319e+001 3.172113e+001 4.763743e+001 3.663995e+001 3.402687e+001 3.737448e+001 2.752102e+001 1.899535e+001 1.969127e+001 9.426043e-002
2.934275e+000 2.367974e+000 2.063026e+000 2.087210e+000 1.616394e+000 2.117690e+000 2.130674e+000 2.043434e+000 1.970868e+000 1.790762e+000 1.261116e+000 1.263335e+000 4.924380e-003
4.322110e-001 3.162542e-001 3.185689e-001 3.113748e-001 2.609292e-001 3.112812e-001 3.507734e-001 3.450337e-001 2.746606e-001 3.081366e-001 2.438419e-001 2.339956e-001 6.489430e-004
<GCONST> 9.249604e+001
<STATE> 3
<MEAN> 39
-1.318572e+001 -1.092466e+001 -3.567029e+000 -8.906710e+000 -5.294247e+000 -3.115153e+000 -9.918097e+000 -8.572701e-001 3.315541e+000 1.189654e+000 -3.244779e+000 -2.416975e+000
7.924251e-001 -2.989916e-001 2.716010e-001 -3.967401e-003 1.676212e-001 2.606258e-001 -1.957146e-002 1.647976e-001 9.175546e-002 -2.698870e-002 -2.544180e-002 -5.323623e-002 -2.549073e-002
1.242388e-002 -2.949355e-001 9.700260e-002 -1.024206e-002 1.149120e-001 1.137591e-001 4.025037e-002 2.146648e-001 2.481181e-002 -5.191229e-002 8.701910e-003 7.540419e-002 5.408960e-002 -
1.930665e-003
<VARIANCE> 39
3.580811e+001 2.824692e+001 2.052508e+001 3.091581e+001 2.972331e+001 2.018601e+001 4.127357e+001 2.837565e+001 1.784853e+001 3.057835e+001 4.629311e+001 2.164336e+001 1.337320e-002
5.667546e+000 2.570634e+000 1.403797e+000 3.105717e+000 2.898274e+000 2.067309e+000 4.647258e+000 2.446209e+000 1.325656e+000 2.214878e+000 2.729871e+000 1.644816e+000 1.444710e-003
5.731558e-001 3.172156e-001 2.745804e-001 4.429399e-001 4.641698e-001 3.450713e-001 5.774266e-001 4.449961e-001 2.547179e-001 4.131224e-001 4.264395e-001 3.026484e-001 3.458421e-004
<GCONST> 9.255043e+001
<STATE> 4
<MEAN> 39
-2.470575e+001 3.776916e-001 -1.676054e+000 -2.151704e-001 1.914982e+000 -1.285746e+000 1.714051e+000 6.886468e-002 -3.306908e-001 5.408483e-001 -1.055583e+000 -4.701853e-001 5.259013e-001
-7.239407e-003 2.515301e-003 1.738524e-002 6.713683e-002 -2.444293e-002 9.396279e-002 9.875386e-002 -3.500801e-003 -4.676116e-002 4.179469e-002 2.350975e-002 4.301067e-002 -1.334955e-002
7.581310e-002 -3.292413e-002 2.233062e-003 -3.639250e-002 -2.311831e-002 -3.601949e-003 -4.537170e-002 -1.322988e-002 7.755023e-005 2.691251e-003 -2.245894e-002 -1.424444e-002 -1.082953e-
003
<VARIANCE> 39
4.148885e+001 3.061886e+001 2.576120e+001 2.764822e+001 2.228556e+001 2.689502e+001 1.900113e+001 1.432310e+001 1.696751e+001 1.265472e+001 9.969807e+000 7.840863e+000 6.985159e-002
1.399555e+000 1.302544e+000 1.089158e+000 1.225884e+000 1.213734e+000 1.278880e+000 1.260468e+000 1.272284e+000 1.154299e+000 1.009454e+000 9.119945e-001 7.514822e-001 1.212041e-003
2.275359e-001 2.073549e-001 1.936593e-001 2.340265e-001 2.361930e-001 2.499457e-001 2.643826e-001 2.593025e-001 2.320609e-001 2.062221e-001 1.987312e-001 1.591521e-001 1.676338e-004
<GCONST> 7.247847e+001
<STATE> 5
<MEAN> 39
-2.756773e+001 -4.196552e+000 -3.883252e+000 -1.323063e+000 -1.232651e-003 -1.224757e+000 1.604688e+000 -5.457349e-001 -1.444138e-001 1.504017e+000 -1.986233e+000 1.657649e-001 2.275809e-
001 -7.648911e-002 -3.981552e-002 -1.662688e-002 -2.992043e-002 -4.695912e-002 -1.344791e-002 -7.876813e-002 -6.394372e-002 -2.538627e-003 -5.452407e-003 -1.976138e-002 -2.387349e-002 -
1.008535e-003 1.385147e-002 9.915282e-003 1.313461e-002 8.234767e-003 9.245361e-003 3.810984e-003 9.927879e-003 1.127801e-002 5.141106e-003 -5.243152e-003 -1.689574e-004 -4.119486e-004
4.229894e-004
<VARIANCE> 39
3.964005e+000 3.075687e+000 6.579390e+000 6.206078e+000 7.482400e+000 6.622754e+000 1.149526e+001 9.062066e+000 9.797117e+000 9.685256e+000 8.580899e+000 6.137663e+000 1.719897e-002
1.043460e-001 2.044517e-001 3.488092e-001 4.221535e-001 5.794712e-001 6.739652e-001 7.358613e-001 7.340000e-001 7.611475e-001 6.951320e-001 7.400761e-001 6.022880e-001 2.609311e-005
1.619741e-002 3.665504e-002 6.310803e-002 7.875957e-002 1.130026e-001 1.269752e-001 1.427574e-001 1.424323e-001 1.438851e-001 1.313062e-001 1.477067e-001 1.183688e-001 4.791806e-006
<GCONST> 3.069950e+001
<TRANSP> 6
0.000000e+000 1.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000 9.752545e-001 2.474552e-002 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000 0.000000e+000 9.455945e-001 5.440552e-002 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000 0.000000e+000 0.000000e+000 9.731176e-001 2.688232e-002 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 9.506513e-001 4.934864e-002 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
<ENDHMM>

```

## 2.4 pav. Balso komandos „hmm\_antras“ modelių failo turinys

Modelių faile esančią informaciją galima išskirti į dvi dalis: bendrą informaciją apie modelių failą ir modeliavimo duomenis. Kiekviename iš šių failų, nurodomas būsenų skaičius reikalingas žodžiui sumodeliuoti, kiek požymių bus naudojama požymių vektoriuje. Bendroji informacija apie modelių failą pateikiama pirmose failo eilutėse.

Modelių failo informacija yra aprašoma tokiose eilutėse:

- ~o – pradžios simbolis;
- <STREAMINFO> 1 39 – skaičius 39 nurodo, kad bus naudojamas 39 požymių rinkinys, o skaičius 1 – kad jie bus išdėstyti vienoje eilutėje;

- $\langle VECSIZE \rangle$  39  $\langle NULLD \rangle$   $\langle MFCC\_E\_D\_A \rangle$  – šioje eilutėje nurodomas vektoriaus dydis - tie patys 39 požymiai, MFCC\_E\_D\_A reiškia, kad bus naudojami MFCC požymis ir E – energijos, D – delta ir A – delta delta požymio koeficientai;
- $\sim h$  "hmm\_\*\*" – tai modelių failo pavadinimas. Norint atlikti tikslų pasirinktosios komandos modeliavimą šis pavadinimas turi sutapti su pagrindiniu komandos modelių failo pavadinimu;
- $\langle BEGINHMM \rangle$  – modelio pradžios žyma;
- $\langle NUMSTATES \rangle$  - nurodomas modelių faile naudojamas būsenų skaičius. Buvo atliekami atpažinimo tikslumo tyrimai naudojant modelius su skirtingu būsenų skaičiumi. Būsenų skaičius buvo parenkamas atsižvelgiant į komandos raidžių skaičių ir pridėdant papildomą būsenų skaičių (+1, +2, ir t.t.). Pavyzdžiui, jeigu skaičių "du" sudaro dvi raidės ir atliekamas atpažinimo tikslumo tyrimas naudojant dešimt papildomų būsenų, tai „ $\langle NUMSTATES \rangle$ “ eilutėje bus rašoma – 12 (2+10). Tyrimo rezultatai naudojant skirtingą komandos modelio būsenų skaičių aprašomi 3.1 skyriuje.
- $\langle STATE \rangle$  2 – modelis visada prasideda nuo antros būsenos;
- $\langle NUMMIXES \rangle$  - nurodomas modelių daile naudojamų Gauso mišinių skaičius. Buvo atliekami atpažinimo tikslumo tyrimai naudojant modelius su skirtingu Gauso mišinių skaičiumi ir taip su visais diktoriais. Tyrimo rezultatai naudojant skirtingą komandos modelio Gauso mišinių skaičių aprašomi 3 skyriuje;
- $\langle MIXTURE \rangle$  - nurodomas pirmasis Gauso mišinys ir jo vertė;
- $\langle MEAN \rangle$  39 -1.653420e+001 -7.481701e+000 ... 1.663400e-003 – požymių vidurkiai, parenkami intuityviai, kurie modelių mokymo metu bus perskaičiuojami automatiškai;
- $\langle VARIANCE \rangle$  39 6.120396e+001 3.128733e+001 ... 6.489430e-004 – požymių dispersijos, parenkamos intuityviai, kurios apmokymo metu bus perskaičiuotos automatiškai;
- $\langle GCONST \rangle$  9.249604e+001 – būsenos stiprinimo koeficientas, parenkamas nuo 0 iki 1.

Tokie veiksmai atliekami su visomis būsenomis iki paskutiniosios. Tokiu būdu apskaičiuojamos tikimybės būti pirmoje, antroje, trečioje ir t.t. būsenose.  $\langle TRANSP \rangle$  - „*transponentė*“ - paskutinė būseną. Šioje būsenoje yra sudaroma tikimybių matrica, kuri atrodo taip:

- $\langle TRANSP \rangle$  6

```
0.000000e+000 1.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000 9.752545e-001 2.474552e-002 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000 0.000000e+000 9.455945e-001 5.440552e-002 0.000000e+000 0.000000e+000
0.000000e+000 0.000000e+000 0.000000e+000 9.731176e-001 2.688232e-002 0.000000e+000
0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 9.506513e-001 4.934864e-002
0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
```



Prie nuorodos į transponentę esantis skaičius žymi paskutinę būseną. Šioje tikimybių matricoje, eilučių ir stulpelių skaičius turi sutapti, darbo metu kuriant naujus modelius į tai ir reikėjo atsižvelgti. Tikimybės šioje matricoje juda į priekį - kiekvienoje eilutėje yra aprašoma tikimybė pereiti į sekančią būseną arba likti toje pačioje. Vykdamas paruoštų modelių apmokymą, šios tikimybių vertės yra automatiškai perskaičiuojamos. Požymių vidurkių (<MEAN>) bei dispersijų (<VARIANCE>) vertės taip pat yra automatiškai perskaičiuojamos modelių apmokymo metu;

- <ENDHMM> modelio pabaigos žyma;

Turint pilnai paruoštą vieną modelio failą, taupant laiką patogiu jį taikyti kaip šabloną kitiems modeliams atitinkamai jį modifikuojant - keičiant pavadinimą, būsenų skaičių ir pridėdant arba pašalinant atitinkamas būsenas ir tikimybių matricos eilutes bei stulpelius, keičiant Gauso mišinių skaičių.

Atliekant tyrimus, naudojant skirtingą būsenų ir Gauso mišinių kiekį, šiais modelių failais buvo įvairiai varijuojama (visų variacijų rezultatai pateikiami 3 skyriuje).

Paruošus modelių apmokymui reikalingus duomenis, prieš apmokymą, susikuriami apmokymui reikalingi papildomi *script* (\*.scp) failai, kurie reikalingi nurodant direktorijas, kuriose yra saugomi požymių (\*.mfc) failai. Kiekvienos komandos pavadinimu yra sukuriamas atskiras *script* failas, pavadinimu „trans\_\*.scp“. Tokia eiga failai sukurti visoms atpažinimui reikalingoms komandoms:

Devynių komandų modelių failai:

---

*trans\_antras*  
*trans\_baigtidarba*  
*trans\_ketvirtas*  
*trans\_penktas*  
*trans\_pirmas*  
*trans\_sestas*  
*trans\_sustok*  
*trans\_trecias*  
*trans\_vaziuok*

---

Diktorius A ištarų komandų failo - „trans\_antras.scp“ turinio fragmentas atrodo taip:

---

D:\-Dokumentai-\Desktop\komandosA\pozymiai\0A00.mfc  
D:\-Dokumentai-\Desktop\komandosA\pozymiai\0A01.mfc  
D:\-Dokumentai-\Desktop\komandosA\pozymiai\0A02.mfc  
.....  
D:\-Dokumentai-\Desktop\komandosA\pozymiai\0A19.mfc

---

Kiekvienas iš šių failų susideda tik iš jo pavadinime parašytų komandų pavadinimų požymių. Pavyzdžiui sistemą tikrinsime su diktoriumi B tai požymių kataloge bus surašytos visų diktorių išskyrus diktorius B ištarto, šiuo atveju žodžio „antras“, komandos.

**5 ETAPAS** – *modelių apmokymas*. Paruošus šiuos failus atliekamas žodžių modelių apmokymas, naudojant „*HRest.exe*“ programą. Kuri naudoja jau paruoštus „*trans\_\*.scp*“ script failus, kuriuose išvardinti žodžių junginio požymių failai. Pasinaudoję „*HRest.exe*“ programa gauname apmokyta modelio failą „*\*.hmm*“, kuri galima naudoti atpažinimo metu. Taip apmokomi visų žodinių komandų modeliai.

„*HRest.bat*“ faile užrašomos tokios komandinės eilutės atitinkamai komandai atskirai:

---

*HRest.exe -T 1 -S trans\_antras.scp hmm\_antras*

---

**6 ETAPAS** – *duomenų paruošimas testavimui*. Atlikus modelių apmokimą reikia atitinkamus failus paruošti ir apmokyta modelių taikymui balso komandoms atpažinti testavimui. Tam reikia susikurti papildomus *script* failus, kurie bus reikalingi kuriant naujus požymius testavimui. Visų pirma, sukuriamas *script* failas pavadinimu „*test.scp*“, kurio turinys atrodo taip pat, kaip pačioje pradžioje apmokymui kurto failo „*failai.scp*“ turinys, skirtumas tik tas, kad šiuo atveju požymių failus reikia sukurti, testavimui paskirtiems (kitiems nei naudojamiems apmokyme) diktorių, garso įrašams. Tai jei sistema testuosime su A diktoriumi tai „*test.scp*“ išstartų komandų požymiai bus tik A diktoriaus. Sukūrus šį failą ir suredagavus direktorijas, yra dar kartą naudojama programa „*Hcopy.exe*“, kuri sukuria naujus, testavimui skirtus požymių failus. Sukurti požymių failų pavadinimai yra surašomi į dar vieną *script* failą „*test\_mfc.scp*“. Atliekamas tas pats procesas kaip ir aprašant „*trans\_\*.scp*“ failus, skirtumas tik tas, kad dabar visi požymių failų pavadinimai yra surašomi į vieną *script* failą.

Toliau sukuriamas failas „*hmmlist*“, kuriame surašomi visi modelių failų pavadinimai. Sukūrus „*hmmlist*“ failą, toliau reikia sukurti tiriamų žodžių tekstinį failą, „*dikt.txt*“. Šio failo paskirtis yra nurodyti atpažinimo sistemai, kokius žodžius reikia atvaizduoti kompiuterio ekrane, kai bus atpažintas žodžio panašiausias modelis. Šio failo turinys atrodo taip:

---

<i>Antras</i>	<i>hmm_antras</i>
<i>Baigtidarba</i>	<i>hmm_baigtidarba</i>
<i>Ketvirtas</i>	<i>hmm_ketvirtas</i>
<i>Penktas</i>	<i>hmm_penktas</i>
<i>Pirmas</i>	<i>hmm_pirmas</i>
<i>Sestas</i>	<i>hmm_sestas</i>
<i>Sustok</i>	<i>hmm_sustok</i>
<i>Trecias</i>	<i>hmm_trecias</i>
<i>Vaziuok</i>	<i>hmm_vaziuok</i>

---

**7 ETAPAS** – *Modelių testavimas*. Atlikus visus paminėtus veiksmus yra atliekamas atpažinimui apmokyta žodžių modelių testavimas. Testavimui atlikti yra naudojama programa „*Hvite.exe*“. Programa atlikusi testavimą, automatiškai sukuria gautųjų atpažinimo rezultatų „*results*“ failą. Šiame

faile, gautieji atpažinimo tikslumo rezultatai, yra pateikiami eilutėmis, kur kiekviena eilutė parodo diktoriaus, tam tikro komandos pavadinimo atpažinimo rezultatą. Gautieji rezultatai yra surašomi tokia tvarka, kokia buvo pateikti faile „*test\_mfc.scp*“. Paskutinis punktas yra paleisti „Hresults.bat“ programą kuri sukure pirmo diktoriaus kuriuo testuojame atveju „rezultataiA“ failą kuriame pateiktas visų išstartų komandų tikslumas viena kitos atžvilgiu, jų tikslumo vidurkiai ir bendras tikslumo rezultatas

Naudojant HTK programinį paketą buvo atliktas tyrimas balso komandų atpažinimo tikslumui įvertinti naudojant paslėptųjų Markovo grandinių metodą tiriamųjų komandų žodžių modeliams sukurti. Lifo valdymo komandų garsyno atpažinimo tikslumo tyrimų rezultatai aprašomi 3 skyryje.

## **2.4 Komandų atpažinimo tyrimo naudojant žodžiais grįstus modelius metodika**

Visi atpažinimo tikslumo tyrimai atlikti naudojant 11 diktorių (4 moterų ir 7 vyrų) lifto valdymo komandų garsyno įrašus. Kurie tyrimo metu naudojami sistemos apmokymui ir testavimui. Apmokymo ir testavimo metu komandų modelių atpažinimo tikslumai nustatomi keičiant parametru vertes, esančias HTK programiniame pakete.

Kiekviena balso komanda modeliuojama viena ištinine Markovo grandine, sukuriant komandoms atitinkamus PMM, kuriuose redaguojami pagrindiniai parametrai, galintys įtakoti balso komandų atpažinimo tikslumą.

### **2.4.1 Papildomų būsenų įterpimas į žodžių modelius**

Atpažinimo tikslumo tyrimui atlikti, su skirtingų pridėtų būsenų skaičiumi, būsenų skaičius parinktas lygus balso komandą sudarančių fonetinių elementų skaičiui, esančiam intervale imtinai nuo 6 iki 21. Valdymo komandos „*Antras*“ modelis turi 6 vidines būsenas, o pvz., valdymo komandos „*Baigti darbą*“ modelis turi 11 būsenų, tokia eiga likusiuose balso komandų modeliuose atitinkamai parenkami būsenų skaičiai. Toliau būsenų skaičiai parenkami, prie pirminio būsenų skaičiaus palaipsniui pridedant 2, 4, 6, 8 ir 10 papildomas būsenas, kaip parodyta 2.1 lentelėje.

## 2.1 lentelė. Lifto modelio valdymo komandų vidinės būsenos

Komandos	Būsenos					
	Pridėjus „0“ būsenų	Pridėjus „2“ būsenas	Pridėjus „4“ būsenas	Pridėjus „6“ būsenas	Pridėjus „8“ būsenas	Pridėjus „10“ būsenų
Antras	6	8	10	12	14	16
Baigti darbą	11	13	15	17	19	21
Ketvirtas	9	11	13	15	17	19
Penktas	7	9	11	13	15	17
Pirmas	6	8	10	12	14	16
Šeštas	6	8	10	12	14	16
Sustok	6	8	10	12	14	16
Trečias	7	9	11	13	15	17
Važiuk	7	9	11	13	15	17

### 2.4.2 Papildomų Gauso mišinių įterpimas į žodžių modelius

Lifto modelio komandų atpažinimo tikslumo tyrimas atliekamas keičiant būsenų ir Gauso mišinių skaičių. Keičiant būsenas ir Gauso mišinius gauname didesnę kiekį duomenų, o tai padeda nuosekliau ir tiksliau atlikti tyrimą. Tyrimas pradedamas nepridedant jokių būsenų prie PMM, keičiami tik Gauso mišinių skaičiai - 2, 4, 6 ir 8. Tyrimas kartojamas keičiant PMM būsenų kiekį – 2, 4 ir 6, Gauso mišiniai keičiami kaip ir 0 būsenų atveju.

Atlikto komandų atpažinimo tikslumo tyrimo, naudojant PMM, rezultatai surašyti 3 skyriuje.

### 2.5 Kitakalbis atpažintuvas

Microsoft Windows operacinėje sistemoje yra įdiegta šnekamosios kalbos atpažinimo sistema - „*Microsoft Speech Recognizer*“. Ši sistema gali atpažinti daugumą plačiausiai pasaulyje vartojamų kalbų – anglų, ispanų, prancūzų ir kt., tačiau lietuvių kalbos atpažinimui ji dar nepritaikyta.

Kaip minima 2.2.2 skyriuje, šiame darbe lietuviškų lifto valdymo komandų atpažinimo tikslumui tirti pasirinktas ispaniškas kalbos atpažintuvas – „*Microsoft Speech Recognizer 8.0 for Windows (Spanish-Spain)*“ – esantis Microsoft Windows operacinėje sistemoje.

MS Windows operacinėje sistemoje norint susirasti, naudoti, apmokyti ir pasirinkti norimus šnekamosios kalbos atpažinimo sistemos parametrus, reikia susirasti pagrindinį atpažinimo sistemos langą tokia veiksmų seka: atidaromas „*Control Panel*“ langas >> pasirenkame „*Speech recognition*“ >> spaudžiame „*Advanced speech options*“ >> „*Speech recognition*“.

Pagrindiniame šnekamosios kalbos atpažinimo sistemos parametrų nustatymo lango skiltyje „*language*“ pasirenkamas ispanų kalbos atpažintuvas – „*Microsoft Speech Recognizer 8.0 for Windows (Spanish-Spain)*“. Susikuriamas ATP\_SP ir apmokomas. Norint apmokyti sukurtą naują atpažinimo sistemos profilį individualiam diktoriui pasirenkame parinktį „*Train profile*“. Apmokymo

metu skaitomas tekstas ir taip suteikdami sistemai reikiamą individualią akustinę ir vokalinę informaciją pasirinktam kalbos atpažinimo profiliui.

Sukuriamas „1.liftokomandos.grxml“ failas, kuriame surašomos visų komandų transkripcijos (žr. 2.2.2 skyrių), vėliau šitas failas bus panaudotas aktyvuojant programą *fil\_voc\_v1*, kuri sukurta buvusioje KTU kalbos signalų tyrimo mokslo laboratorijoje. Kitakalbis atpažintuvas kiekvieną ištartą diktorius komandą lygina su transkripcija ir išveda rezultatą. Žodžio *antras* transkripcijos pavyzdys pateiktas 2.5 pav.

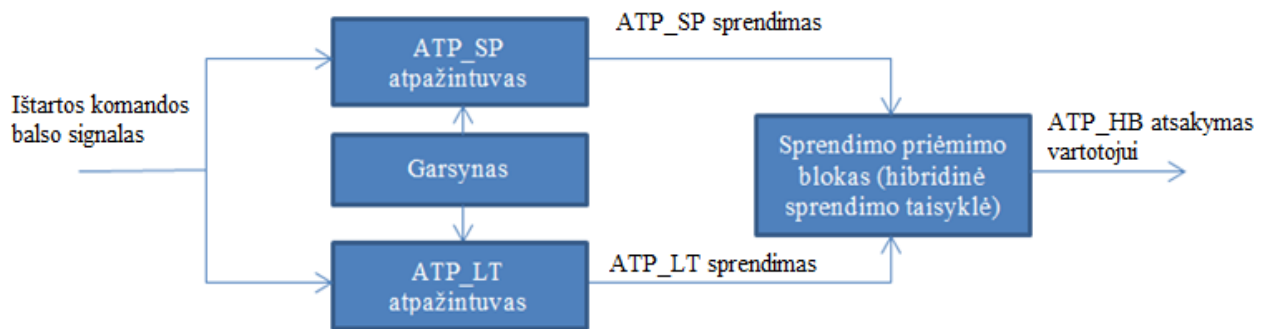
```
<item>
  <one-of>
    <item>hantras</item>
    <item>antras</item>
    <item>antrass</item>
    <item>aaantras</item>
  </one-of>
  <tag>$. _value = "Antras"</tag>
</item>
```

### 2.5 pav. Komandos „antras“ transkripcijos pavyzdys

Įkeliami visi 1980 ištartų komandų garso įrašai ir paleidžiama programa „fil\_voc\_v1“. Ji patikrina, kokių tikslumu kiekvienas įrašas yra atpažystamas. Baigus darbą, programa išveda rezultatus į „rezultatai.txt“ failą. Gauti rezultatai atvaizduoti 3 skyriuje.

### 2.6 Dviejų balso atpažintuvų sujungimo metodika

Atlikus tyrimą su HTK programiniu įrankių paketu ir tyrimą su kitakalbiu ispanišku atpažintuvu, gauti rezultatai (žr. 3 skyrių), kurie parodo, kad balso atpažinimo sistema neturi aukšto (daugiau 95%) atpažinimo tikslumo. Norint padidinti balso komandų atpažinimo tikslumą, reikalingas abiejų atpažintuvų apjungimas. Lygiagretus atpažintuvų naudojimas (2.6 pav) reikalingas, kai ATP\_LT atpažintuvas neatpažysta ištartos komandos, o ATP\_SP atpažysta tą komandą, arba atvirkščiai, tada pasitelkiant ATP\_HB, balso komandų atpažinimo tikslumas pagerės, nes bent vienas iš dviejų atpažintuvų atpažins ištartą komandą. Balso atpažinimo sistemos tikslumas nepakils tik tais atvejais, kai abudu ATP\_LT ir ATP\_SP atpažintuvai neatpažins ištartos komandos.



**2.6 pav. Hibridinio atpažintuvo veikimo schema**

Tyrimo metu gavus ATP\_SP ir ATP\_LT rezultatus, rezultatai suvedami į vieną „\*.excel“ dokumentą. Tame „\*.excel“ dokumente užpildomi tokie stulpeliai:

- Diktorius. Jame pildomi visi diktoriai abėcėlės tvarka;
- Failas. Jame surašomi visi įrašų pavadinimai pvz., 0A01, 0A02 ir t.t.;
- Teisingas sprendimas. Užrašomas koks žodis išstartas;
- LT\_sprendimas. Jame užrašomi visi gauti ATP\_LT sprendimai;
- LT\_atpažinimo\_tikimybė. Čia surašomi, kokia tikimybė buvo atpažintas išstartas žodis;
- SP\_sprendimas. Jame užrašomi visi gauti ATP\_SP sprendimai;
- SP\_atpažinimo\_tikimybė. Čia užrašomi, kokia tikimybė buvo atpažintas išstartas žodis;
- Lytis. Užrašoma diktoriaus lytis. F (angl. Female) arba M (angl. Male);
- Klasė. Užrašoma atpažintuvo klase. LT arba SP;
- True\_LT. Jei lt atpažino žodį įvedama 1, jei neatpažino tai įvedama 0;
- True\_SP. Jei sp atpažino žodį įvedama 1, jei neatpažino tai įvedama 0;
- Kiekviename atskirame stulpelyje surašomos komandose naudojamos abėcėlės raidės, atpažintos komandos raidės atitinkamuose stulpeliuose tada supildomos;

Kitas žingsnis pilnai užpildžius „\*.excel“ dokumentą, užpildyti papildomumo lentelę, pagal kurią apskaičiuosime „nemokyta“ arba kitaip vadinama „aklą“ atpažinimo taisyklę. Neužpildytos lentelės, kuri 3.4 skyriuje hibridinio atpažintuvo tyrime užpildoma, pateikta žemiau.

## 2.2 lentelė. ATP\_HB atpažinimo tikslumo papildomumo lentelė

Poaibis	Aprašymas
T=T	Atpažintuvų siūlomi sprendimai sutampa ir yra teisingi
F=F	Atpažintuvų siūlomi sprendimai sutampa ir yra neteisingi
T-	Atpažintuvas LT siūlo teisingą sprendimą, atpažintuvas SP sprendimo nesiūlo
F-	Atpažintuvas LT siūlo neteisingą sprendimą, atpažintuvas SP sprendimo nesiūlo
-T	Atpažintuvas SP siūlo teisingą sprendimą, atpažintuvas LT sprendimo nesiūlo
-F	Atpažintuvas SP siūlo neteisingą sprendimą, atpažintuvas LT sprendimo nesiūlo
--	Abu atpažintuvai nesiūlo sprendimo
TF	Atpažintuvų siūlomi sprendimai nesutampa, LT sprendimas teisingas
FT	Atpažintuvų siūlomi sprendimai nesutampa, SP sprendimas teisingas
FF	Atpažintuvų siūlomi sprendimai nesutampa, abu sprendimai neteisingi
Iš viso	

## 2.7 Duomenų požymių tyrimo metodika

Duomenų požymių tyrimui atlikti naudojama duomenų analizės sistema „WEKA“ (žr. 1.7 skyrių). Hibridinės sprendimo priėmimo taisyklės efektyvumas tikrinamas dviem būdais:

Įprastinis 10-kartų kryžminis patikrinimas su grafine WEKA sąsaja. WEKA programoje pakete pasirenkami norimi tirti duomenys, pasirenkame failą „visi.arff“ kuriame surašyti, visi dviejų atpažintuvų atpažinimo požymiai (8 PRIEDAS). Tyrimui pasirenkame „Cross-validation“ funkciją ir nustatome, kad būtų visi duomenys suskirstyti į aplnakalus iš kurių 10% bus skirti testavimui, o 90% apmokymui. Sistema „visi.arff“ failo duomenis suskirsto į 10 lygių dalių ir tada testuoja tikslumą kryžminiu būdu. Prieš pradėdant testavimą pasirenkamas norimas klasifikatorius. Klasifikatoriaus parametrai nekeičiami, tik radus efektyviausią klasifikatorių, kitame tyrimo etape bus keičiami efektyviausio klasifikatoriaus parametrai. Toks klasifikavimas leidžia prognozuoti klasifikavimo tikslumą (tuo pačiu ir hibridinio atpažintuvo tikslumą) žinomam kalbėtoji (vienam iš garsyno diktorių).

Sudėtingesnis 11 kartų kryžminis patikrinimas, kur 11-diktorių skaičius. Antru atveju – atidaromas WEKA duomenų analizės sistemos paketo įrankis „SimpleCLI“. Šio įrankio komandinėje eilutėje yra nurodomas duomenų klasifikatoriaus tipas, klasifikatorius, klasifikatoriaus apmokymo duomenų direktorija ir testavimo duomenų direktorija, pvz.:

---

```
java weka.classifiers.trees.RandomForest
-t D:/Weka/hybrid_lt_sp_train_test_sets/1_train.arff
-T D:/Weka/hybrid_lt_sp_train_test_sets/1_test.arff
```

Sistema apmokoma „train.arff“ failu, kuriame surašyti 10 diktorių atpažinimo tikslumo požymiai. Testuojama „test.arff“ failu, kuriame surašyti likusio 1 diktoriaus atpažinimo tikslumo požymiais. Toks apmokymo – testavimo ciklas kartojamas 11 kartų. Tokio klasifikavimo rezultatai leidžia

prognozuoti klasifikavimo tikslumą (tuo pačiu ir hibridinio atpažintuvo tikslumą) nežinomam kalbėtojiui.

Efektyviausio klasifikatoriaus nustatymo tyrimo metu pasirenkami 25 klasifikatoriai: BN, NB, Logistic, Mp, SGD, SL, SMO, VP, Ibk, Kstar, AdaBoost1, Bagging, MCC, Stacking, Vote, DT, RIPPER, OneR, PART, ZeroR, C4.5, LMT, RF, RT, REPTree. Visi išvardinti klasifikatoriai yra pasirinkti nes visi jie yra tikrinami naujos optimaliausių duomenų parametų radimo ir efektyviausio klasifikatoriaus radimo funkcijos, Auto-WEKA (žr 1.7.1 skyrius).

Suradus efektyviausia klasifikatorių koreguojami jo parametrai, kol bus pasiektas aukščiausias tiskslumas. Suradus optimaliausius parametrus, atliekamas tikslumo priklausomybės nuo požymių tyrimas, mažinant požymių klases. Failuose „visi.arff“ , „1-11train.arff“ ir „1-11test.arff“ panaikinamos požymių klasės, taip sukuriant naujus failus tais pačiais pavadinimais, kurie vėl bus testuojami su jau nustatytu efektyviausiu klasifikatoriumi ir nustatytais jo parametrais.



### 3. ATLIKTŲ TYRIMŲ REZULTATŲ DALIS

Tyrimo metu sistema buvo apmokoma dešimties diktorių išstartų komandų įrašais, testuojama vieno diktoriaus išstartais komandų įrašais, toks tyrimas priskiriamas aukščiausiam kalbos atpažinimo lygiui. Kadangi sistema buvo testuojama 11 atskirų diktorių, tai buvo sukurti atskiri vienuolika aplankų. Kiekvienu atveju kalbos atpažinimas buvo tikrinamas vis su skirtingu diktoriumi. Tyrimo eiga atlikta pasitelkiant 2.2 skyriuje aprašytus septynis etapus. Rezultatai gauti naudojant HTK programinį įrankių paketą, kai kalbos atpažinimo sistemos tikslumas randamas keičiant būsenų skaičių (3.1 skyriu) ir keičiant būsenas ir gauso mišinius (3.2 skyrius). Lipto valdymo komandų garsyno atpažinimo tikslumo tyrimas atliktas ir su kitakalbiu ispanišku atpažintuvu, aprašytu 2.5 skyriuje, gauti rezultatai pateikti 3.3 skyriuje. Gavus ATP\_LT ir ATP\_SP rezultatus, tolesnėje tyrimo eigoje, mašininio apjungimo metodu, sudaromas hibridinis atpažintuvas ATP\_HB (žr 2.6 skyrius). Randama „akla“ hibridinio atpažinimo taisyklė. Galiausiai atliekamas tyrimas su duomenų analizės sistema WEKA. Su kuria, tyrimo metu ieškomas efektyviausias klasifikatorius, su kuriuo gaunami geriausi atpažinimo tikslumo rezultatai. Radus efektyviausią klasifikatorių, atliekami tyrimai, nustatant lifto komandų atpažinimo tikslumo priklausomybę nuo laimėjusio klasifikatoriaus kintančių parametrų.

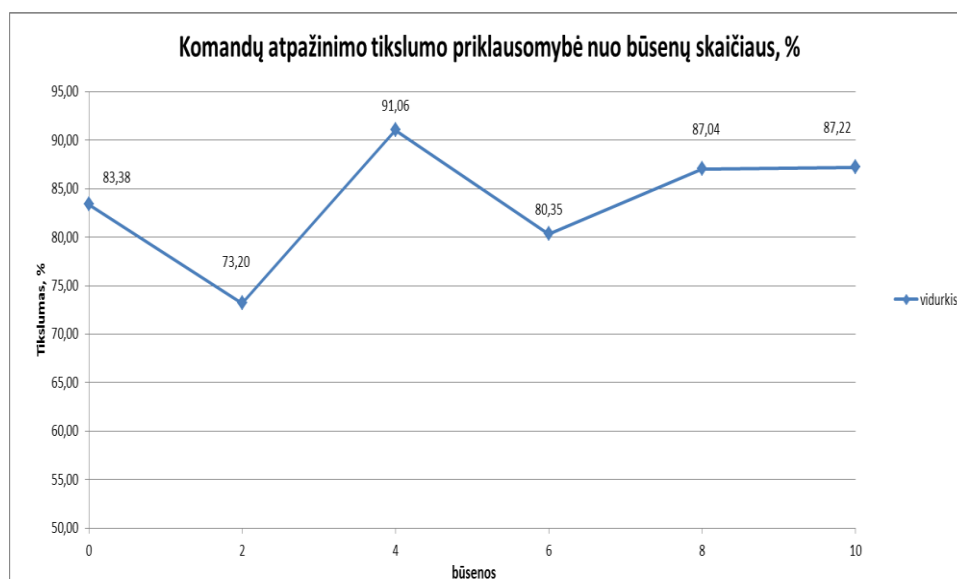
#### 3.1 Komandų atpažinimo tyrimas naudojant HTK metodą, pridedant tik papildomas būsenas

Pagal 2.2 skyriuje aprašytą metodiką, sumodeliuoti lifto valdymo garsyno komandų paslėptieji Markovo modeliai, kuriais naudojantis tikrinama sistema ir gaunami išstartų lifto komandų atpažinimo tikslumo rezultatai. Rezultatai gauti testuojant atpažinimo tikslumą su kiekvienu diktoriumi, to pasekoje, kai „hmm“ yra su nuliu papildomų būsenų, atpažinimo sistema patikrinama vienuolikos diktorių išstartų komandų įrašų, prie „hmm“ modelių pridėjus papildomas būsenas, atpažinimo sistema ir vėl patikrinama vienuolika kartų ir t.t. Visi rezultatai gauti didinant būsenų kiekį nuo 0 iki 10 pateikti 3.1 lentelėje. Lentelė, kurioje surašyti visų diktorių lifto komandų atpažinimo tikslumo rezultatai, keičiant vidinių būsenų skaičių, pateikta 5 PRIEDE.

3.1 lentelė. Komandų atpažinimo tikslumo priklausomybė nuo būsenų skaičiaus

Vidinių pridėtų būsenų skaičius	0	2	4	6	8	10
Atpažinimo tikslumas, %	83,38±5,7 0	73,20±5,9 1	91,06±5,6 9	80,35±6,4 0	87,04±4,2 1	87,22±5, 86

Atliekant tyrimą manyta, kad didinant vidinių būsenų skaičių, išstartų lifto komandų atpažinimo tikslumas gerės. Bet iš rezultatų matome, kad rezultatai kinta ne tik gerėjimo eiga, bet ir tam tikrais atvejais sumažėja. Kad lengviau išanalizuoti gautus atpažinimo tikslumo duomenis, kintant vidinių būsenų skaičiui, gauti rezultatai atvaizduoti ašinėje diagramoje (3.1 pav.).



**3.1 pav. Komandų atpažinimo tikslumo priklausomybė nuo būsenų skaičiaus**

Iš 3.1 paveikslo matome, kad praščiausias lifto komandų PMM atpažinimo tikslumas gautas, prie „hmm“ modelių, pridėjus dvi papildomas būsenas ( $73,20 \pm 5,91\%$ ), o aukščiausias atpažinimo PMM tikslumas gautas prie „hmm“ modelių pridėjus keturias papildomas būsenas ( $91,06 \pm 5,69\%$ ). Pridedant aštuonias ar dešimt būsenų, matomas rezultatų nusistovėjimas, gaunasi persimokymas (žr. 1.4.3 skyrius), šituo atveju rezultatai neparodo tikrojo balso atpažinimo tikslumo.

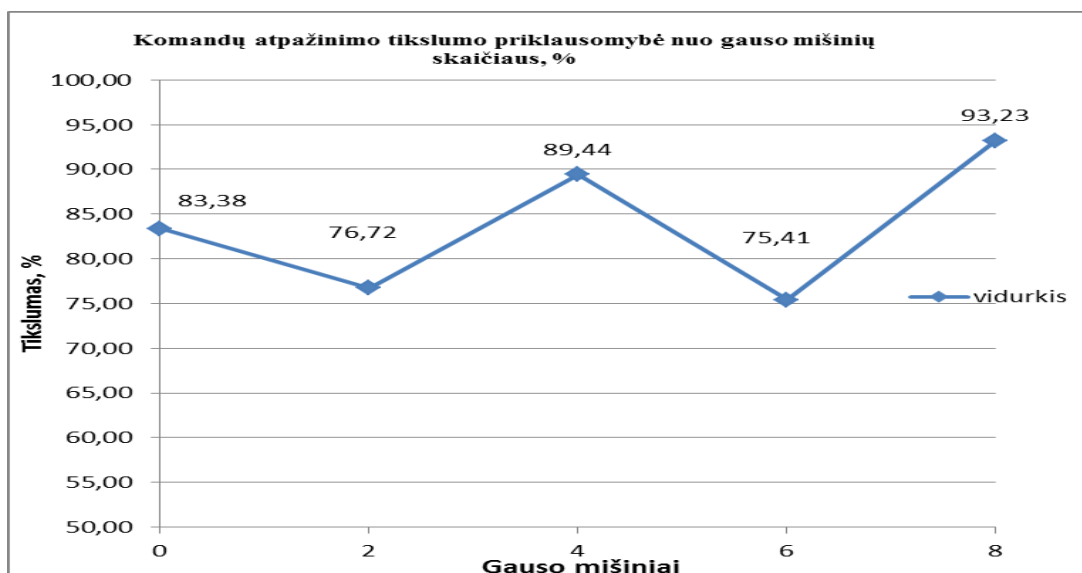
### **3.2 Komandų atpažinimo tyrimas naudojant HTK metodą, pridedant papildomas būsenas ir Gauso mišinius**

Pagal 2.2 skyriuje aprašytą, HTK programinių įrankių paketo taikymo balso komandų atpažinimui tirti metodiką, buvo sumodeliuoti lifto valdymo garsyno komandų paslėptieji Markovo modeliai, kurių atpažinimo tikslumo rezultatai, naudojant skirtingų modelių būsenų ir gauso mišinių skaičius, pateikiami šiame skyriuje. Rezultatai gauti tikrinant atpažinimo tikslumą su kiekvienu diktoriaus išstartos komandos įrašu, to pasekoje kai „hmm“ yra su nulių papildomų būsenų ir nulių papildomų gauso mišinių, atpažinimo sistema patikrinama vienuolika skirtingų atvejų. Pavyzdžiui., prie „hmm“ modelių pridėjus dvi papildomas būsenas ir du papildomus gauso mišinius, atpažinimo sistema pratestuojama vienuolika kartų ir t.t. Gauti rezultatai, kai prie „hmm“ nepridėtos papildomos būsenos, o didinamas gauso mišinių kiekis nuo dviejų iki aštuonių, pateikti 3.2 lentelėje. Lentelė kurioje surašyti visų diktorių lifto komandų atpažinimo tikslumo rezultatai naudojant PMM metodą, kai vidinių būsenų skaičius „0“ o, Gauso mišinių skaičius kinta nuo „0“ iki „8“ pateikta 6 PRIEDE.

**3.2 lentelė. Komandų atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus, nepridėjus būsenų**

Pridėta 0 papildomų vidinių būsenų					
Gauso mišinių skaičius	0	2	4	6	8
Atpažinimo tikslumas, %	83,38±5,70	76,71±7,57	89,44±6,38	75,41±7,84	93,23±4,24

Gauso mišinių įterpimas į lifto komandų PMM, 0, 4 ir 8 papildomų vidinių būsenų atvejais, pagerino atpažinimo tikslumo rezultatus. Iš 3.2 lentelėje pateiktų duomenų matome, kad geriausias lifto komandų modelių atpažinimo tikslumas gautas, į „hmm“ modelius įterpus aštuonis papildomus Gauso mišinius (93,23±4,24%), bet kaip ir keičiant tik būsenas, šitas rezultatas neadekvatus dėl persimokymo (žr. 2 skyrius). Todėl aukščiausiu gautu atpažinimo tikslumu laikome, kai į „hmm“ modelius įterpti keturi Gauso mišiniai (89,44±6,38%). Žemiausias rezultatas gaunamas, kai į „hmm“ modelius įterpti šeši Gauso mišiniai (75,41±7,84%). Tyrimo priklausomybė nuo Gauso mišinių skaičiaus pateikta ašinėje diagramoje pavaizduotoje 3.2 paveiksle.



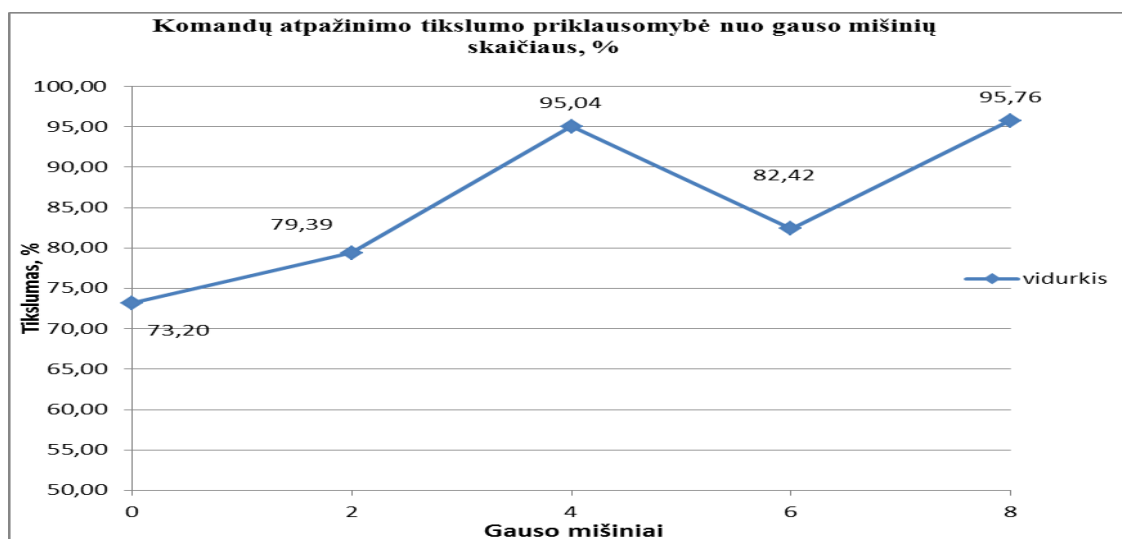
**3.2 pav. Komandų atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus, nepridėjus papildomų būsenų**

Toliau rezultatai gauti, kai prie „hmm“ pridėtos dvi papildomos būsenos, o Gauso mišinių skaičius didinamas nuo „0“ iki „8“, pateikti 3.3 lentelėje. Lentelė kurioje surašyti visų diktorių lifto komandų atpažinimo tikslumo rezultatai, kai prie „hmm“ modelių vidinių būsenų pridėtos dvi papildomos būsenos, o Gauso mišiniai kinta nuo „0“ iki „8“ pateikta 6 PRIEDE.

**3.3 lentelė. Komandų atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus, pridėjus 2 būsenas**

Pridėtos 2 papildomos vidinės būsenos					
Gauso mišinių skaičius	0	2	4	6	8
Atpažinimo tikslumas, %	73,20±5,91	79,39±8,39	95,04±4,05	82,42±8,26	95,76±4,97

Gauso mišinių įterpimas į lifto komandų modelius visais atvejais, išskyrus neįterpus jokių Gauso mišinių, pagerino atpažinimo tikslumo rezultatus. Iš 3.3 lentelėje pateiktų duomenų matome, kad geriausias lifto komandų modelių atpažinimo tikslumas PMM metodu gautas, į „hmm“ modelius įterpus aštuonis papildomus Gauso mišinius (95,76±4,97% ), bet kaip ir įterpus aštuonis Gauso mišinius bei ne pridėjus lifto komandų PMM jokių papildomų vidinių būsenų, gautas rezultatas laikomas neadekvačiu, dėl persimokymo (žr. 2 skyrius). Todėl aukščiausiu gautu atpažinimo tikslumu laikome, kai į „hmm“ modelius yra įterpti keturi Gauso mišiniai (95,04±4,05%). Žemiausias rezultatas gaunamas, kai į „hmm“ modelius neįterptas nei vienas Gauso mišinys (73,20±5,91%). Tyrimo priklausomybė nuo Gauso mišinių skaičiaus pateikta ašinėje diagramoje pavaizduotoje 3.3 paveiksle.



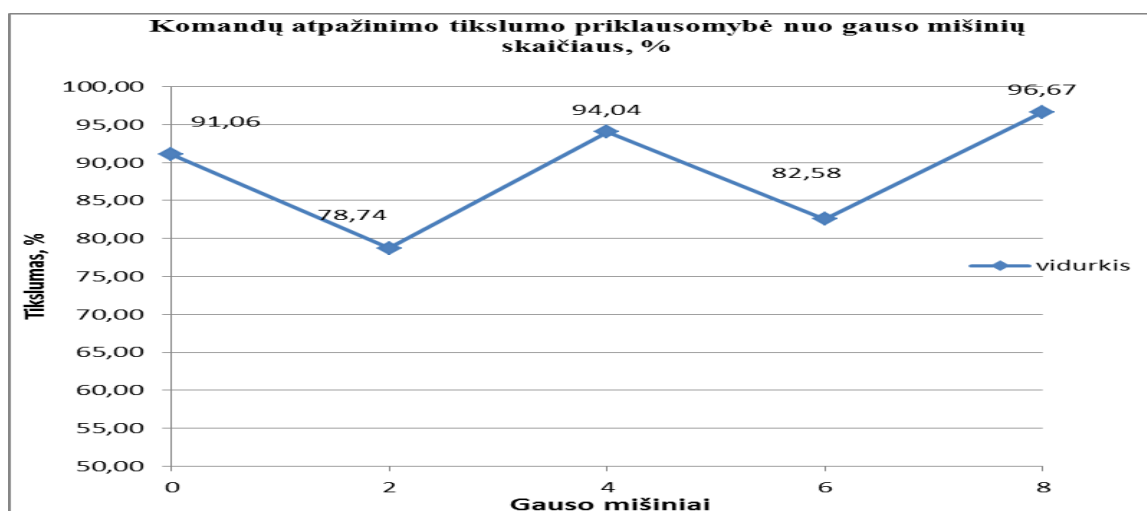
**3.3 pav. Komandų atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus, pridėjus dvi papildomas būsenas**

Toliau rezultatai gauti, kai prie „hmm“ pridėtos keturios papildomos būsenos, o Gauso mišinių skaičius didinamas nuo „0“ iki „8“, pateikti 3.4 lentelėje. Lentelė kurioje surašyti visų diktorių lifto komandų atpažinimo tikslumo rezultatai, kai prie „hmm“ modelių vidinių būsenų pridėtos keturios papildomos būsenos, o Gauso mišinių kiekis kinta nuo „0“ iki „8“ pateikta 6 PRIEDE.

**3.4 lentelė. Komandų atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus, pridėjus 4 būsenas**

Pridėtos 4 papildomos vidinės būsenos					
Gauso mišinių skaičius	0	2	4	6	8
Atpažinimo tikslumas, %	91,06±5,69	78,74±7,68	95,04±4,54	82,58±6,46	96,67±3,95

Pridėjus keturias papildomas vidines būsenas į lifto komandų modelius ir įterpus Gauso mišinius kintančius nuo nulio iki aštuonių, gauname rezultatus, kurie kinta kitokiu desningumu nei prieš tai gautuose atvejuose. Iš 3.4 lentelėje pateiktų duomenų matome, kad geriausias lifto komandų modelių atpažinimo tikslumas gautas, į „hmm“ modelius įterpus aštuonis papildomus Gauso mišinius (96,67±3,95%), bet kaip ir prieš tai gautais atvejais galime laikyti, kad rezultatas neadekvatus, dėl persimokymo (žr. 2 skyrius). Todėl aukščiausiu gautu atpažinimo tikslumu laikome, kai į „hmm“ modelius yra įterpti keturi Gauso mišiniai (95,04±4,54%), bet šituo atveju atpažinimo tikslumo rezultatas gavosi mažesnis nei prie lifto komandų PMM pridėjus dvi papildomas būsenas ir į „hmm“ modelius įterpus keturis papildomus Gauso mišinius. Žemiausias rezultatas gaunamas, kai į „hmm“ modelius įterpti du Gauso mišiniai (78,74±7,68%). Tyrimo priklausomybė nuo Gauso mišinių skaičiaus pateikta ašinėje diagramoje pavaizduotoje 3.4 paveiksle.



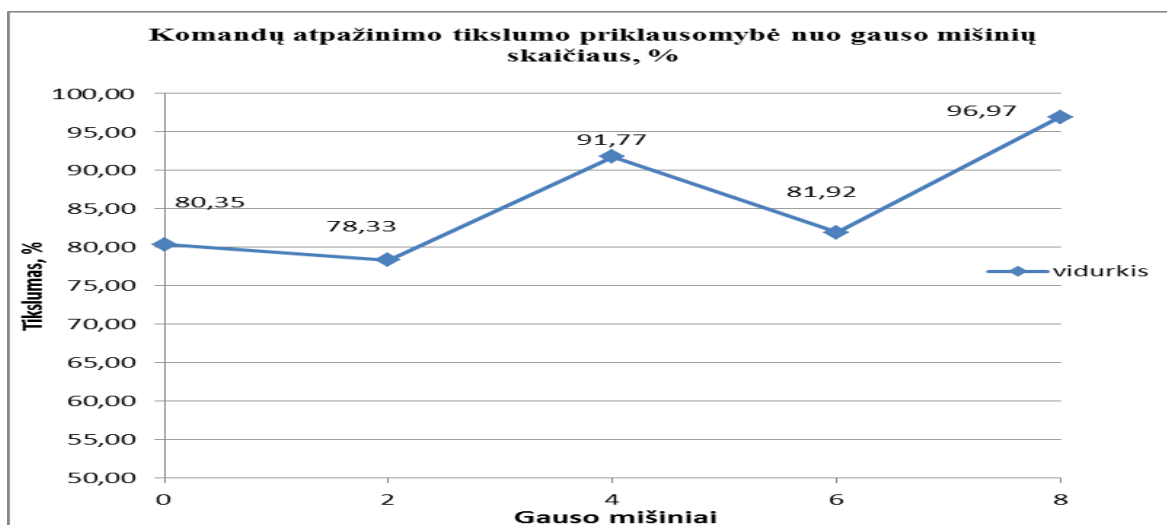
**3.4 pav. Komandų atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus, pridėjus keturias papildomas būsenas**

Toliau rezultatai gauti, kai prie „hmm“ pridėtos šešios papildomos būsenos, o Gauso mišinių skaičius didinamas nuo „0“ iki „8“, pateikti 3.5 lentelėje. Lentelė kurioje surašyti visų diktorių lifto komandų atpažinimo tikslumo rezultatai, kai prie „hmm“ modelių vidinių būsenų pridėtos šešios papildomos būsenos, o Gauso mišinių skaičius kinta nuo „0“ iki „8“ pateikta 6 PRIEDE.

**3.5 lentelė. Komandų atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus, pridėjus 6 būsenas**

Pridėtos 6 papildomos vidinės būsenos					
Gauso mišinių skaičius	0	2	4	6	8
Atpažinimo tikslumas, %	80,35±6,40	78,33±7,62	91,77±4,85	81,92±6,30	96,96±2,74

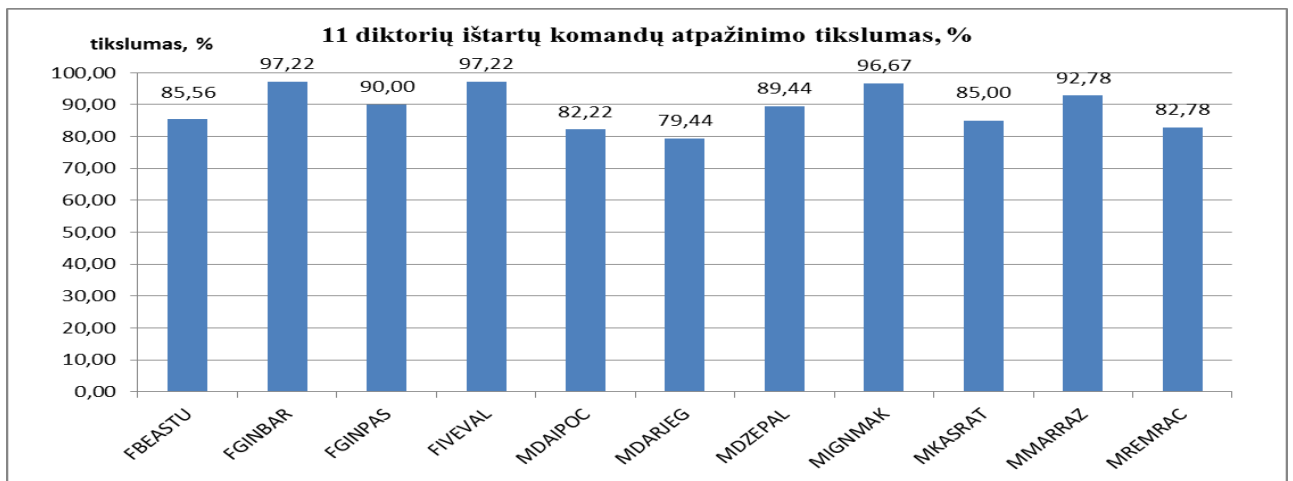
Pridėjus šešias papildomas vidines būsenas į lifto komandų PMM ir įterpus Gauso mišinius kintančius nuo nulio iki aštuonių, gauname rezultatus, kurie kinta panašiu dėsningumu, kaip ir 3.4 lentelėje pateikti rezultatų duomenys. Iš 3.5 lentelėje pateiktų duomenų matome, kad geriausias lifto komandų modelių atpažinimo tikslumas gautas, į „hmm“ modelius įterpus aštuonis papildomus Gauso mišinius (96,96±2,74% ), bet kaip ir prieš tai gautais atvejais galime laikyti, kad rezultatas neadekvatus, dėl persimokymo (žr. 2 skyrius). Todėl aukščiausiu gautu atpažinimo tikslumu laikome, kai į „hmm“ modelius įterpti keturi Gauso mišiniai (91,77±4,85%), bet šituo atveju atpažinimo tikslumo rezultatas gavosi dar mažesnis nei prie lifto komandų PMM pridėjus keturias papildomas būsenas ir į „hmm“ modelius įterpus keturias papildomas Gauso mišinius. Žemiausias rezultatas gaunamas, kai į „hmm“ modelius neįterptas nei vienas Gauso mišinys (80,35±6,40%). Tyrimo priklausomybė nuo Gauso mišinių skaičiaus pateikta ašinėje diagramoje pavaizduotoje 3.4 paveiksle.



**3.5 pav. Komandų atpažinimo tikslumo priklausomybė nuo Gauso mišinių skaičiaus, pridėjus šešias papildomas būsenas**

### 3.3 Tyrimas naudojant kitakalbį atpažintuvą

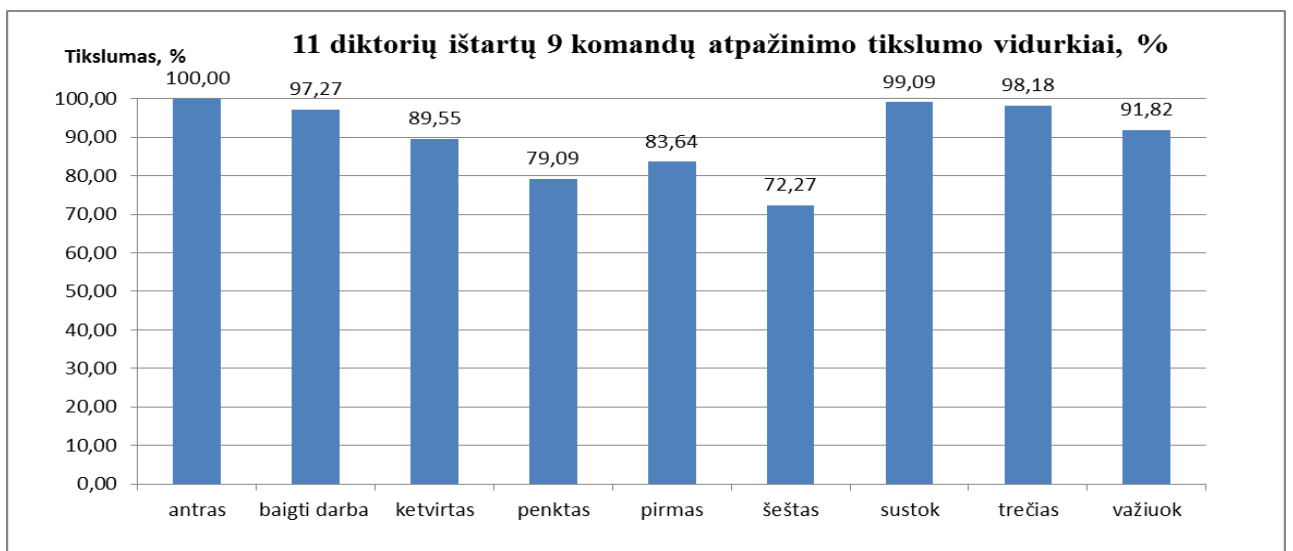
Pagal aprašytą 2.5 skyriuje metodiką, atliekamas tyrimas su kitakalbiu ispanišku atpažintuvu. Rezultatai lifto modelio valdymo komandų garsyno atpažinimo tikslumo, naudojant kitakalbį atpažintuvą pateikti 3.6 paveiksle.



**3.6 pav. Kitakalbio atpažintuvo 11 diktorių išstartų komandų atpažinimo tikslumas**

Iš gautų duomenų matome, kad visų diktorių komandų atpažinimo rezultatai skirtingi, bendras atpažinimo tikslumo rezultatų vidurkis, visų 11 diktorių, yra  $88,94 \pm 3,78\%$ . Atsižvelgus į rezultatus gautus HTK metodu, galime teigti, kad ispaniško atpažintuvo gautas atpažinimo rezultato vidurkis yra didesnis nei HTK metodu gautas vidurkis, kai HTK metodu nepridėta prie PMM jokių papildomų vidinių būsenų ir nėra įterpta Gauso mišinių. Bet didinant būsenų ir gauso mišinių kiekius buvo gauti geresni rezultatai nei su kitakalbiu ispanišku atpažintuvu.

Ispaniško kitakalbio atpažintuvo lifto modelio valdymo komandų rezultatai parodyti 3.7 paveiksle.



**3.7 pav. Kitakalbio atpažintuvo 11 diktorių išstartų 9 komandų atpažinimo tikslumai**

Iš 3.7 paveikslo matome, kad visų komandų atpažinimo tikslumai yra skirtingi. Geriausiai atpažintas žodis yra „antras“ (100%), o blogiausiai „šeštas“ (72,27%). Lyginant rezultatus su kitais moksliniais darbais ir juose gautus tyrimų rezultatus [15], [16] pastebėta, kad visais atvejais lietuviški

žodžiai, kurie turi raides š arba č yra dažnai blogai atpažystami, todėl atpažinimo tikslumas mažėja. Kad padidinti komandos „šeštas“ tikslumą reikėtų didinti transkripcijų skaičių, ko šitame darbe dėl laiko stokos nebuvo padaryta. Kad gauti labai aukštą tikslumą (daugiau 95%) reikėtų lietuviško atpažintuvo, kuris dėja dar nėra sukurtas.

### 3.4 Hibridinio atpažintuvo galimybių tyrimas

Apjungus ATP\_LT ir ATP\_SP, gaunamas ATP\_HB. Gavus ATP\_LT komandų atpažinimo rezultatus (3.1 ir 3.2 skyriai) ir ATP\_SP komandų atpažinimo rezultatus (3.3 skyrius), sudarome bendrą atpažintuvų papildomumo lentelę. Joje surašomi, pagal 2.6 skyriuje pateiktą šabloną, rezultatai pagal kuriuos skaičiuojamas ATP\_HB komandų atpažinimo tikslumas. Atpažintuvų ATP\_SP ir ATP\_LT rezultatų papildomumo užpildyta lentelė pateikta žemiau.

### 3.6 lentelė. Atpažintuvų ATP\_SP ir ATP\_LT rezultatų papildomumas.

Poaibis	10-kartų kryžminis patikrinimas	11-kartų kryžminis patikrinimas, esant nežinomam diktoriui										
		1	2	3	4	5	6	7	8	9	10	11
T=T	1685	109	172	160	174	136	141	152	172	140	167	162
F=F	1	1	0	0	0	0	0	0	0	0	0	0
T-	6	0	0	0	0	0	0	0	0	0	0	6
F-	0	0	0	0	0	0	0	0	0	0	0	0
-T	0	0	0	0	0	0	0	0	0	0	0	0
-F	0	0	0	0	0	0	0	0	0	0	0	0
--	0	0	0	0	0	0	0	0	0	0	0	0
TF	191	22	5	20	5	34	37	15	6	22	13	12
FT	81	41	3	0	1	10	2	9	2	13	0	0
FF	16	7	0	0	0	0	0	4	0	5	0	0
Iš viso	1980	180	180	180	180	180	180	180	180	180	180	180

Iš 3.6 papildomumo lentelės aiškiai matome, kad Įprastiniame 10-kartų kryžminiame patikrinime lietuviško atpažintuvo vidutinis tikslumas yra 94,4% (T=T, T-, TF poaibiai), o ispaniško - 89,27% (T=T, FT poaibiai). Nors ispaniško atpažintuvo atpažinimo tikslumas yra mažesnis, tačiau šis atpažinimo metodas tam tikrus balso įrašus atpažįsta geriau nei lietuviškas atpažintuvas (FT=81). Todėl galima daryti prielaidą, kad apjungus abu atpažintuvus galima pasiekti didesnę atpažinimo tikslumą negu naudojant tik lietuvišką atpažintuvą.



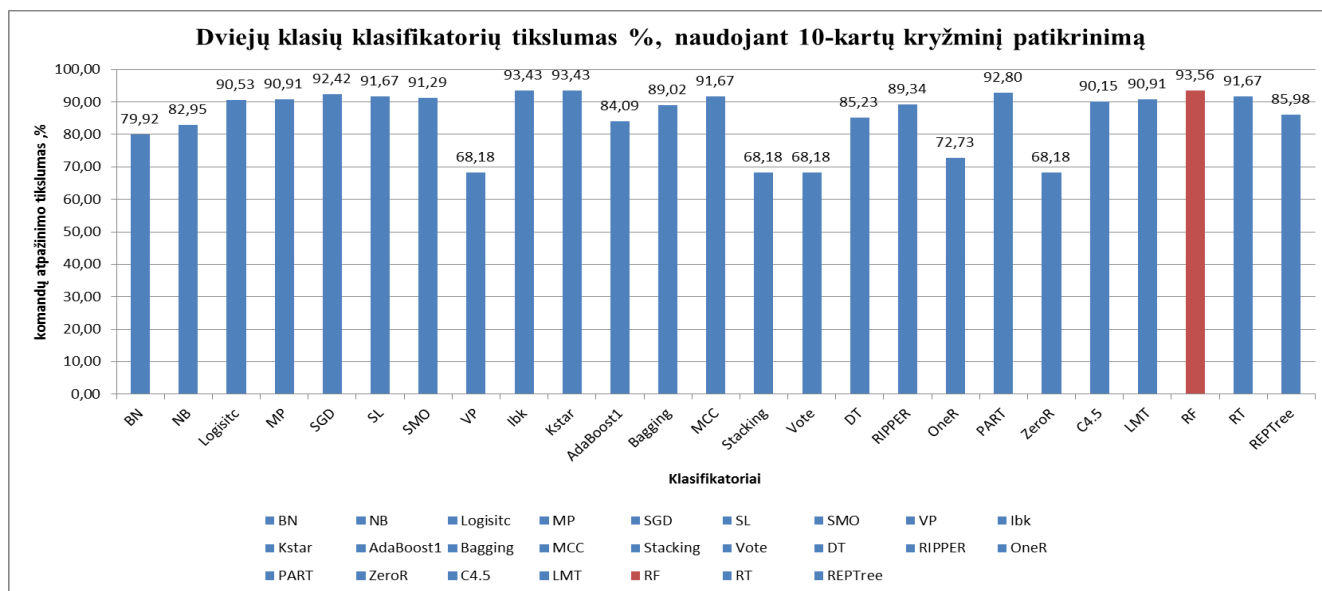
### 3.4.1 Hibridinio atpažintuvo galimybių tyrimas naudojant WEKA

Su WEKA duomenų analizės sistema, tyrimo metu nustatomas efektyviausias klasifikatorius. Nustačius efektyviausią klasifikatorių, parenkami optimaliausi parametrai. Efektyviausio klasifikatoriaus nustatymo tyrimo metu buvo pasirinkti 25 klasifikatoriai (žr. 1.7.1 skyrius): BN, NB, Logistic, Mp, SGD, SL, SMO, VP, Ibk, Kstar, AdaBoost1, Bagging, MCC, Stacking, Vote, DT, RIPPER, OneR, PART, ZeroR, C4.5, LMT, RF, RT, REPTree. WEKA sitemoje, hibridinė sprendimo priėmimo taisyklė buvo mokoma ir tikrinama pagal 2.7 skyriuje aprašytą metodiką. Gauti atpažinimo rezultatai su išvardintais klasifikatoriais pateikti 3.7 lentelėje. Pilna lentelė, su klasifikatoriu gautais rezultatais, pateikta 7 PRIEDE.

**3.7 lentelė. Visų failų ir atskirų 11 aplankų vidurkio atpažinimo tikslumai taikant skirtingus klasifikatorius.**

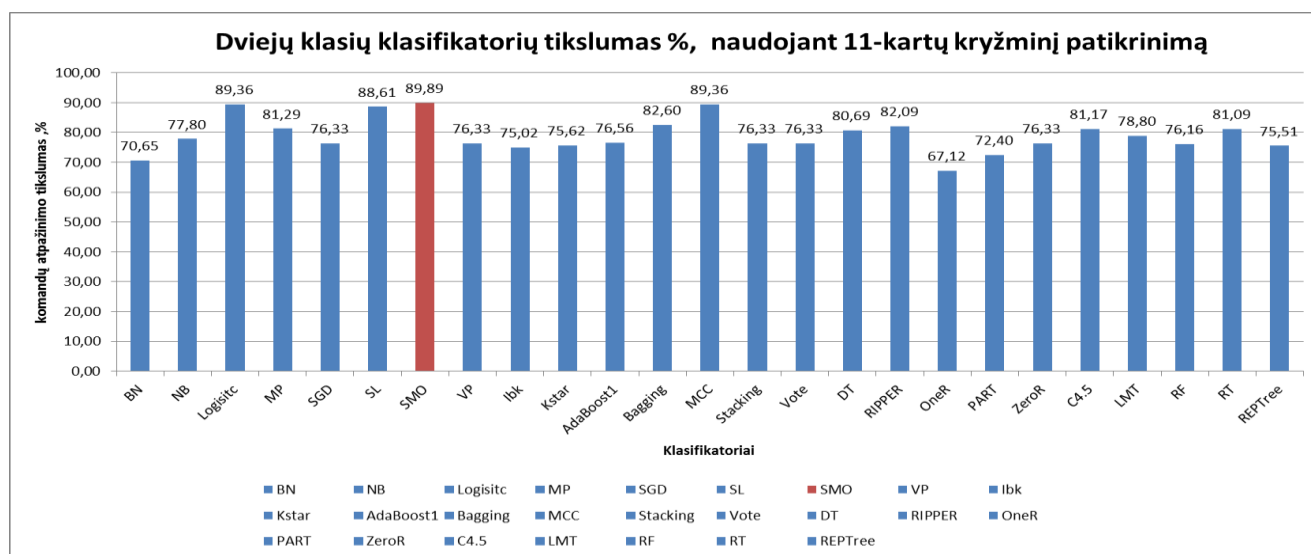
Klasifikatorius	Visų įrašų, padalintų į 10 lygių dalių, atpažinimo tikslumas, %	11-kartų kryžmintų atskirų aplankų vidurkių atpažinimo tikslumas, %
BN	79,92	70,65±15,20
NB	82,95	77,80±13,54
Logisitic	90,53	89,36±5,82
MP	90,91	81,29±9,65
SGD	92,42	76,33±13,02
SL	91,67	88,61±8,90
SMO	91,29	89,89±7,32
VP	68,18	76,33±13,02
Ibk	93,43	75,02±11,81
Kstar	93,43	75,62±11,20
AdaBoost1	84,09	76,56±9,21
Bagging	89,02	82,60±12,29
MCC	91,67	89,36±5,82
Stacking	68,18	76,33±13,02
Vote	68,18	76,33±13,02
DT	85,23	80,69±12,51
RIPPER	89,34	82,09±12,25
OneR	72,73	67,12±14,20
PART	92,8	72,40±11,37
ZeroR	68,18	76,33±13,02
C4.5	90,15	81,17±7,84
LMT	90,91	78,80±10,54
RF	93,56	76,16±11,48
RT	91,67	81,09±12,35
REPTree	85,98	75,51±14,48

Tirtų klasifikatorių parametrų reikšmės tyrimo metu nekeičiamos, paliekant klasifikatorių parametrus ant pradinių reikšmių. Iš 3.7 lentelės rezultatų pastebime, kad visų įrašų, padalintų į 10 lygių dalių, atpažinimo tikslumas su visais klasifikatoriais skyrėsi, nuo tų pačių klasifikatorių atpažinimo tikslumo, tiriant vienuolika atskirų aplankų kryžminio patikrinimo būdu. Kad lengviau būtų išskirti, kurie klasifikatoriai efektyviausi, atpažinimo tikslumo priklausomybė nuo klasifikatoriaus tiriant visus įrašus suskirščius į 10 lygių dalių, kryžminio patikrinimo būdu, pavaizduota 3.8 paveiksle, o tiriant atskirų vienuolikos aplankų kryžminio patikrinimo būdu 3.9 paveiksle.



3.8 pav. Efektyviausio klasifikatoriaus radimas tiriant visų komandų įrašų atpažinimo tikslumą

Iš 3.8 paveikslo matome, kad geriausias atpažintuvas su pradiniais parametrais yra „Random forest“ (93,56%). Literatūros analizės metu iširta, kad daugumoje atvejų efektyviausias klasifikatorius yra „Random forest“.



3.9 pav. Efektyviausio klasifikatoriaus radimas tiriant vienolikos kartų kryžmintų 11 aplankų komandų įrašų atpažinimo tikslumą

Iš 3.9 paveikslo matome, kad geriausias klasifikatorius su pradiniais parametrais yra „SMO“ (89,89%). Literatūros analizės metu ištirta, kad tik nuo naujesnių WEKA-paketo duomenų analizės sistemos versijų sukurtas SMO klasifikatorius. Efektyviausio klasifikatoriaus gautas atpažinimo tikslumo rezultatas tiriant atskirų 11 failų kryžminio patikrinimo būdą (89,89%), mažesnis nei efektyviausio klasifikatoriaus atpažintuvo rezultatas tiriant visų garsyno įrašų, suskirstytų į 10 dalių kryžminio patikrinimo būdu (93,56%).

Kadangi gauti rezultatai yra nekeičiant klasifikatorių parametrų, tai kitas žingsnis yra keisti efektyviausių klasifikatorių: SMO ir RandomFores parametrus ir fiksuoti gaunamus rezultatus. Keičiant duomenų analizės klasifikatoriaus „RandomForest“ parametrus, gauti rezultatai surašyti 3.8 lentelėje, o keičiant „SMO“ parametrus, gauti rezultatai surašyti 3.9 lentelėje.

**3.8 lentelė. Klasifikavimo tikslumo priklausomybė nuo „RF“ klasifikatoriaus kintančių parametrų**

P-bagSizePercent	Batch size	max Depth	numDecimalPlaces	numExecutionsSlots	k-numFeatures	I-numIterations	S-seed	10-kartų kryžminio patikrinimo atpažinimo tikslumas, %	11-kartų kryžminio patikrinimo atpažinimo tikslumas, %
100	100	0	2	1	0	100	1	93,56	76,19±11,49
100	0	0	0	0	0	1	1	90,15	83,69±8,36
100	0	0	0	0	0	3	1	92,01	80,20±12,21
100	0	0	0	0	0	10	1	92,42	84,03±7,27
100	0	0	0	0	0	10	2	97,01	85,99±8,17
100	0	0	0	0	0	11	2	95,08	83,37±8,9
100	100	0	0	0	0	11	2	95,08	83,37±8,9
100	100	100	0	0	0	11	2	95,08	83,37±8,9
100	100	100	100	0	0	11	2	95,08	83,37±8,9
100	100	100	100	100	0	11	2	95,08	83,37±8,9
100	100	100	100	100	1	11	2	94,68	82,07±9,88
100	0	0	100	100	0	10	1	92,42	85,82±7,49

Kaip matome iš lentelės pateiktų duomenų, geriausias klasifikavimo tikslumas (97,01%) gautas nustatant parametrų reikšmes: p=100, I=10 ir S=2, o visus kitus parametrus nustatant ant 0. Palyginus šį gautą rezultatą (97,01%) su geriausiu atpažinimo tikslumo rezultatu, gautu HTK modeliu apmokymo būdu keičiant vidinių būsenų skaičių (91,06%) ar gauso mišinių skaičių (95,04%) ir kitakalbio atpažintuvo komandų atpažinimo tikslumo rezultatu (88,94%), galime teigti, kad sistema atpažysta balso komandas geresniu tikslumu. SMO klasifikatoriaus tikslumas tyrime siekia 93,18%, jis yra mažesnis negu naudojant klasifikatorių „RF“, kuriuo visų įrašų, padalintų į 11 lygių dalių, tikslumas siekia 97,01%. Bet SMO 11 aplankų kryžminimo atveju, tikslumas siekia 92,93%, o RF 11 aplankų kryžminimo atveju tesiekia 85,99%.

**3.9 lentelė. Klasifikavimo tikslumo priklausomybė nuo „SMO“ klasifikatoriaus kintančių parametrų**

BatchSize	c	epsilon	numDecimalPlaces	numFolds	randomSeed	toleranceParameter	kernel	10-kartų kryžminio patikrinimo atpažinimo tikslumas, %	11-kartų kryžminio patikrinimo atpažinimo tikslumas, %
100	1	1.0E-12	2	-1	1	0,001	PolyKernel	91,29	89,89±7,32
100	1	1.0E-12	2	-1	1	0,001	Puk	93,18	82,57±11,58
100	1	1.0E-12	2	-1	1	0,001	RBFKernel	79,18	80,54±11,59
100	5	1.0E-12	2	-1	1	0,001	PolyKernel	92,05	90,93±5,96
100	1	1.0E-6	2	-1	1	0,001	PolyKernel	90,15	92,93±5,14
100	5	1.0E-12	2	-1	1	0,001	Puk	93,18	78,35±10,37
100	1	1.0E-6	2	-1	1	0,001	Puk	93,18	83,98±9,78
100	5	1.0E-12	2	-1	1	0,001	RBFKernel	87,89	75,08±16,35
100	1	1.0E-6	2	-1	1	0,001	RBFKernel	78,79	78,71±12,35

Naudojant Wekos 3.8.1 versiją atliekamas tyrimas su nauja funkcija Auto-WEKA, kuri darbo metu automatiškai tikrina failo „visi.arff“ visų paminėtų 25 klasifikatorių tikslumus. Auto-WEKA veikimas aprašytas 1.7.1 skyriuje. Tyrimo metu Auto-WEKA nustatė, kad efektyviauses klasifikatorius yra „Random Forest“ su parametrais nustatytais : I= 10, K= 0 ir depth=0 (9 PRIEDAS). Gautas klasifikatoriaus tikslumo rezultatas 98,48%. Bandant pakartoti gautą rezultatą, rankiniu būdu, pakeičiant klasifikatoriaus „RandomForest“ parametrus, į tuos kuriuos gavo Auto-WEKA, rezultatas nesiekė daugiau 97% tikslumo. Kyla prielaida, kad kaip paminėta 1.7.1 skyriuje, optimizuojant parametrus gaunami nuokrypiai ir neina realiai priartėti prie rezultato, kurį gavo gan vis dar nauja optimizuotų parametrų radimo funkcija, Auto-WEKA.

Turint efektyviausio klasifikatoriaus rezultatus ir klasifikatoriaus su optimaliais parametrais rezultatus, galime apskaičiuoti, koku tikslumu veiks hibridinis atpažintuvas. Tokio atpažintuvo atpažinimo tikslumo rezultatai apskaičiuoti pagal žemiau pateiktą formulę (6), rezultatai pateikiami 3.10 ir 3.11 lentelėse.

$$\frac{\left( (X_{TF} + X_{FT}) * T_{klasif} \right) + X_{T=T} + X_{T-}}{X_{visų}} * 100 . \quad (6)$$

$T_{klasif}$  – klasifikatoriaus tikimybė teisingai parinkti atpažintuvą;

$X_{visų}$  – iš viso atpažinti naudojamų frazių kiekis

Šioje formulėje suvedami papildomumo 3.6 lentelės duomenys ir gauti atpažintuvo atpažinimo tikslumo rezultatai.

**3.10 lentelė. Lietuviško ir Hibridinio atpažintuvų atpažinimo tikslumų pokyčiai, esant žinomam kalbėtojui**

	Atpažintuvo atpažinimo tikslumas, %	Hibridinio atpažintuvo atpažinimo tikslumas, %	Atpažinimo klaidingumo sumažėjimas, %
ATP_LT	95,04	98,46	68,95
ATP_SP	88,94	97,62	78,48
ATP_HB „RandomForest“ su pradiniais parametrais	93,56	98,26	72,98
ATP_HB „RandomForest“ su nustatytais parametrais	97,01	98,73	57,53
ATP_HB Auto-WEKA „RandomForest“	98,48	98,98	32,89

Iš lentelės 3.10 duomenų matome, kad visais atvejais hibridinio atpažintuvo atpažinimo tikslumo rezultatai gavosi didesni už neapjungtų atpažintuvų atpažinimo tikslumo rezultatus. Didžiausias klaidingumo sumažėjimas gautas ATP\_SP (72,98%). Geriausias atpažinimo tikslumas gaunamas su Auto-WEKA (98,98%), teigiant, kad gali būti gautos optimizavimo paklaidos, gautas geriausias tikslumo rezultatas priimamas, kad yra pasitelkiant ATP\_HB su „RandomForest“ klasifikatoriumi, kurio parametrai optimaliai nustatyti (98,73%) .

**3.11 lentelė. Lietuviško ir Hibridinio atpažintuvų atpažinimo tikslumų pokyčiai, esant nežinomam kalbėtojui**

	Lietuviško atpažintuvo atpažinimo tikslumas, %	Hibridinio atpažintuvo atpažinimo tikslumas, %	Atpažinimo klaidingumo sumažėjimas, %
ATP_LT	95,04	98,46	68,95
ATP_SP	88,94	97,62	78,48
ATP_HB „SMO“ su pradiniais parametrais	89,9	97,75	77,72
ATP_HB „SMO“ su nustatytais parametrais	92,93	98,17	74,12

Iš lentelės 3.11 duomenų matome, hibridinio atpažintuvų atpažinimo tikslumų rezultatai, esant nežinomam kalbėtojui, visais atvejais gavosi didesni už neapjungtų atpažintuvų atpažinimo tikslumo rezultatus. Didžiausias klaidingumo sumažėjimas gautas taip pat su ATP\_SP (78,48%). Geriausias atpažinimo tikslumas gautas ATP\_LT atpažintuvą (98,46 %).

## IŠVADOS IR REZULTATAI

1. Atlikus lifto komandų garsyno atpažinimo tikslumo tyrimą, taikant HTK paketą ir žodžiais grįstus PMM, aukščiausias lifto komandų garsyno atpažinimo tikslumas buvo gautas naudojant PMM su 4 papildomomis būsenomis (91.06%);
2. Atlikus lifto komandų garsyno atpažinimo tikslumo tyrimą, taikant HTK paketą ir žodžiais grįstus PMM, aukščiausias komandų garsyno atpažinimo tikslumas buvo gautas naudojant PMM su 6 papildomomis būsenomis ir 8 Gauso mišiniais ( $96,96 \pm 2,74$ ). Priskyrus šį rezultatą prie duomenų, kurie neatitinka tikslių reikšmių dėl persimokymo, gaunasi, kad geriausias rezultatas yra gautas pridėjus 2 būsenas ir 4 Gauso mišinius ( $95,04 \pm 4,54$ );
3. Tyrimui su kitakalbiu atpažintuvu buvo pasitelktas ispaniškas kitakalbis atpažintuvas, gautas komandų atpažinimo rezultatų vidurkis  $88,94 \pm 3,78\%$ . Kitakalbio atpažintuvo atpažinimo rezultatas gavosi didesnis nei HTK metodu ( $83,38 \pm 5,70\%$ ), kai sistemos atpažinimo rezultatas gautas ne pridėjus PMM papildomų būsenų ar Gauso mišinių;
4. Naudojant WEKA duomenų analizės sistemos gautus duomenis nustatyta, kad kai sistema tirama 10-kartų kryžminiu patirkinimu, esant žinomam kalbėtojui (diktoriui) tiksliausias klasifikatorius, taikytinas dviejų atpažintuvų sujungimui, yra „RandomForest“. O kai sistema tirama 11-kartų kryžminiu patikrinimu, esant nežinomam kalbėtojui (diktoriui) tiksliausias klasifikatorius, taikytinas dviejų atpažintuvų sujungimui, yra „SMO“.
5. Tiriant WEKA duomenų analizės sistemos parametrų optimizavimo funkcijos Auto-WEKA gautus duomenis nustatyta, kad tiksliausias klasifikatorius, dviejų atpažintuvų sujungimui, yra taip pat „atsitiktinis miškas“ (random forest). Rankiniu būdu nustatant efektyviausio klasifikatoriaus parametrus, nepavyko pasiekti Auto-WEKA gauto tikslumo. Galimas optimizavimo paklaidos [24] atsiradimas, dėl kurios ir neišėjo gauti tokių pačių rezultatų.
6. Įvertinus optimizavimo paklaidas ir priėmus, kad geriausias atpažinimo rezultatas, sistemos atpažinimo tikslumo nustatymo 10-kartų kryžminio patikrinimo metodu esant žinomam kalbėtojui, gautas 98,73%, galime teigti, kad naudojant hibridinį atpažintuvą sistemos atpažinimo klaidingumas sumažėjo 57,53%.
7. Įvertinus optimizavimo paklaidas ir priėmus, kad geriausias atpažinimo rezultatas, sistemos atpažinimo tikslumo nustatymo 11-kartų kryžminio patikrinimo metodu esant nežinomam kalbėtojui, gautas 98,46%, galime teigti, kad naudojant hibridinį atpažintuvą sistemos atpažinimo klaidingumas sumažėjo 68,95%.

## LITERATŪROS ŠALTINIAI

1. **Bondarenka, E., Ratkevičius,** SAPI sąsajos analizė ir taikymai. Automatika ir valdymo technologijos - 2005, Kaunas, Technologija, 2005, p. 67-71.
2. **Kasparaitis P.** Kompiuterinė lingvistika. Kalbos signalų apdorojimas, 2013, p. 12. [žiūrėta 2015-04-09]. Prieiga per internetą: <http://www.mif.vu.lt/~pijus/CLF/SignApd.pdf>.
3. **Rudžionis, V., Ratkevičius, K., Rudžionis, A., Raškinis, G., Maskeliūnas, R.** "Recognition of voice commands using hybrid approach", In: Proc. Of 19th International Conference, ICIST 2013, Kaunas, Lithuania, Lecture Notes Communications in Computer and Information Science, vol. 403, p. 186-197, Springer.
4. **C. Myers, L. R. Rabiner, A. E. Rosenberg.** „Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition“. IEEE MII-DS-07T-14-13 61 transactions on acoustics, speech, and signal processing, Vol. ASSP-28, No. 6, 1980, p. 623 – 635.
5. **G. Hinton, L. Deng, D. Yu, G. E. Dahl.** „Deep Neural Network for Acoustic Modeling in Speech Recognition“. IEEE Signal Processing vol 82, 2012 ,p. 16.
6. **Young S., Evermann G., Kershaw D., Moore G., Odell J., Ollason D., Valtchev V., Woodland P.** The HTK Book (version 3.4). Microsoft Corporation, 2012, p 386.
7. **Jelinek F.** Statistical Methods for Speech Recognition ISBN 0-262-10066-5. USA, 1997, p. 279 . Interaktyvus [žiūrėta 2016-05-10] Prieiga per internetą: <http://books.google.lt/books?id=1C9dzcJTWoWC&printsec=frontcover&hl=en#PPP1,M1>.
8. **K. B. Khanchandani, M. A. Hussain.** „Emotion recognition using multilayer perceptron and generalized feed forward neural network“. Journal of Scientific & Industrial Research, Vol. 68, 2009, p. 367 – 371.
9. **M. Gales, S. Young.** „The Application of Hidden Markov Models in Speech Recognition“. Foundations and Trends in Signal Processing, Vol. 1, No. 3, 2007, p. 195 – 304.
10. **S. Theodoridis, K. Koutroumbas.** „Pattern Recognition Fourth Edition“. ISBN: 978-1-59749-272-0, Academic Press, USA, 2009.
11. **Tebelskis J.** Speech Recognition using Neural Networks, 1995, p 190. Interaktyvus.[žiūrėta 2016-04-19]. Prieiga per internetą: <http://isl.anthropomatik.kit.edu/pdf/Tebelskis1995.pdf> >.
12. **Grigas, N.** Elektros prietaisų valdymo balsu programa: Baigiamasis bakalauro projektas: technologijos mokslai, Auromatika ir valdymas / Kauno technikos universitetas, Kaunas , 2012, p 54.
13. **Laurinčiukaitė, S.** Lietuvių šnekos atpažinimo akustinis modeliavimas: daktaro disertacija: technologijos mokslai, informatikos inžinerija 07T / Vilniaus Gedimino technikos universitetas, Vilnius: Technika, 2012, p 134. Interaktyvus [žiūrėta 2016-03-29]. Prieiga per internetą: [www.mii.lt/files/mii\\_dis\\_08\\_laurinciukaite.pdf](http://www.mii.lt/files/mii_dis_08_laurinciukaite.pdf)

- 14. Lileikytė, R.** Šnekos atpažinimo požymių kokybės vertinimas: daktaro disertacija: Technologijos mokslai, informatikos inžinerija (07T) / Vilniaus universitetas, 2012. Interaktyvus [žiūrėta 2014-04-01]. Prieiga per internetą: <[http://www.mii.lt/files/mii\\_dis\\_2012\\_lileikyte.pdf](http://www.mii.lt/files/mii_dis_2012_lileikyte.pdf)>
- 15. Maskeliūnas, R.** Lietuviškų balso komandų atpažinimas daugybinių transkripcijų pagrindu: daktaro disertacija: technologijos mokslai, informatikos inžinerija 07T / Kauno technikos universitetas, Kaunas: Technika, 2009, p. 158.
- 16. Pašauskaitė, G.** Balsu ištartų atpažinimo sistemos sukūrimas ir tyrimas: Baigiamasis magistro projektas: technologijos mokslai, Auromatika ir valdymas / Kauno technikos universitetas, Kaunas. 2015, p. 70.
- 17. Tamulevičius, G.** Pavienių žodžių atpažinimo sistemų kūrimas: daktaro disertacija: technologijos mokslai, informatikos inžinerija 07T / Vilniaus Gedimino technikos universitetas, Vilnius: Technika. 2008, p. 134. Interaktyvus [žiūrėta 2016-04-10]. Prieiga per internetą: <[http://donelaitis.vdu.lt/disertacijos/Tamulevicius\\_2008.pdf](http://donelaitis.vdu.lt/disertacijos/Tamulevicius_2008.pdf)>
- 18. Vačiulytė, J.** Automatinio šnekos atpažinimo metodų tyrimas ir taikymai balso įrašams stenografuoti: Magistro baigiamasis darbas: technologijos mokslai, Šiaulių niversitetas, informatikos katedra, 2015, p. 45.
- 19. Vaičiūnas, A.** Lietuvių kalbos statistinių modelių ir jų taikymo šnekos atpažinimui tyrimas, kai naudojami labai dideli žodynai: daktaro disertacija: fiziniai mokslai, informatika 09P / Vytauto Didžiojo universitetas, Kaunas, 2006. Interaktyvus [žiūrėta 2016-05-19]. Prieiga per internetą: <[http://donelaitis.vdu.lt/disertacijos/Vaiciunas\\_2006.pdf](http://donelaitis.vdu.lt/disertacijos/Vaiciunas_2006.pdf)>.
- 20. Valentaitė, I.** Lietuviškų vardų atpažinimo tikslumo tyrimas: Bakalauro baigiamasis projektas: KTU Socialinių, humanitarinių mokslų ir menų fakultetas, 2014, p. 82.
- 21. Raškinis, A., Raškinis, G., Kazlauskienė, A.** Lietuvių kalbos kompiuteriniai tyrimai (IX sekcija): VDU bendrinės lietuvių šnekos universalus anotuotas garsynas / Vytauto Didžiojo universitetas, 2003 Interaktyvus [žiūrėta 2015-04-12]. Prieiga per internetą: [http://www.ktu.lt/lt/apie\\_renginius/konferencijos/2006/k6\\_02/it2003/sekcija09.pdf](http://www.ktu.lt/lt/apie_renginius/konferencijos/2006/k6_02/it2003/sekcija09.pdf)
- 22. Rudžionis, A., Ratkevičius K., Rudžionis V., Kasparaitis P., Šalna B.** Balso technologijų taikymo lietuvių kalbai analizė ir perspektyvinių veiklos krypčių pagrindimas: ataskaita / Kauno technologijos universitetas, Vilniaus universitetas, Teismo ekspertizės centras. Kaunas, Vilnius, 2001, p. 84. Interaktyvus [žiūrėta 2015-04-11]. Prieiga per internetą: <[http://www.likit.lt/frames/balso\\_tech/balsotech\\_st.htm](http://www.likit.lt/frames/balso_tech/balsotech_st.htm)>.
- 23. Kasparaitis, P.** Kompiuterinė lingvistika. SAPI5 architektūra, 2005, p. 8 p. [žiūrėta 2016-04-11 d.]. Prieiga per internetą: <<http://www.mif.vu.lt/~pijus/CL/Sapi.pdf>>.



24. **F. Hutter, H. Hoos, K. Leyton-Brown, K. Murphy.** „Time-Bounded Sequential Parameter Optimization. 2017, p. 15. Interaktyvus [žiūrėta 2017-04-10]. Prieiga per internetą : <<https://www.cs.ubc.ca/~murphyk/Papers/hutter-lion10.pdf>>.
25. **L. Kattoff, C. Thornot, F.Hutter.** „User Guide for Auto-Weka version 2.3“. 2017, p. 15. Interaktyvus [žiūrėta 2017-04-10]. Prieiga per internetą : <<http://www.cs.ubc.ca/labs/beta/Projects/autoweka/manual.pdf>>.
26. **Lebedev, S.** Išmanaus namo įrenginių valdymo balsu sistemos sukūrimas: Baigiamasis bakalauro projektas: technologijos mokslai, Robotika / Kauno technikos universitetas, Kaunas ,2016 ,p. 35.
27. Kalbos atpažinimo programinis įrakis Julius Oficialus tinklalapis - [žiūrėta 2016-05-10]. Prieiga per internetą: <[http://julius.osdn.jp/en\\_index.php](http://julius.osdn.jp/en_index.php)> .
28. Projektas „LIEPA“ .Oficialus tinklalapis - [žiūrėta 2016-04-10]. Prieiga per internetą: <<https://www.raštija.lt/liepa/paslaugos-vartotojams/7469>>.
29. Vilniaus Gedimino technikos universiteto distancinio mokymo kurso medžiaga, „Šiuolaikinės SSA priemonės“. Oficialus tinklalapis Prieiga per internetą: <<http://www2.el.vgtu.lt/ssa/node1.html>>.
30. Duomenų analizės sistemos paketas WEKA Oficialus tinklalapis - [žiūrėta 2017-02-10]. Prieiga per internetą: <<http://www.cs.waikato.ac.nz/ml/weka>>.

# PRIEDAI

## Priedas 1. Komandos Antras PMM.

### Skaičiaus „antras“ modelis be papildomų būsenų ir Gauso mišinių :

```
~o
<STREAMINFO> 1 39
<VECSIZE> 39<NULLD><MFCC_E_D_A>
~h "hmm_antras"
<BEGINHMM>
<NUMSTATES> 6
<STATE> 2
<MEAN> 39
-1.635420e+001 -7.481701e+000 -3.241516e+000 -6.746332e+000 -1.905021e+000 -2.098491e-001 -3.556307e+000 -1.653795e+000 -
3.160717e+000 -7.614004e-002 -1.860593e+000 -2.659124e+000 5.936241e-001 1.045115e-001 -9.938418e-002 1.852256e-003 -
1.078802e-001 -7.139887e-002 -5.955098e-002 -1.166002e-001 -1.363539e-002 5.293483e-002 -1.531312e-002 1.978809e-002
1.269146e-003 1.283406e-002 5.581629e-002 -1.794550e-002 1.215951e-003 -2.029217e-002 -3.292636e-002 -1.788949e-002 -
6.102315e-002 -6.571918e-003 2.086528e-002 -6.610165e-003 -8.879054e-003 -7.258497e-003 1.663400e-003
<VARIANCE> 39
6.120396e+001 4.128733e+001 3.451609e+001 3.580319e+001 3.172113e+001 4.763743e+001 3.663995e+001 3.402687e+001
3.737448e+001 2.752102e+001 1.899535e+001 1.969127e+001 9.426043e-002 2.934275e+000 2.367974e+000 2.063026e+000
2.087210e+000 1.616394e+000 2.117690e+000 2.130674e+000 2.043434e+000 1.970868e+000 1.790762e+000 1.261116e+000
1.263335e+000 4.924380e-003 4.322110e-001 3.162542e-001 3.185689e-001 3.113748e-001 2.609292e-001 3.112812e-001 3.507734e-
001 3.450337e-001 2.746606e-001 3.081366e-001 2.438419e-001 2.339956e-001 6.489430e-004
<GCONST> 9.249604e+001
<STATE> 3
<MEAN> 39
-1.318572e+001 -1.092466e+001 -3.567029e+000 -8.906710e+000 -5.294247e+000 -3.115153e+000 -9.918097e+000 -8.572701e-001
3.315541e+000 1.189654e+000 -3.244779e+000 -2.416975e+000 7.924251e-001 -2.989916e-001 2.716010e-001 -3.967401e-003
1.676212e-001 2.606258e-001 -1.957146e-002 1.647976e-001 9.175546e-002 -2.698870e-002 -2.544180e-002 -5.323623e-002 -
2.549073e-002 1.242388e-003 -2.949355e-001 9.700260e-002 -1.024206e-002 1.149120e-001 1.137591e-001 4.025037e-002
2.146648e-001 2.481181e-002 -5.191229e-002 8.701910e-003 7.540419e-002 5.408960e-002 -1.930665e-003
<VARIANCE> 39
3.580811e+001 2.824692e+001 2.052508e+001 3.091581e+001 2.972331e+001 2.018601e+001 4.127357e+001 2.837565e+001
1.784853e+001 3.057835e+001 4.629311e+001 2.164336e+001 1.337320e-002 5.667546e+000 2.570634e+000 1.403797e+000
3.105717e+000 2.898274e+000 2.067309e+000 4.647258e+000 2.446209e+000 1.325656e+000 2.214878e+000 2.729871e+000
1.644816e+000 1.444710e-003 5.731558e-001 3.172156e-001 2.745804e-001 4.429399e-001 4.641698e-001 3.450713e-001 5.774266e-
001 4.449961e-001 2.547179e-001 4.131224e-001 4.264395e-001 3.026484e-001 3.458421e-004
<GCONST> 9.255043e+001
<STATE> 4
<MEAN> 39
-2.470575e+001 3.776916e-001 -1.676054e+000 -2.151704e-001 1.914982e+000 -1.285746e+000 1.714051e+000 6.886468e-002 -
3.306908e-001 5.408483e-001 -1.055583e+000 -4.701853e-001 5.259013e-001 -7.239407e-003 2.515301e-003 1.738524e-002
6.713683e-002 -2.444293e-002 9.396279e-002 9.875386e-002 -3.500801e-003 -4.676116e-002 4.179469e-002 2.350975e-002
4.301067e-002 -1.334955e-002 7.581310e-002 -3.292413e-002 2.233062e-003 -3.639250e-002 -2.311831e-002 -3.601949e-003 -
4.537170e-002 -1.322988e-002 7.755023e-005 2.691251e-003 -2.245894e-002 -1.424444e-002 -1.082953e-003
<VARIANCE> 39
4.148885e+001 3.061886e+001 2.576120e+001 2.764822e+001 2.228556e+001 2.689502e+001 1.900113e+001 1.432310e+001
1.696751e+001 1.265472e+001 9.969807e+000 7.840863e+000 6.985159e-002 1.399555e+000 1.302544e+000 1.089158e+000
1.225884e+000 1.213734e+000 1.278880e+000 1.260468e+000 1.272284e+000 1.154299e+000 1.009454e+000 9.119945e-001
7.514822e-001 1.212041e-003 2.275359e-001 2.073549e-001 1.936593e-001 2.340265e-001 2.361930e-001 2.499457e-001 2.643826e-
001 2.593025e-001 2.320609e-001 2.062221e-001 1.987312e-001 1.591521e-001 1.676338e-004
<GCONST> 7.247847e+001
<STATE> 5
<MEAN> 39
-2.756773e+001 -4.196552e+000 -3.883252e+000 -1.323063e+000 -1.232651e-003 -1.224757e+000 1.604688e+000 -5.457349e-001 -
1.444138e-001 1.504017e+000 -1.986233e+000 1.657649e-001 2.275809e-001 -7.648911e-002 -3.981552e-002 -1.662688e-002 -
2.992043e-002 -4.695912e-002 -1.344791e-002 -7.876813e-002 -6.394372e-002 -2.538627e-003 -5.452407e-003 -1.976138e-002 -
2.387349e-002 -1.008535e-003 1.385147e-002 9.915282e-003 1.313461e-002 8.234767e-003 9.245361e-003 3.810984e-003 9.927879e-
003 1.127801e-002 5.141106e-003 -5.243152e-003 -1.689574e-004 -4.119486e-004 4.229894e-004
<VARIANCE> 39
3.964005e+000 3.075687e+000 6.579390e+000 6.206078e+000 7.482400e+000 8.622754e+000 1.149526e+001 9.062066e+000
9.797117e+000 9.685256e+000 8.580899e+000 6.137663e+000 1.719897e-002 1.043460e-001 2.044517e-001 3.488092e-001
4.221535e-001 5.794712e-001 6.739652e-001 7.358613e-001 7.340000e-001 7.611475e-001 6.951320e-001 7.400761e-001 6.022880e-
001 2.609311e-005 1.619741e-002 3.665504e-002 6.310803e-002 7.875957e-002 1.130026e-001 1.269752e-001 1.427574e-001
1.424323e-001 1.438851e-001 1.313062e-001 1.477067e-001 1.183688e-001 4.791806e-006
<GCONST> 3.069950e+001
```

```
<TRANSP> 6
0.000000e+000 1.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000 9.752545e-001 2.474552e-002 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000 0.000000e+000 9.455945e-001 5.440552e-002 0.000000e+000 0.000000e+000
0.000000e+000 0.000000e+000 0.000000e+000 9.731176e-001 2.688232e-002 0.000000e+000
0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 9.506513e-001 4.934864e-002
0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
<ENDHMM>
```

## Skaičiaus „antras“ modelis su dviem papildomomis būsenomis ir be papildomų Gauso mišinių :

```
~o
<STREAMINFO> 1 39
<VECSIZE> 39<NULLD><MFCC_E_D_A>
~h "hmm_antras"
<BEGINHMM>
<NUMSTATES> 8
<STATE> 2
<MEAN> 39
-2.224034e+001 -9.435076e+000 -5.048159e+000 -3.810936e+000 -1.629610e+000 5.725823e-001 -2.405606e-001 -1.268536e+000 -
2.748012e-001 2.272855e-001 -1.334968e+000 -9.159253e-001 4.639516e-001 7.446030e-001 -5.810293e-001 -4.765591e-001 -
5.830492e-001 -1.988819e-001 4.328637e-001 -8.778036e-002 -2.439832e-001 -9.774882e-002 -5.955769e-002 -9.628293e-002 -
2.270138e-001 4.047811e-002 3.796544e-003 -6.970916e-002 -1.713630e-002 2.046285e-002 3.698641e-002 3.121348e-002
1.851000e-002 -1.411188e-002 1.791200e-002 1.328690e-002 -1.215760e-004 -2.230594e-003 1.302925e-003
<VARIANCE> 39
7.678761e+001 2.150076e+001 2.296858e+001 2.739786e+001 1.670839e+001 1.906981e+001 1.212172e+001 1.498827e+001
1.174800e+001 1.034145e+001 1.442731e+001 9.660731e+000 9.676021e-002 5.007016e+000 1.612759e+000 1.764590e+000
1.977036e+000 1.633161e+000 1.678805e+000 1.447080e+000 1.442976e+000 1.186843e+000 1.061935e+000 1.373086e+000
1.003093e+000 2.371406e-003 7.475359e-001 2.602972e-001 2.929242e-001 3.597002e-001 3.047428e-001 2.771697e-001 2.572564e-
001 2.568433e-001 2.177816e-001 2.029279e-001 2.607328e-001 1.851808e-001 3.822480e-004
<GCONST> 8.039425e+001
<STATE> 3
<MEAN> 39
-9.202844e+000 -1.784840e+001 -1.217256e+001 -1.183318e+001 -3.095848e+000 8.443068e+000 -1.073280e+000 -5.621612e+000 -
2.046721e+000 1.637791e-001 -2.085003e+000 -3.762616e+000 9.431213e-001 7.933047e-002 3.914866e-001 1.075519e-001 -
4.638899e-001 -2.844626e-001 1.237544e-001 1.175126e-001 -1.375105e-002 -1.147519e-001 1.645544e-002 2.695060e-001
2.264875e-001 -4.007293e-003 1.179085e-002 2.507535e-001 7.862370e-002 -1.245352e-001 -2.016698e-001 -6.676440e-002 -
5.976374e-002 4.769512e-002 -6.505015e-002 -4.348942e-002 2.448581e-002 7.685947e-002 -5.062681e-003
<VARIANCE> 39
4.234580e+000 2.502983e+001 3.875093e+001 2.396964e+001 3.487826e+001 2.726643e+001 3.871782e+001 3.550382e+001
2.314047e+001 2.572614e+001 2.550534e+001 1.276445e+001 3.815162e-003 1.003794e-001 7.210637e-001 2.970866e-001
8.969572e-001 1.140358e+000 5.654069e-001 7.949201e-001 6.156223e-001 6.004307e-001 5.574681e-001 6.019158e-001 6.372266e-
001 1.948667e-004 2.110798e-002 5.991885e-002 5.937750e-002 8.351576e-002 9.623849e-002 9.231465e-002 1.228870e-001
9.360448e-002 1.035513e-001 1.114756e-001 1.015010e-001 9.188274e-002 3.773835e-005
<GCONST> 4.767663e+001
<STATE> 4
<MEAN> 39
-8.442029e+000 -6.857489e+000 -7.571534e+000 -1.905339e+001 -1.189935e+001 7.915112e+000 -2.696264e+000 -5.452076e+000 -
4.901017e+000 -2.306584e+000 6.229540e-001 2.695398e-001 7.545667e-001 -2.346797e-001 1.152057e+000 7.212335e-001
3.762238e-001 -7.566272e-002 -5.704480e-001 -9.881273e-001 -3.341198e-001 -2.863970e-001 -2.558100e-001 4.629203e-002
1.488702e-001 -3.759138e-002 -2.666767e-001 -2.374715e-001 6.330378e-003 5.236853e-001 6.512154e-001 -2.257593e-001 -
3.358182e-002 5.142982e-003 1.635445e-001 1.502222e-001 -7.425988e-002 -1.937088e-001 -3.705083e-004
<VARIANCE> 39
3.197255e+000 2.510438e+001 4.571828e+001 2.227342e+001 2.183280e+001 1.870118e+001 2.731335e+001 1.912001e+001
1.289325e+001 2.545442e+001 1.817407e+001 1.385645e+001 8.717831e-003 2.558284e-001 8.581850e-001 6.158238e-001
1.912799e+000 3.270812e+000 1.137606e+000 1.306167e+000 1.408389e+000 1.522582e+000 1.698651e+000 1.241562e+000
1.283266e+000 1.682975e-004 8.952072e-002 9.003301e-002 1.220616e-001 1.998668e-001 2.031175e-001 2.019373e-001 2.038536e-
001 3.312033e-001 2.412291e-001 3.071065e-001 2.060015e-001 1.816065e-001 4.051255e-005
<GCONST> 6.471024e+001
<STATE> 5
<MEAN> 39
-1.853772e+001 -4.310448e+000 -3.552770e+000 -8.164673e+000 6.999109e-002 -9.210060e-001 -5.185853e+000 -3.570730e+000 -
2.273977e+000 -5.542932e-001 -6.285675e-001 -1.316124e+000 6.387384e-001 -3.694398e+000 4.883115e-001 7.117413e-001
2.315238e+000 2.588073e+000 -1.890787e+000 1.140414e+000 1.644094e+000 9.774515e-001 3.699464e-001 -2.925389e-001
8.762685e-002 3.459723e-002 -1.716473e-001 2.419996e-001 -1.576937e-001 -2.212598e-001 -4.635184e-001 2.471579e-001
3.526976e-001 9.688067e-002 -4.828326e-002 -6.913028e-002 5.979405e-002 1.498669e-001 1.687993e-002
<VARIANCE> 39
5.608731e+001 1.512457e+001 1.550771e+001 3.388330e+001 4.023286e+001 3.768227e+001 3.362915e+001 3.387448e+001
1.589798e+001 1.927535e+001 2.768580e+001 9.615556e+000 1.589037e-002 1.728838e+000 2.010970e+000 1.912367e+000
3.270527e+000 3.219346e+000 2.862995e+000 4.868194e+000 2.314469e+000 1.620272e+000 2.234421e+000 2.024268e+000
```

```

1.285169e+000 2.784873e-003 9.388565e-001 2.640010e-001 2.997254e-001 4.524141e-001 7.014796e-001 6.192534e-001 9.277158e-
001 7.497192e-001 3.347042e-001 3.563234e-001 3.824834e-001 2.259316e-001 3.741827e-004
<GCONST> 9.323972e+001
<STATE> 6
<MEAN> 39
-2.258955e+001 -4.882915e+000 -5.686263e+000 -2.795079e+000 2.085170e-001 -1.321276e+000 -1.729041e+000 -1.035718e-001
5.699197e-001 -9.616368e-001 -3.408606e+000 -7.629361e-001 5.237163e-001 9.378217e-002 -9.456268e-002 -6.517871e-002
7.539794e-002 -1.017307e-001 1.190223e-001 4.178457e-002 -1.762026e-002 -1.369701e-002 2.514965e-003 -1.710978e-002
2.899183e-002 -1.674196e-002 4.851230e-002 -2.785517e-002 8.103525e-003 -3.368272e-002 -1.800207e-002 6.916074e-003 -
2.316036e-002 -2.312394e-002 -1.095767e-002 4.331213e-003 2.428368e-003 -6.148133e-003 -1.660002e-003
<VARIANCE> 39
5.800303e+001 4.992685e+001 2.631833e+001 3.715316e+001 2.558200e+001 2.557890e+001 4.252958e+001 1.828900e+001
3.098787e+001 3.456209e+001 2.421217e+001 1.654825e+001 7.863732e-002 2.123047e+000 2.073200e+000 1.367388e+000
1.797721e+000 1.608339e+000 1.645812e+000 2.232048e+000 1.591902e+000 1.760363e+000 1.837532e+000 1.609011e+000
1.299625e+000 4.056156e-003 2.928524e-001 2.888820e-001 2.331598e-001 2.736689e-001 2.939970e-001 2.897603e-001 3.809552e-
001 3.115594e-001 2.982334e-001 3.060435e-001 2.754297e-001 2.424458e-001 6.005454e-004
<GCONST> 8.879329e+001
<STATE> 7
<MEAN> 39
-2.763654e+001 -4.790185e+000 -3.605862e+000 -3.522055e-001 -1.101045e+000 -6.098756e-001 1.046809e+000 -6.192632e-001 -
5.679334e-001 1.430753e+000 -1.935552e+000 -5.076191e-001 9.939103e-002 -1.010352e-001 3.934089e-002 5.646769e-002 -
1.111293e-002 -7.340685e-002 -3.848038e-002 -4.948992e-003 -4.321055e-002 1.000698e-003 3.810209e-002 -2.463884e-002 -
6.569945e-002 -1.035833e-003 2.189754e-002 -9.940707e-003 -7.760807e-003 4.328845e-003 1.119572e-002 5.232370e-003 -
1.058264e-002 5.971968e-003 -3.660016e-003 -1.042358e-002 -4.525979e-003 -4.303210e-003 3.614624e-004
<VARIANCE> 39
2.594506e+000 3.283653e+000 6.318902e+000 6.876115e+000 7.291316e+000 7.273031e+000 8.128176e+000 9.138272e+000
9.697571e+000 9.353967e+000 7.207251e+000 7.600332e+000 1.640218e-002 1.090791e-001 2.124203e-001 3.463729e-001
4.200224e-001 5.165685e-001 6.056819e-001 6.864191e-001 7.603245e-001 7.361273e-001 8.121951e-001 6.952008e-001 6.576343e-
001 2.619833e-005 1.832814e-002 3.829753e-002 6.107337e-002 7.993657e-002 1.036041e-001 1.168243e-001 1.368370e-001
1.503360e-001 1.424157e-001 1.635571e-001 1.391274e-001 1.332673e-001 4.581994e-006
<GCONST> 2.999607e+001
<TRANSP> 8
0.000000e+000 1.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000 9.471146e-001 5.288544e-002 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000 0.000000e+000 9.161847e-001 8.381531e-002 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000 0.000000e+000 0.000000e+000 8.526058e-001 1.473942e-001 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 8.060100e-001 1.939900e-001 0.000000e+000 0.000000e+000
0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 9.749369e-001 2.506310e-002 0.000000e+000
0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 9.652961e-001 3.470387e-002
0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
<ENDHMM>

```

## Skaičiaus „antras“ modelis su dviem papildomomis būsenomis ir su dviem papildomais Gauso mišiniais :

```

~o
<STREAMINFO> 1 39
<VECSIZE> 39<NULLD><MFCC_E_D_A>
~h "hmm_antras"
<BEGINHMM>
<NUMSTATES> 8
<STATE> 2
<NUMMIXES> 2
<MIXTURE> 1 0.5
<MEAN> 39
-2.042068e+001 -8.923848e-001 6.197814e+000 -6.973698e+000 -1.052477e+001 -3.465064e+000 -6.566501e+000 -6.070755e+000 -
6.233189e+000 1.300967e+000 -4.177918e+000 -6.200885e-001 6.330425e-001 1.011781e-002 7.701541e-002 -2.121529e-001
1.401554e-001 -6.718002e-002 -1.179366e-001 1.098138e-001 -1.704486e-002 6.214689e-002 -7.688358e-002 5.494358e-002
5.979980e-002 -3.523110e-003 2.218961e-002 -3.043799e-002 -1.170038e-001 9.025424e-002 -1.097904e-002 -1.206738e-001
5.613745e-002 2.109582e-002 9.728801e-003 -4.673741e-002 6.592489e-002 4.485193e-002 3.891919e-004
<VARIANCE> 39
1.585973e+001 1.420715e+001 9.535910e+001 4.966489e+001 6.136670e+001 2.501188e+001 5.003208e+001 3.653926e+001
3.826283e+001 3.545856e+001 3.133808e+001 1.774069e+001 1.231384e-001 1.400327e-001 3.269013e-001 4.616032e-001
4.682965e-001 7.635614e-001 9.293998e-001 8.307741e-001 7.852990e-001 7.934859e-001 7.808621e-001 7.324698e-001 5.820923e-
001 1.517821e-004 2.809575e-002 5.297998e-002 7.862090e-002 9.452185e-002 1.415429e-001 1.373536e-001 1.422058e-001
1.420333e-001 1.425058e-001 1.613213e-001 1.309799e-001 1.050639e-001 8.814284e-005
<GCONST> 5.968438e+001
<MIXTURE> 2 0.5
<MEAN> 39

```

-2.013910e+001 -1.554538e+000 2.130829e+000 -4.594718e+000 -9.993279e+000 -7.021964e+000 -4.452765e+000 -4.509310e+000 -5.366842e+000 -3.674736e-001 -2.044261e+000 5.954765e-002 6.811878e-001 1.614682e-001 2.338288e-002 5.649990e-001 -2.243853e-001 -5.039554e-001 -3.914024e-001 -3.166579e-001 -2.285825e-001 -2.732812e-001 2.500196e-002 -4.710709e-002 9.918462e-002 2.806606e-002 -6.861713e-002 9.198548e-002 -1.388376e-002 1.938332e-002 9.890646e-002 9.777014e-002 8.268993e-003 2.338418e-002 4.631674e-002 -1.439677e-002 -2.994410e-002 -1.683951e-002 -1.061693e-003

<VARIANCE> 39  
2.287702e+001 2.152219e+001 6.258891e+001 3.369932e+001 5.759293e+001 3.949613e+001 3.058543e+001 3.118122e+001 3.157868e+001 2.492481e+001 2.136730e+001 1.502354e+001 6.799258e-002 2.466468e+000 1.937276e+000 3.740450e+000 2.507331e+000 4.961355e+000 4.018602e+000 2.537044e+000 2.534334e+000 2.401454e+000 2.323826e+000 1.941323e+000 1.342319e+000 4.615553e-003 5.254085e-001 3.760918e-001 4.718153e-001 3.634428e-001 1.052032e+000 7.591074e-001 4.575962e-001 4.968810e-001 4.482580e-001 3.546348e-001 3.300207e-001 2.375163e-001 7.122869e-004

<GCONST> 9.910361e+001  
<STATE> 3  
<NUMMIXES> 2  
<MIXTURE> 1 0.5  
<MEAN> 39  
-2.872570e+001 8.107331e+000 -8.462882e+000 4.395343e+000 -3.239334e+000 -1.652872e+000 -2.872872e-001 -1.113325e+000 6.898523e-002 -4.523702e-001 -5.206417e-002 9.294522e-001 8.369013e-001 -1.348732e-002 -1.461253e-001 -1.040093e-001 -3.155515e-002 2.030972e-001 2.901814e-001 7.950487e-002 1.955587e-002 5.418148e-002 8.685634e-002 -1.359159e-001 -1.600641e-001 -1.044781e-002 1.095615e-001 -1.514440e-001 1.213490e-001 -1.076894e-001 -6.295859e-002 -1.334659e-001 -1.560277e-002 -5.885288e-002 -4.504165e-002 -5.291352e-002 -3.499404e-002 -4.850917e-003 -5.951312e-003

<VARIANCE> 39  
5.500633e+000 1.919014e+001 8.021699e+000 1.091546e+001 1.769786e+001 1.864742e+001 1.532973e+001 1.043314e+001 1.030747e+001 8.746243e+000 8.103961e+000 6.034599e+000 1.332301e-002 3.990799e-001 7.367094e-001 4.843076e-001 5.494981e-001 5.742678e-001 7.760509e-001 7.327683e-001 7.723723e-001 7.121204e-001 7.376012e-001 6.737109e-001 5.735466e-001 5.170217e-004 1.039824e-001 1.132762e-001 1.004947e-001 9.770069e-002 1.334285e-001 1.500234e-001 1.468545e-001 1.612394e-001 1.539059e-001 1.528871e-001 1.432771e-001 1.193415e-001 8.166770e-005

<GCONST> 4.871796e+001  
<MIXTURE> 2 0.5  
<MEAN> 39  
-2.335729e+001 4.884802e+000 -2.488992e+000 1.344309e+000 -8.465964e+000 -9.211511e+000 -2.126368e+000 -4.042031e+000 -2.815417e+000 -3.445592e+000 -5.134268e-001 2.448218e+000 7.558542e-001 -2.110135e+000 1.487766e+000 -2.066937e+000 1.364159e+000 2.155301e+000 1.412246e+000 1.077299e+000 1.132666e+000 1.236201e+000 1.781826e-001 4.568086e-001 -7.978114e-002 2.826611e-002 3.863931e-002 -1.761236e-001 2.563801e-001 -3.209911e-001 -2.213591e-001 2.829556e-001 -1.834939e-001 -6.237684e-002 -1.471131e-001 4.154226e-001 -3.890483e-002 -1.285883e-001 8.156298e-003

<VARIANCE> 39  
2.463126e+001 1.799234e+001 2.104924e+001 1.856295e+001 3.879193e+001 2.593937e+001 3.047203e+001 2.034790e+001 1.850941e+001 1.609296e+001 1.792523e+001 1.243703e+001 1.738227e-002 1.153755e+000 2.035185e+000 1.486460e+000 2.793319e+000 2.125411e+000 2.373443e+000 2.569979e+000 1.959746e+000 1.840599e+000 2.924544e+000 2.226105e+000 1.518368e+000 1.405348e-003 4.998006e-001 3.005824e-001 1.923046e-001 2.736934e-001 4.696732e-001 4.888027e-001 4.348677e-001 3.743523e-001 3.443286e-001 4.929680e-001 3.779147e-001 2.592022e-001 2.940678e-004

<GCONST> 8.562492e+001  
<STATE> 4  
<NUMMIXES> 2  
<MIXTURE> 1 0.5  
<MEAN> 39  
-2.235646e+001 -9.180241e-001 -5.666777e+000 1.577423e-001 -1.076693e+000 -4.556596e-001 -6.550251e-001 -9.578027e-001 -9.641941e-001 -9.647576e-001 -1.769981e+000 -1.201897e+000 2.338116e-001 2.038768e-001 -5.607866e-003 6.420328e-002 9.255057e-002 -2.692704e-002 2.515347e-001 5.003777e-002 2.959998e-003 -5.006030e-002 -4.101387e-002 3.340326e-002 -3.325016e-002 -2.524204e-002 -1.550421e-001 1.263244e-001 -2.210924e-003 2.229116e-002 -5.505619e-003 -3.721339e-002 5.851221e-002 -6.751872e-003 1.217733e-002 3.518476e-002 1.994713e-002 9.129198e-003 1.190723e-002

<VARIANCE> 39  
5.561412e+000 1.063594e+001 9.529383e+000 1.022407e+001 1.171827e+001 1.213078e+001 1.457921e+001 1.484935e+001 1.182074e+001 1.253242e+001 1.123659e+001 8.575001e+000 1.648472e-002 3.975174e-001 5.474601e-001 6.299397e-001 8.783894e-001 1.103457e+000 1.063707e+000 1.189366e+000 1.121244e+000 1.214456e+000 1.037277e+000 9.521356e-001 7.722362e-001 9.471345e-004 5.834745e-002 1.181435e-001 1.155992e-001 1.633291e-001 2.066334e-001 1.998391e-001 2.258854e-001 2.211320e-001 2.155264e-001 2.114474e-001 1.913038e-001 1.524979e-001 1.357204e-004

<GCONST> 5.676590e+001  
<MIXTURE> 2 0.5  
<MEAN> 39  
-2.708500e+001 2.109504e+000 -6.196382e+000 -1.046138e-001 -8.763934e-001 -2.037864e+000 -7.422319e-002 -1.150411e+000 -6.227893e-001 3.580226e-001 -1.643995e+000 -6.142570e-001 5.872476e-001 8.338419e-001 -1.303796e+000 2.754999e-001 -6.197311e-001 1.242598e-001 -6.280512e-002 -2.024779e-001 3.218675e-002 -1.769971e-001 -6.550252e-002 -6.778578e-002 -1.886816e-001 -7.513892e-002 1.315946e-001 5.896227e-002 -3.088307e-002 1.264918e-001 -2.709933e-002 1.166573e-001 -6.517309e-002 3.919019e-002 -2.354181e-002 -7.638054e-002 1.879531e-002 4.484680e-002 -7.690480e-003

<VARIANCE> 39  
8.429265e+000 2.193938e+001 1.599471e+001 2.136557e+001 2.759167e+001 2.233913e+001 2.155847e+001 1.633491e+001 1.736330e+001 1.317918e+001 1.132262e+001 1.024677e+001 3.391748e-002 6.120156e-001 1.147130e+000 1.041910e+000 1.214359e+000 1.615375e+000 1.584367e+000 1.482397e+000 1.622903e+000 1.383483e+000 1.249648e+000 1.132468e+000

9.317443e-001 1.724160e-003 1.104958e-001 2.314942e-001 1.767621e-001 2.289604e-001 3.121437e-001 2.792797e-001 2.885087e-001 3.094684e-001 2.973017e-001 2.402075e-001 2.307887e-001 1.844940e-001 2.618249e-004  
 <GCONST> 7.198704e+001  
 <STATE> 5  
 <NUMMIXES> 2  
 <MIXTURE> 1 0.5  
 <MEAN> 39  
 -2.309502e+001 -1.699957e+000 -6.812952e+000 -1.502924e+000 -2.704899e+000 -7.678589e-001 -8.316554e-001 -1.703271e+000 -1.767646e+000 -1.650425e+000 -1.904234e+000 -1.334583e+000 1.926288e-001 -1.519826e-002 2.746881e-003 -7.521145e-004 -2.330649e-002 -1.452930e-002 4.449343e-003 8.697313e-003 -7.820139e-003 -1.568750e-002 -2.625489e-002 -1.901195e-002 3.897514e-003 -8.404683e-004 3.773958e-003 9.668951e-003 2.375328e-002 3.785788e-002 3.847256e-002 2.517306e-002 1.535971e-002 1.200692e-002 9.967598e-003 1.356440e-002 6.703648e-003 6.292262e-003 3.603999e-004  
 <VARIANCE> 39  
 1.691265e+000 2.658423e+000 3.097668e+000 3.931717e+000 5.472877e+000 6.970726e+000 8.204988e+000 1.057566e+001 1.084378e+001 1.014274e+001 9.014739e+000 7.005922e+000 9.245542e-003 6.427964e-002 1.592140e-001 2.741498e-001 3.986159e-001 5.269048e-001 6.577153e-001 7.463201e-001 8.213047e-001 9.293794e-001 8.979327e-001 7.947415e-001 6.757315e-001 1.654950e-005 1.253725e-002 3.059860e-002 5.173211e-002 7.325652e-002 9.896697e-002 1.254741e-001 1.455501e-001 1.577117e-001 1.837488e-001 1.811525e-001 1.561922e-001 1.341105e-001 3.328579e-006  
 <GCONST> 2.630504e+001  
 <MIXTURE> 2 0.5  
 <MEAN> 39  
 -2.287300e+001 8.440299e-001 -3.010209e+000 2.445473e+000 7.964596e-001 1.627146e+000 1.070139e+000 -1.729109e-001 -5.655214e-001 -1.511322e-001 -6.126329e-001 -2.323046e-001 1.502536e-001 -3.626747e-002 -2.633287e-002 -2.099266e-002 -2.272648e-002 -2.126951e-002 -1.572300e-002 1.080966e-002 -6.597116e-003 1.060628e-002 9.410312e-003 1.435712e-003 5.752541e-003 -9.275221e-004 -3.092465e-003 -3.489682e-002 -7.191115e-002 -9.681918e-002 -9.970449e-002 -8.423927e-002 -5.090431e-002 -2.448103e-002 -2.015325e-002 -3.494697e-002 -3.345631e-002 -2.034126e-002 2.024452e-004  
 <VARIANCE> 39  
 4.576448e+000 3.217183e+000 4.874683e+000 4.912004e+000 6.159999e+000 7.889365e+000 9.313688e+000 1.379009e+001 1.202154e+001 9.293769e+000 9.381868e+000 7.398954e+000 1.674059e-002 8.200133e-002 1.771310e-001 2.971694e-001 4.469141e-001 5.628617e-001 6.396690e-001 7.327306e-001 7.714055e-001 8.278959e-001 8.482979e-001 7.271792e-001 6.123903e-001 2.995073e-005 1.488839e-002 3.219865e-002 5.410300e-002 8.164508e-002 1.053251e-001 1.238816e-001 1.505603e-001 1.612691e-001 1.716750e-001 1.630337e-001 1.475711e-001 1.232586e-001 5.644965e-006  
 <GCONST> 3.093649e+001  
 <STATE> 6  
 <NUMMIXES> 2  
 <MIXTURE> 1 0.5  
 <MEAN> 39  
 -2.309502e+001 -1.699957e+000 -6.812952e+000 -1.502924e+000 -2.704899e+000 -7.678589e-001 -8.316554e-001 -1.703271e+000 -1.767646e+000 -1.650425e+000 -1.904234e+000 -1.334583e+000 1.926288e-001 -1.519826e-002 2.746881e-003 -7.521145e-004 -2.330649e-002 -1.452930e-002 4.449343e-003 8.697313e-003 -7.820139e-003 -1.568750e-002 -2.625489e-002 -1.901195e-002 3.897514e-003 -8.404683e-004 3.773958e-003 9.668951e-003 2.375328e-002 3.785788e-002 3.847256e-002 2.517306e-002 1.535971e-002 1.200692e-002 9.967598e-003 1.356440e-002 6.703648e-003 6.292262e-003 3.603999e-004  
 <VARIANCE> 39  
 1.691265e+000 2.658423e+000 3.097668e+000 3.931717e+000 5.472877e+000 6.970726e+000 8.204988e+000 1.057566e+001 1.084378e+001 1.014274e+001 9.014739e+000 7.005922e+000 9.245542e-003 6.427964e-002 1.592140e-001 2.741498e-001 3.986159e-001 5.269048e-001 6.577153e-001 7.463201e-001 8.213047e-001 9.293794e-001 8.979327e-001 7.947415e-001 6.757315e-001 1.654950e-005 1.253725e-002 3.059860e-002 5.173211e-002 7.325652e-002 9.896697e-002 1.254741e-001 1.455501e-001 1.577117e-001 1.837488e-001 1.811525e-001 1.561922e-001 1.341105e-001 3.328579e-006  
 <GCONST> 2.630504e+001  
 <MIXTURE> 2 0.5  
 <MEAN> 39  
 -2.287300e+001 8.440299e-001 -3.010209e+000 2.445473e+000 7.964596e-001 1.627146e+000 1.070139e+000 -1.729109e-001 -5.655214e-001 -1.511322e-001 -6.126329e-001 -2.323046e-001 1.502536e-001 -3.626747e-002 -2.633287e-002 -2.099266e-002 -2.272648e-002 -2.126951e-002 -1.572300e-002 1.080966e-002 -6.597116e-003 1.060628e-002 9.410312e-003 1.435712e-003 5.752541e-003 -9.275221e-004 -3.092465e-003 -3.489682e-002 -7.191115e-002 -9.681918e-002 -9.970449e-002 -8.423927e-002 -5.090431e-002 -2.448103e-002 -2.015325e-002 -3.494697e-002 -3.345631e-002 -2.034126e-002 2.024452e-004  
 <VARIANCE> 39  
 4.576448e+000 3.217183e+000 4.874683e+000 4.912004e+000 6.159999e+000 7.889365e+000 9.313688e+000 1.379009e+001 1.202154e+001 9.293769e+000 9.381868e+000 7.398954e+000 1.674059e-002 8.200133e-002 1.771310e-001 2.971694e-001 4.469141e-001 5.628617e-001 6.396690e-001 7.327306e-001 7.714055e-001 8.278959e-001 8.482979e-001 7.271792e-001 6.123903e-001 2.995073e-005 1.488839e-002 3.219865e-002 5.410300e-002 8.164508e-002 1.053251e-001 1.238816e-001 1.505603e-001 1.612691e-001 1.716750e-001 1.630337e-001 1.475711e-001 1.232586e-001 5.644965e-006  
 <GCONST> 3.093649e+001  
 <STATE> 7  
 <NUMMIXES> 2  
 <MIXTURE> 1 0.5  
 <MEAN> 39  
 -2.309502e+001 -1.699957e+000 -6.812952e+000 -1.502924e+000 -2.704899e+000 -7.678589e-001 -8.316554e-001 -1.703271e+000 -1.767646e+000 -1.650425e+000 -1.904234e+000 -1.334583e+000 1.926288e-001 -1.519826e-002 2.746881e-003 -7.521145e-004 -2.330649e-002 -1.452930e-002 4.449343e-003 8.697313e-003 -7.820139e-003 -1.568750e-002 -2.625489e-002 -1.901195e-002

3.897514e-003 -8.404683e-004 3.773958e-003 9.668951e-003 2.375328e-002 3.785788e-002 3.847256e-002 2.517306e-002 1.535971e-002 1.200692e-002 9.967598e-003 1.356440e-002 6.703648e-003 6.292262e-003 3.603999e-004  
 <VARIANCE> 39  
 1.691265e+000 2.658423e+000 3.097668e+000 3.931717e+000 5.472877e+000 6.970726e+000 8.204988e+000 1.057566e+001  
 1.084378e+001 1.014274e+001 9.014739e+000 7.005922e+000 9.245542e-003 6.427964e-002 1.592140e-001 2.741498e-001  
 3.986159e-001 5.269048e-001 6.577153e-001 7.463201e-001 8.213047e-001 9.293794e-001 8.979327e-001 7.947415e-001 6.757315e-001  
 1.654950e-005 1.253725e-002 3.059860e-002 5.173211e-002 7.325652e-002 9.896697e-002 1.254741e-001 1.455501e-001  
 1.577117e-001 1.837488e-001 1.811525e-001 1.561922e-001 1.341105e-001 3.328579e-006  
 <GCONST> 2.630504e+001  
 <MIXTURE> 2 0.5  
 <MEAN> 39  
 -2.287300e+001 8.440299e-001 -3.010209e+000 2.445473e+000 7.964596e-001 1.627146e+000 1.070139e+000 -1.729109e-001 -  
 5.655214e-001 -1.511322e-001 -6.126329e-001 -2.323046e-001 1.502536e-001 -3.626747e-002 -2.633287e-002 -2.099266e-002 -  
 2.272648e-002 -2.126951e-002 -1.572300e-002 1.080966e-002 -6.597116e-003 1.060628e-002 9.410312e-003 1.435712e-003  
 5.752541e-003 -9.275221e-004 -3.092465e-003 -3.489682e-002 -7.191115e-002 -9.681918e-002 -9.970449e-002 -8.423927e-002 -  
 5.090431e-002 -2.448103e-002 -2.015325e-002 -3.494697e-002 -3.345631e-002 -2.034126e-002 2.024452e-004  
 <VARIANCE> 39  
 4.576448e+000 3.217183e+000 4.874683e+000 4.912004e+000 6.159999e+000 7.889365e+000 9.313688e+000 1.379009e+001  
 1.202154e+001 9.293769e+000 9.381868e+000 7.398954e+000 1.674059e-002 8.200133e-002 1.771310e-001 2.971694e-001  
 4.469141e-001 5.628617e-001 6.396690e-001 7.327306e-001 7.714055e-001 8.278959e-001 8.482979e-001 7.271792e-001 6.123903e-001  
 2.995073e-005 1.488839e-002 3.219865e-002 5.410300e-002 8.164508e-002 1.053251e-001 1.238816e-001 1.505603e-001  
 1.612691e-001 1.716750e-001 1.630337e-001 1.475711e-001 1.232586e-001 5.644965e-006  
 <GCONST> 3.093649e+001  
 <TRANSP> 8  
 0.000000e+000 1.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000  
 0.000000e+000 9.737582e-001 2.624176e-002 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000  
 0.000000e+000 0.000000e+000 9.368320e-001 6.316803e-002 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000  
 0.000000e+000 0.000000e+000 0.000000e+000 9.116822e-001 8.831779e-002 0.000000e+000 0.000000e+000 0.000000e+000  
 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 9.612586e-001 3.874145e-002 0.000000e+000 0.000000e+000  
 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 9.612586e-001 3.874145e-002 0.000000e+000  
 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 9.612586e-001 3.874145e-002  
 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000  
 <ENDHMM>

## Priedas 2. Lifto valdymo komandų gramatikos failas

Transkripcijos esančios 1.liftokomandos.grxml faile:

```
<grammar xmlns:sapi="http://schemas.microsoft.com/Speech/2002/06/SRGSExtensions" xml:lang="en-US"
tag-format="semantics-ms/1.0" version="1.0" mode="voice" xmlns="http://www.w3.org/2001/06/grammar"
sapi:alphabet="x-microsoft-ups">
  <rule id="Rule" scope="public">
    <one-of>
      <item>
        <one-of>
          <item>hantras</item>
          <item>antras</item>
          <item>angtrass</item>
          <item>aaantras</item>
          <item>hantras</item>
          <item>hhntrras</item>
          <item>eintrass</item>
          <item>awntrras</item>
          <item>hahntrras</item>
        </one-of>
        <tag>$. _value = "Antras"</tag>
      </item>
      <item>
        <one-of>
          <item>baigtidarba</item>
          <item>baigtidarba</item>
        </one-of>
        <tag>$. _value = "Baigtidarba"</tag>
      </item>
      <item>
        <one-of>
          <item>getvirt</item>
          <item>getviirt</item>
          <item>getvirrt</item>
          <item>ketvirt</item>
        </one-of>
        <tag>$. _value = "ketvirtas"</tag>
      </item>
      <item>
        <one-of>
          <item>pent</item>
          <item>penkat</item>
          <item>peenkt</item>
          <item>peengt</item>
          <item>penkt</item>
        </one-of>
        <tag>$. _value = "Penktas"</tag>
      </item>
      <item>
        <one-of>
          <item>piirmas</item>
          <item>ppirmas</item>
          <item>piirmmas</item>
          <item>pirm</item>
        </one-of>
      </item>
    </one-of>
  </rule>
</grammar>
```



```

        </one-of>
        <tag>$. _value = "pirmas"</tag>
    </item>
    <item>
        <one-of>
        <item>shesht</item>
        <item>cheesht</item>
        <item>chesht</item>
        <item>sest</item>
        </one-of>
        <tag>$. _value = "sestas"</tag>
    </item>
    <item>
        <one-of>
        <item>suustog</item>
        <item>ssustog</item>
        <item>suustook</item>
        <item>sustok</item>
        </one-of>
        <tag>$. _value = "sustok"</tag>
    </item>
    <item>
        <one-of>
        <item>trechi</item>
        <item>aeechii</item>
        <item>treechi</item>
        <item>trecki</item>
        <item>rjechias</item>
        <item>thrichias</item>
        <item>rechias</item>
        <item>iachias</item>
        </one-of>
        <tag>$. _value = "trecias"</tag>
    </item>
    <item>
        <one-of>
        <item>vazoiug</item>
        <item>vaziok</item>
        <item>vaziuok</item>
        </one-of>
        <tag>$. _value = "vaziuok"</tag>
    </item>
</one-of>
</rule>
</grammar>

```

### **Priedas 3. Lifto komandų sąrašo failas**

Lifto valdymo komandos esančios 1.liftokomandos.doc faile:

Antras

Baigti darbą

Ketvirtas

Penktas

Pirmas

Šeštas

Sustok

Trečias

Važiuok

#### Priedas 4. Daugiafunkcinės sistemos valdymo balsu programos kodas

Daugiafunkcinės valdymo balsu sistemos pilnas programos kodas:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Speech.Recognition;
using System.IO.Ports;
using System.Speech.Recognition.SrgsGrammar;
using InTheHand.Net.Ports;
using InTheHand.Net.Bluetooth;
using InTheHand.Net;
using InTheHand.Windows.Forms;
using System.Threading;
using LibUsbDotNet;
using LibUsbDotNet.Main;

namespace demo
{
    public partial class Form1 : Form
    {
        bool desine = false;
        bool apacia = false;
        bool kaire = false;
        bool virsus = false;
        bool failosiuntimas = false;
        bool Narsykle = false;
        bool COM = false;
        bool Beviele = false;
        bool PCprograma = false;
        bool USB = false;
        bool Grotuvas = false;
        bool operanarsykle = false;
        bool chromenarsykle = false;
        bool explorernarsykle = false;
        int Selected;
        public static UsbDevice MyUsbDevice;
        public static UsbDeviceFinder MyUsbFinder = new UsbDeviceFinder(1234, 1);

        InTheHand.Net.BluetoothAddress[] address_array = new BluetoothAddress[1000];
        private Thread thrSend;
```

```

private SpeechRecognitionEngine recognizer = new SpeechRecognitionEngine();
public Form1()

{
    InitializeComponent();
    Grammar gramatika = new Grammar("0.1Cikloveikimas.grxml", "Rule");
    recognizer.LoadGrammar(gramatika);
    richTextBox1.Clear();
    richTextBox1.LoadFile("0.1Cikloveikimas.rtf");
    recognizer.SetInputToDefaultAudioDevice();
    recognizer.SpeechRecognized += Recognizer_SpeechRecognized;
    recognizer.RecognizeAsync(RecognizeMode.Multiple);
    this.Focus();
}

private void Recognizer_SpeechRecognized(object sender, SpeechRecognizedEventArgs e)
//balso fiksavimas/atpažinimas
{
    string word = e.Result.Text.ToString();
    textBox1.Text = word;
    string komanda = e.Result.Semantics.Value.ToString();
    textBox2.Text = komanda;
    SendKeys.Send(komanda);
    if (komanda == "Apacia")
    {
        y += 100;
        SetDesktopLocation(x, y);
    }
    else if (komanda == "Kaire")
    {
        x -= 100;
        SetDesktopLocation(x, y);
    }
    else if (komanda == "Virsu") {
        y -= 100;
        SetDesktopLocation(x, y);
    }
    else if (komanda == "Desine") {
        x += 100;
        SetDesktopLocation(x, y);
        SendKeys.Send("{TAB}");
    }
}

private void sendfile() // Failo siuntimas
{
    int index = Selected;
    InTheHand.Net.BluetoothAddress address = this.address_array[index];
    System.Uri uri = new Uri("obex://" + address.ToString() + "/" + "bandymas1.txt");
}

```

```

ObexWebRequest request = new ObexWebRequest(uri);
//SendKeys.Send(word);
request.ReadFile("d:\\bandymas1.txt");
ObexWebResponse response = (ObexWebResponse)request.GetResponse();
response.Close();
}

private void button12_Click(object sender, EventArgs e) // 1.COM sąsajos mygtukas
{
    COM = true;
    button1.Visible = true;
    button2.Visible = true;
    button3.Visible = true;
    button4.Visible = false;
    button5.Visible = false;
    button6.Visible = false;
    button7.Visible = false;
    button8.Visible = false;
    button9.Visible = false;
    button10.Visible = false;
    button13.Visible = false;
    button14.Visible = false;
    button15.Visible = false;
    button16.Visible = false;
    Įrenginiai.Visible = false;
    richTextBox1.Visible = true;
    richTextBox1.Clear();
    textBox1.Clear();
    textBox2.Clear();
    Įrenginiai.Items.Clear();
    textBox3.Text = "Komandų užkrovimui pasirinkite `Komandos` mygtuką";
    SerialPort SP = new SerialPort("COM1", 9600);
    SP.Write(komanda);
}

private void button4_Click(object sender, EventArgs e) // 2.USB sąsajos mygtukas
{
    USB = true;
    button1.Visible = true;
    button2.Visible = true;
    button3.Visible = true;
    button5.Visible = false;
    button6.Visible = false;
    button7.Visible = false;
    button8.Visible = false;
    button9.Visible = false;
    button10.Visible = false;
    button12.Visible = false;
    button13.Visible = false;
    button14.Visible = false;
    button15.Visible = false;
    button16.Visible = false;
}

```

```

        button23.Visible = true;
        Įrenginiai.Visible = false;
        richTextBox1.Visible = true;
        richTextBox1.Clear();
        textBox1.Clear();
        textBox2.Clear();
        Įrenginiai.Items.Clear();
        textBox3.Text = "Komandų užkrovimui pasirinkite `Komandos` mygtuką";
    }
    private void button23_Click(object sender, EventArgs e) //2.USB siuntimas
    {

        ErrorCode ec = ErrorCode.None;
        try
        {
            MyUsbDevice = UsbDevice.OpenUsbDevice(MyUsbFinder);
            if (MyUsbDevice == null) throw new Exception("Device Not Found.");
            IUsbDevice wholeUsbDevice = MyUsbDevice as IUsbDevice;
            if (!ReferenceEquals(wholeUsbDevice, null))
            {
                wholeUsbDevice.SetConfiguration(1);
                wholeUsbDevice.ClaimInterface(0);
            }
            UsbEndpointReader reader = MyUsbDevice.OpenEndpointReader(ReadEndpointID.Ep01);

            byte[] readBuffer = new byte[1024];
            while (ec == ErrorCode.None)
            {
                int bytesRead;
                ec = reader.Read(readBuffer, 5000, out bytesRead);
                if (bytesRead == 0) throw new Exception(string.Format("{0}:No more bytes!", ec));
                Console.WriteLine("{0} bytes read", bytesRead);
                Console.Write(Encoding.Default.GetString(readBuffer, 0, bytesRead));
            }
            Console.WriteLine("\r\nDone!\r\n");
        }
        catch (Exception ex)
        {
            Console.WriteLine();
            Console.WriteLine((ec != ErrorCode.None ? ec + ":" : String.Empty) + ex.Message);
        }
        finally
        {
            if (MyUsbDevice != null)
            {
                if (MyUsbDevice.IsOpen)
                {
                    IUsbDevice wholeUsbDevice = MyUsbDevice as IUsbDevice;
                    if (!ReferenceEquals(wholeUsbDevice, null))
                    {

```

```

        wholeUsbDevice.ReleaseInterface(0);
    }

    MyUsbDevice.Close();
}
MyUsbDevice = null;
UsbDevice.Exit();
}
Console.ReadKey();
}

}

private void button5_Click(object sender, EventArgs e) // 3.Interneto naršyklės mygtukas
{
    Narsykle = true;
    button1.Visible = true;
    button2.Visible = true;
    button3.Visible = true;
    button4.Visible = false;
    button6.Visible = false;
    button7.Visible = false;
    button8.Visible = false;
    button9.Visible = false;
    button10.Visible = false;
    button12.Visible = false;
    button13.Visible = false;
    button14.Visible = true;
    button15.Visible = true;
    button16.Visible = true;
    Įrenginiai.Visible = false;
    richTextBox1.Visible = true;
    richTextBox1.Clear();
    textBox1.Clear();
    textBox2.Clear();
    Įrenginiai.Items.Clear();
    textBox3.Text = "Komandų užkrovimui pasirinkite `Komandos` mygtuką";
}

private void button14_Click(object sender, EventArgs e) // 3.Interneto narsykle OPERA
{
    textBox3.Text = "Darbui pradėti pasirenkamas `Start` mygtukas";
    System.Diagnostics.Process.Start("C:\\Program Files (x86)\\Opera\\launcher.exe");
    Grammar gramatika = new Grammar("3.1OperaNarsykle.grxml", "Rule");
    recognizer.LoadGrammar(gramatika);
    richTextBox1.Clear();
    richTextBox1.LoadFile("3.1OperaNarsykle.rtf");
}

private void button15_Click(object sender, EventArgs e) // 3.interneto narsykle CHROME
{
    textBox3.Text = "Darbui pradėti pasirenkamas `Start` mygtukas";
}

```

```

System.Diagnostics.Process.Start("C:\\Program Files\\Chrome\\chrome.exe");
Grammar gramatika = new Grammar("3.2ChromeNarsykle.grxml", "Rule");
recognizer.LoadGrammar(gramatika);
richTextBox1.Clear();
richTextBox1.LoadFile("3.2ChromeNarsykle.rtf");
}

private void button16_Click(object sender, EventArgs e) // 3.interneto narsykle EXPLORER
{
    textBox3.Text = "Darbui pradėti pasirenkamas `Start` mygtukas";
    System.Diagnostics.Process.Start("C:\\Program Files\\Internet Explorer\\iexplore.exe");
    Grammar gramatika = new Grammar("3.3ExplorerNarsykle.grxml", "Rule");
    recognizer.LoadGrammar(gramatika);
    richTextBox1.Clear();
    richTextBox1.LoadFile("3.3ExplorerNarsykle.rtf");
}

private void button6_Click(object sender, EventArgs e) // 4.PC programos mygtukas
{
    PCprograma = true;
    button1.Visible = true;
    button2.Visible = true;
    button3.Visible = true;
    button4.Visible = false;
    button5.Visible = false;
    button7.Visible = false;
    button8.Visible = false;
    button9.Visible = false;
    button10.Visible = false;
    button12.Visible = false;
    button13.Visible = false;
    button14.Visible = false;
    button15.Visible = false;
    button16.Visible = false;
    Įrenginiai.Visible = false;
    richTextBox1.Visible = true;
    richTextBox1.Clear();
    textBox1.Clear();
    textBox2.Clear();
    Įrenginiai.Items.Clear();
    textBox3.Text = "Komandų užkrovimui pasirinkite `Komandos` mygtuką";
}

private void button13_Click(object sender, EventArgs e) // 5.Grotuvo mygtukas
{
    Grotuvas = true;
    button1.Visible = true;
    button2.Visible = true;
    button3.Visible = true;
    button4.Visible = false;
    button5.Visible = false;
}

```



```

        button6.Visible = false;
        button7.Visible = false;
        button8.Visible = false;
        button9.Visible = false;
        button10.Visible = false;
        button12.Visible = false;
        button14.Visible = false;
        button15.Visible = false;
        button16.Visible = false;
        Įrenginiai.Visible = false;
        richTextBox1.Visible = true;
        richTextBox1.Clear();
        textBox1.Clear();
        textBox2.Clear();
        Įrenginiai.Items.Clear();
        System.Diagnostics.Process.Start("C:\\Program Files (x86)\\Windows Media
Player\\wmplayer.exe");
        textBox3.Text = "Komandų užkrovimui pasirinkite `Komandos` mygtuką";
    }

private void button10_Click(object sender, EventArgs e) // 6.Bluetooth sąsajos mygtukas
{
    Beviele = true;
    button1.Visible = true;
    button2.Visible = false;
    button3.Visible = false;
    button4.Visible = false;
    button5.Visible = false;
    button6.Visible = false;
    button7.Visible = true;
    button8.Visible = true;
    button9.Visible = true;
    button10.Visible = true;
    button12.Visible = false;
    button13.Visible = false;
    button14.Visible = false;
    button15.Visible = false;
    button16.Visible = false;
    button22.Visible = true;
    Įrenginiai.Visible = true;
    richTextBox1.Visible = true;
    textBox4.Visible = true;
    label3.Visible = true;
    richTextBox1.Clear();
    textBox1.Clear();
    textBox2.Clear();
    Įrenginiai.Items.Clear();
    textBox3.Text = "Komandų užkrovimui pasirinkite `Komandos` mygtuką";
}

private void button8_Click_1(object sender, EventArgs e) // 6.Bluetooth> Ieškoti mygtukas
{

```

```

textBox3.Text = "Iš sąrašo pasirinkite įrenginį ir spauskite `Adresas` mygtuką";
InTheHand.Net.Sockets.BluetoothClient bc = new InTheHand.Net.Sockets.BluetoothClient();
InTheHand.Net.Sockets.BluetoothDeviceInfo[] array = bc.DiscoverDevices();
for (int i = 0; i < array.Length; i++)
{
    this.address_array[i] = array[i].DeviceAddress;
    this.Įrenginiai.Items.Add(array[i].DeviceName);
}
}

private void button9_Click_1(object sender, EventArgs e) // 6.Bluetooth> Adresas mygtukas
{

    textBox3.Text = "Failo siuntimui pasirinkite `Siusti` mygtuką";
    if (this.Įrenginiai.SelectedIndex == -1)
    {
        MessageBox.Show("Pažymėkite įrenginį.");
        return;
    }
    int index = this.Įrenginiai.SelectedIndex;
    string mac = this.address_array[index].ToString();

    textBox4.Text = "MAC : " + mac.ToString();
}
private void button22_Click(object sender, EventArgs e) //6.Bluetooth Failo pasirinkimo
mygtukas
{
    FolderBrowserDialog folderDlg = new FolderBrowserDialog();
    folderDlg.ShowNewFolderButton = true;
    // Show the FolderBrowserDialog.
    DialogResult result = folderDlg.ShowDialog();
    if (result == DialogResult.OK)
    {
        textBox1.Text = folderDlg.SelectedPath;
        Environment.SpecialFolder root = folderDlg.RootFolder;
    }
}

private void button7_Click_1(object sender, EventArgs e) // 6.Bluetooth> Siųsti mygtukas
{
    textBox3.Text = "Norint grįžti atgal pasirinkite `Atgal` mygtuką";
    if (this.Įrenginiai.SelectedIndex == -1)
    {
        MessageBox.Show("Please select a device.");
        return;
    }
    Selected = this.Įrenginiai.SelectedIndex;
    this.thrSend = new Thread(new ThreadStart(sendfile));
    this.thrSend.Start();
}

private void button11_Click(object sender, EventArgs e) // Atgal mygtukas

```

```

    {
        if (panel1.Visible == true && panel2.Visible == true && button5.Visible == true &&
            button4.Visible == true && button6.Visible == true && button10.Visible == true &&
            button12.Visible == true &&
            button13.Visible == true)
        {
            textBox2.Clear();
            textBox3.Clear();
            textBox3.Text = "Pasirinkite atitinkamas komandas pozicionuoti programa, baigus
            poziciuonuoti patvirtinkite";
            panel3.Visible = true;
            Grammar gramatika = new Grammar("0.1Cikloveikimas.grxml", "Rule");
            recognizer.LoadGrammar(gramatika);
            richTextBox1.Clear();
            richTextBox1.LoadFile("0.1Cikloveikimas.rtf");
        }

        else
        {
            COM = false;
            Narsykle = false;
            USB = false;
            PCprograma = false;
            Beviele = false;
            Grotuvas = false;
            button1.Visible = false;
            button2.Visible = false;
            button3.Visible = false;
            button4.Visible = true;
            button5.Visible = true;
            button6.Visible = true;
            button7.Visible = false;
            button8.Visible = false;
            button9.Visible = false;
            button10.Visible = true;
            button12.Visible = true;
            button13.Visible = true;
            button14.Visible = false;
            button15.Visible = false;
            button16.Visible = false;
            button22.Visible = false;
            button23.Visible = false;
            Irenginiai.Visible = false;
            richTextBox1.Visible = true;
            textBox4.Visible = false;
            label3.Visible = false;
            recognizer.RecognizeAsyncStop();
            richTextBox1.Clear();

            recognizer.UnloadAllGrammars();
            Irenginiai.Items.Clear();
            Grammar gramatika = new Grammar("0.2SasajuPasirinkimas.grxml", "Rule");

```

```

        recognizer.LoadGrammar(gramatika);
        richTextBox1.Clear();
        richTextBox1.LoadFile("0.2SasajuPasirinkimas.rtf");
        textBox1.Clear();
        textBox2.Clear();
        textBox3.Text = "Pasirinkite norimą valdyti įrenginį";
    }
}

private void button1_Click(object sender, EventArgs e) // Komandu uzkrovimo mygtukas
{
    textBox3.Text = "Darbui pradėti pasirenkamas `Start` mygtukas";
    if (COM == true)
    {
        Grammar gramatika = new Grammar("1.liftokomandos.grxml", "Rule");
        recognizer.LoadGrammar(gramatika);
        richTextBox1.Clear();
        richTextBox1.LoadFile("1.liftokomandos.rtf");
    }
    else if (USB == true)
    {
        Grammar gramatika = new Grammar("2.USBkomandos.grxml", "Rule");
        recognizer.LoadGrammar(gramatika);
        richTextBox1.Clear();
        richTextBox1.LoadFile("2.USBkomandos.rtf");
    }
    else if (PCprograma == true)
    {
        Grammar gramatika = new Grammar("4.PCkomandos.grxml", "Rule");
        recognizer.LoadGrammar(gramatika);
        richTextBox1.Clear();
        richTextBox1.LoadFile("4.PCkomandos.rtf");
    }
    else if (Grotuvas == true)
    {
        Grammar gramatika = new Grammar("5.Grotuvokomandos.grxml", "Rule");
        recognizer.LoadGrammar(gramatika);
        richTextBox1.Clear();
        richTextBox1.LoadFile("5.Grotuvokomandos.rtf");
    }

    else if (Beviele == true)
    {
        textBox3.Text = "Įrenginių ieškojimui pasirinkite `Paieška` mygtuką";
        Grammar gramatika = new Grammar("6.Bevieleskomandos.grxml", "Rule");
        recognizer.LoadGrammar(gramatika);
        richTextBox1.Clear();
        richTextBox1.LoadFile("6.Bevieleskomandos.rtf");
    }
}
}

```

```

private void button2_Click(object sender, EventArgs e) // Start mygtukas
{
    textBox3.Text = "Ištarkite iš sąrašo pasirinktą komandą";
    recognizer.SetInputToDefaultAudioDevice();
    recognizer.SpeechRecognized += Recognizer_SpeechRecognized;
    recognizer.RecognizeAsync(RecognizeMode.Multiple);
}

private void button3_Click(object sender, EventArgs e) // Stop mygtukas
{
    textBox3.Text = "Komandų užkrovimui pasirinkite `Komandos` mygtuką";
    recognizer.RecognizeAsyncStop();
    richTextBox1.Clear();
    textBox1.Clear();
    recognizer.UnloadAllGrammars();
}

int y = 0;
int x = 0;

private void button17_Click_1(object sender, EventArgs e) //Pozicija> į Apačią
{
    apacia = true;
    if (textBox2.Text == "Apacia" || apacia == true)
        y += 100;
    SetDesktopLocation(x, y);
}

private void button18_Click(object sender, EventArgs e) // Pozicija> į Viršų
{
    labas:
    y -= 100;
    SetDesktopLocation(x, y);
}

private void button19_Click(object sender, EventArgs e) // Pozicija> į Kairę
{
    labas:
    kaire = true;
    if (textBox2.Text == "Kaire" || kaire == true)
        x -= 100;
    SetDesktopLocation(x, y);
}

private void button20_Click(object sender, EventArgs e) // Pozicija> į Dešinę
{
    desine = true;
    if (textBox2.Text == "Desine" || desine == true)
        x += 100;
    SetDesktopLocation(x, y);
}

private void button21_Click(object sender, EventArgs e) //Pozicija> Patvirtinti
{

```

```
richTextBox1.Clear();
textBox1.Clear();
textBox2.Clear();
Grammar gramatika = new Grammar("0.2SasajuPasirinkimas.grxml", "Rule");
recognizer.LoadGrammar(gramatika);
richTextBox1.Clear();
richTextBox1.LoadFile("0.2SasajuPasirinkimas.rtf");
panel1.Visible = true;
panel2.Visible = true;
panel3.Visible = false;
textBox3.Text = "Pasirinkite norimą valdyti įrenginį";
    }
}
}
```

## Priedas 5. PMM modelių atpažinimo tikslumas keičiant būsenas

**5.1 lentelė.** Visų diktorių lifto komandų atpažinimo tikslumo rezultatai keičiant vidinių būsenų skaičių

Diktorius	Pridėtos papildomos vidinės būsenos "+0",%	Pridėtos papildomos vidinės būsenos "+2",%	Pridėtos papildomos vidinės būsenos "+4",%	Pridėtos papildomos vidinės būsenos "+6",%	Pridėtos papildomos vidinės būsenos "+8",%	Pridėtos papildomos vidinės būsenos "+10",%
FBEASTU	61,11	53,89	68,89	62,78	72,22	68,33
FGINBAR	81,11	78,33	92,78	91,11	89,44	93,89
FGINPAS	90	78,33	97,78	80,56	86,67	86,67
FIVEVAL	89,44	68,89	100	85,56	90	88,89
MDAIPOC	82,22	72,22	81,11	66,11	77,78	77,78
MDARJEG	79,44	86,11	87,22	90	89,44	98,33
MDZEPAL	81,11	86,67	86,67	73,89	83,33	83,89
MIGNMAK	93,33	62,78	98,89	91,11	96,11	97,78
MKASRAT	75	67,44	99,44	67,78	86,33	75,56
MMARRAZ	91,11	70	91,11	85,56	90,56	91,67
MREMRAC	93,33	80,56	97,78	89,44	95,56	96,67
Vidurkis	83,38	73,20	91,06	80,35	87,04	87,22

## Priedas 6. PMM modelių atpažinimo tikslumas, keičiant busenas ir Gauso mišinius

**6.1 lentelė.** Visų diktorių lifto komandų atpažinimo tikslumo rezultatai, kai vidinių būsenų skaičius nepakite, o įtarpemi Gauso mišiniai kinta nuo 0 iki 8.

Diktorius	Įterpti Gauso mišiniai "+0",%	Įterpti Gauso mišiniai "+2",%	Įterpti Gauso mišiniai "+4",%	Įterpti Gauso mišiniai "+6",%	Įterpti Gauso mišiniai "+8",%
FBEASTU	61,11	52,78	59,44	43,89	74,44
FGINBAR	81,11	69,44	89,44	75,56	92,78
FGINPAS	90	76,67	93,89	83,89	95,56
FIVEVAL	89,44	81,11	95,56	78,89	92,22
MDAIPOC	82,22	80,56	90	69,44	88,33
MDARJEG	79,44	90,56	92,78	88,33	94,44
MDZEPAL	81,11	68,89	85	61,67	93,33
MIGNMAK	93,33	86,67	99,44	78,89	100
MKASRAT	75	60	92,22	87,78	95,56
MMARRAZ	91,11	82,78	88,33	74,44	99,44
MREMRAC	93,33	94,44	97,78	86,67	99,44
Vidurkis	83,38	76,72	89,44	75,40	93,23

**6.2 lentelė.** Visų diktorių lifto komandų atpažinimo tikslumo rezultatai, kai pridamos dvi papildomos vidinės būsenos, o įtarpemi Gauso mišiniai kinta nuo 0 iki 8.

Diktorius	Įterpti Gauso mišiniai "+0",%	Įterpti Gauso mišiniai "+2",%	Įterpti Gauso mišiniai "+4",%	Įterpti Gauso mišiniai "+6",%	Įterpti Gauso mišiniai "+8",%
FBEASTU	53,89	56,11	78,78	55,00	71,11
FGINBAR	78,33	76,67	98,33	68,33	94,44
FGINPAS	78,33	82,22	100,00	82,72	98,89
FIVEVAL	68,89	91,11	99,44	88,33	98,33
MDAIPOC	72,22	78,89	92,22	73,89	99,44
MDARJEG	86,11	97,78	98,89	100,00	99,44
MDZEPAL	86,67	62,78	88,89	73,33	94,44
MIGNMAK	62,78	89,44	98,89	93,89	100,00
MKASRAT	67,44	59,44	90,00	98,89	98,89
MMARRAZ	70	87,22	100,00	80,00	98,33
MREMRAC	80,56	91,67	100,00	92,22	100,00
Vidurkis	73,20	79,39	95,04	82,42	95,76



**6.3 lentelė.** Visų diktorių lifto komandų atpažinimo tikslumo rezultatai, kai pridedamos keturios papildomos vidinės būsenos , o įtarpemi Gauso mišiniai kinta nuo 0 iki 8.

Diktorius	Įterpti Gauso mišiniai "+0",%	Įterpti Gauso mišiniai "+2",%	Įterpti Gauso mišiniai "+4",%	Įterpti Gauso mišiniai "+6",%	Įterpti Gauso mišiniai "+8",%
FBEASTU	68,89	60,00	72,78	60,00	77,22
FGINBAR	92,78	76,11	90,56	72,78	98,33
FGINPAS	97,78	83,33	99,44	84,44	100,00
FIVEVAL	100	96,11	100,00	88,33	99,44
MDAIPOC	81,11	75,00	91,67	76,11	97,22
MDARJEG	87,22	93,33	94,44	97,78	98,89
MDZEPAL	86,67	63,33	95,56	77,78	93,89
MIGNMAK	98,89	87,22	98,33	92,22	100,00
MKASRAT	99,44	60,00	96,67	94,44	99,44
MMARRAZ	91,11	82,78	96,11	78,89	99,44
MREMRAC	97,78	88,89	98,89	85,56	99,44
Vidurkis	91,06	78,74	94,04	82,58	96,66

**6.4 lentelė.** Visų diktorių lifto komandų atpažinimo tikslumo rezultatai, kai pridedamos šešios papildomos vidinės būsenos , o įtarpemi Gauso mišiniai kinta nuo 0 iki 8.

Diktorius	Įterpti Gauso mišiniai "+0",%	Įterpti Gauso mišiniai "+2",%	Įterpti Gauso mišiniai "+4",%	Įterpti Gauso mišiniai "+6",%	Įterpti Gauso mišiniai "+8",%
FBEASTU	62,78	56,67	77,78	61,11	84,44
FGINBAR	91,11	74,44	92,22	72,22	95,00
FGINPAS	80,56	81,67	99,44	82,22	99,44
FIVEVAL	85,56	96,11	100,00	87,78	98,33
MDAIPOC	66,11	72,22	81,67	77,78	94,44
MDARJEG	90	96,11	100,00	98,33	100,00
MDZEPAL	73,89	66,11	86,11	77,78	96,11
MIGNMAK	91,11	82,22	96,11	85,56	100,00
MKASRAT	67,78	63,89	97,78	97,78	99,44
MMARRAZ	85,56	84,44	82,78	81,11	100,00
MREMRAC	89,44	87,78	95,56	79,44	99,44
Vidurkis	80,35	78,33	91,77	81,92	96,97

## Priedas 7. Hibridinio atpažintuvo klasifikatorių atpažinimo tikslumai

7.1 lentelė. Atpažinimo tikslumas tiriant skirtingus klasifikatorius, visų įrašų atveju ir 11 atskirų aplankų atveju

Klasifikatorius	10-kartų kryžminio patikrinimo atpažinimo tikslumas,%	11 aplankų atpažinimo tikslumas, %											11-kartų kryžminio patikrinimo atpažinimo tikslumo vidurkis,%
		1	2	3	4	5	6	7	8	9	10	11	
BN	79,92	61,90	87,50	95,00	100,00	78,57	92,31	79,17	37,50	57,14	71,43	16,67	70,65
NB	82,95	65,08	100,00	100,00	100,00	90,47	92,31	83,33	62,50	65,71	71,43	25,00	77,80
Logisite	90,53	80,95	100,00	95,00	100,00	85,71	87,18	79,17	75,00	80,00	100,00	100,00	89,36
MP	90,91	80,95	62,50	100,00	100,00	90,48	87,18	70,93	75,00	77,14	100,00	50,00	81,29
SGD	92,42	34,92	62,50	100,00	83,33	76,19	94,87	50,00	75,00	62,86	100,00	100,00	76,33
SL	91,67	65,08	100,00	100,00	100,00	83,33	87,18	79,17	60,00	100,00	100,00	100,00	88,61
SMO	91,29	84,13	100,00	100,00	100,00	97,62	87,18	79,17	75,00	65,71	100,00	100,00	89,89
VP	68,18	34,92	62,50	100,00	83,33	76,19	94,87	50,00	75,00	62,85	100,00	100,00	76,33
Ibk	93,43	69,84	50,00	100,00	100,00	83,33	87,18	87,50	37,50	80,00	71,49	58,33	75,02
Kstar	93,43	69,84	100,00	100,00	83,33	90,48	87,18	70,83	37,50	62,86	71,43	58,33	75,62
AdaBoost1	84,09	58,73	62,50	100,00	83,33	85,71	87,18	58,33	75,00	65,71	65,71	100,00	76,56
Bagging	89,02	65,08	100,00	100,00	100,00	80,95	87,18	75,00	37,50	62,85	100,00	100,00	82,60
MCC	91,67	80,95	100,00	95,00	100,00	85,71	87,18	79,17	75,00	80,00	100,00	100,00	89,36
Stacking	68,18	34,92	62,50	100,00	83,33	76,19	94,87	50,00	75,00	62,86	100,00	100,00	76,33
Vote	68,18	34,92	62,50	100,00	83,33	76,19	94,87	50,00	75,00	62,86	100,00	100,00	76,33
DT	85,23	60,32	75,00	100,00	100,00	95,24	87,18	66,67	37,50	65,71	100,00	100,00	80,69
RIPPER	89,34	61,90	100,00	100,00	83,33	95,24	87,18	75,00	37,50	62,86	100,00	100,00	82,09
OneR	72,73	46,03	62,50	100,00	83,33	52,38	87,18	58,33	75,00	48,58	100,00	25,00	67,12
PART	92,80	65,08	100,00	45,00	83,33	95,34	87,18	70,83	37,50	65,71	71,43	75,00	72,40
ZeroR	68,18	34,92	62,50	100,00	83,33	76,19	94,87	50,00	75,00	62,86	100,00	100,00	76,33
C4.5	90,15	50,79	87,50	100,00	83,33	95,24	87,18	70,93	75,00	82,18	85,71	75,00	81,17
LMT	90,91	50,79	100,00	65,00	100,00	64,29	87,18	70,83	87,50	82,86	100,00	58,33	78,80
RF	93,56	61,90	100,00	100,00	83,33	90,48	90,18	79,18	37,50	57,14	71,43	66,67	76,16
RT	91,67	46,03	100,00	95,00	83,33	97,62	87,18	79,17	37,50	82,86	100,00	83,33	81,09
REPTree	85,98	60,31	100,00	85,00	83,33	80,95	92,31	83,33	37,50	82,86	100,00	25,00	75,51

Visi tirti klasifikatoriai panaudoti iš šių direktorijų: bayes.BayesNet; bayes.NaiveBayes; functions.Logistic; functions.MultilayerPerceptron; functions.SGD; functions.SimpleLogistic; functions.SMO; functions.VotedPerceptron; lazy.IBk; lazy.KStar; meta.AdaBoostM1; meta.Bagging; meta.MultiClassClassifier; meta.Stacking; meta.Vote; rules.DecisionTable; rules.JRip; rules.OneR; rules.PART; rules.ZeroR; trees.J48; trees.LMT; trees.RandomForest; trees.RandomTree; trees.REPTree;

## Priedas 8. Hibridinio atpažintuvo atpažinimo tikslumo požymių failas

**8.1** Apjungtų ATP\_LT ir ATP\_SP atpažintuvų lifto modelio valdymo balso komandų atpažinimo tikslumo faile „visi.arff“ požymiai:

```
@RELATION LT_SP
@ATTRIBUTE LT_prob NUMERIC
@ATTRIBUTE SP_prob NUMERIC
@ATTRIBUTE gender {M,F}
@ATTRIBUTE lt_a NUMERIC
@ATTRIBUTE lt_b NUMERIC
@ATTRIBUTE lt_ch NUMERIC
@ATTRIBUTE lt_d NUMERIC
@ATTRIBUTE lt_e NUMERIC
@ATTRIBUTE lt_g NUMERIC
@ATTRIBUTE lt_i NUMERIC
@ATTRIBUTE lt_k NUMERIC
@ATTRIBUTE lt_m NUMERIC
@ATTRIBUTE lt_n NUMERIC
@ATTRIBUTE lt_o NUMERIC
@ATTRIBUTE lt_p NUMERIC
@ATTRIBUTE lt_r NUMERIC
@ATTRIBUTE lt_s NUMERIC
@ATTRIBUTE lt_sh NUMERIC
@ATTRIBUTE lt_t NUMERIC
@ATTRIBUTE lt_u NUMERIC
@ATTRIBUTE lt_v NUMERIC
@ATTRIBUTE lt_z NUMERIC
@ATTRIBUTE sp_a NUMERIC
@ATTRIBUTE sp_b NUMERIC
@ATTRIBUTE sp_ch NUMERIC
@ATTRIBUTE sp_d NUMERIC
@ATTRIBUTE sp_e NUMERIC
@ATTRIBUTE sp_g NUMERIC
@ATTRIBUTE sp_i NUMERIC
@ATTRIBUTE sp_k NUMERIC
@ATTRIBUTE sp_m NUMERIC
@ATTRIBUTE sp_n NUMERIC
@ATTRIBUTE sp_o NUMERIC
@ATTRIBUTE sp_p NUMERIC
@ATTRIBUTE sp_r NUMERIC
@ATTRIBUTE sp_s NUMERIC
@ATTRIBUTE sp_sh NUMERIC
@ATTRIBUTE sp_t NUMERIC
@ATTRIBUTE sp_u NUMERIC
@ATTRIBUTE sp_v NUMERIC
@ATTRIBUTE sp_z NUMERIC
@ATTRIBUTE class {LT,SP}

@DATA
-66.73,347,F,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,33.3,0,0,0,0,0,0,0,0,0,16.7,0,0,16.7,16.7,0,16.7,0,0,0,SP
-62.89,701,F,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,33.3,0,0,0,0,0,0,0,0,0,16.7,0,0,16.7,16.7,0,16.7,0,0,0,SP
-66.76,679,F,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,33.3,0,0,0,0,0,0,0,0,0,16.7,0,0,16.7,16.7,0,16.7,0,0,0,SP
-79.46,569,F,11.1,0,0,0,11.1,0,0,11.1,1,1,0,0,0,0,11.1,1,1,0,22.2,0,11.1,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT
-70.31,590,F,11.1,0,0,0,11.1,0,0,11.1,1,1,0,0,0,0,11.1,1,1,0,22.2,0,11.1,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT
```







-81.59,599,M,27.3,18.2,0,9,1,0,9,1,18.2,0,0,0,0,9,1,0,0,9,1,0,0,0,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,LT  
-74.99,621,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,SP  
-67.65,610,M,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,14.3,0,14.3,0,14.3,0,0,0,0,LT  
-67.59,629,M,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,14.3,0,14.3,0,14.3,0,0,0,0,LT  
-69.44,637,M,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,16.7,0,0,0,16.7,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT  
-70.65,631,M,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,16.7,0,0,0,16.7,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT  
-70.91,637,M,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,14.3,0,14.3,0,14.3,0,0,0,0,14.3,14.3,0,14.3,0,0,0,LT  
-64.15,672,M,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,14.3,0,14.3,0,14.3,0,0,0,0,14.3,14.3,0,14.3,0,0,0,LT  
-66.31,605,M,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,16.7,0,0,0,16.7,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT  
-68.44,619,M,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,16.7,0,0,0,16.7,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT  
-69.12,605,M,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,16.7,0,0,0,16.7,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT  
-69.56,655,M,33.3,0,0,0,0,0,0,0,0,16.7,0,0,16.7,0,16.7,0,16.7,0,0,0,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,SP  
-71.96,619,M,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,0,0,0,0,16.7,0,0,16.7,0,0,33.3,0,16.7,16.7,0,0,LT  
-71.06,587,M,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,0,0,0,0,16.7,0,0,16.7,0,0,33.3,0,16.7,16.7,0,0,LT  
-69.11,624,M,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,0,0,0,0,16.7,0,0,16.7,0,0,33.3,0,16.7,16.7,0,0,LT  
-68.17,607,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,SP  
-73.94,614,M,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,0,0,0,0,16.7,0,0,16.7,0,0,33.3,0,16.7,16.7,0,0,LT  
-70.6,647,M,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,0,0,0,0,16.7,0,0,16.7,0,0,33.3,0,16.7,16.7,0,0,LT  
-68.45,608,M,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,0,0,0,0,16.7,0,0,16.7,0,0,33.3,0,16.7,16.7,0,0,LT  
-70.88,636,M,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,0,0,0,0,16.7,0,0,16.7,0,0,33.3,0,16.7,16.7,0,0,LT  
-62.01,658,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,0,14.3,14.3,0,14.3,0,0,0,SP  
-64.35,664,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,0,14.3,14.3,0,14.3,0,0,0,SP  
-62.81,643,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,0,14.3,14.3,0,14.3,0,0,0,SP  
-60.89,620,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,0,14.3,14.3,0,14.3,0,0,0,SP  
-66.47,641,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,0,14.3,14.3,0,14.3,0,0,0,SP  
-60.4,672,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,0,14.3,14.3,0,14.3,0,0,0,SP  
-62.61,667,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,0,14.3,14.3,0,14.3,0,0,0,SP  
-61.09,635,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,0,14.3,14.3,0,14.3,0,0,0,SP  
-61.8,641,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,0,14.3,14.3,0,14.3,0,0,0,SP  
-60.01,667,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,0,14.3,14.3,0,14.3,0,0,0,SP  
-76.97,405,M,27.3,18.2,0,9,1,0,9,1,18.2,0,0,0,0,9,1,0,0,9,1,0,0,0,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,LT  
-67.91,663,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT  
-66.09,694,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT  
-78.11,566,M,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,0,0,0,0,16.7,0,0,16.7,0,0,33.3,0,16.7,16.7,0,0,LT  
-77.8,570,M,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,0,0,0,0,16.7,0,0,16.7,0,0,33.3,0,16.7,16.7,0,0,LT  
-56.55,541,M,14.3,0,0,0,0,14.3,14.3,0,0,14.3,0,0,0,0,14.3,14.3,14.3,0,0,0,0,0,0,16.7,0,0,16.7,0,0,33.3,0,16.7,16.7,0,0,LT  
-55.7,555,M,14.3,0,0,0,0,14.3,14.3,0,0,14.3,0,0,0,0,14.3,14.3,14.3,0,0,0,0,0,0,16.7,0,0,16.7,0,0,33.3,0,16.7,16.7,0,0,LT  
-67.39,612,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT  
-69.3,617,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT  
-68.58,686,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT  
-65.42,612,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT  
-69.91,613,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT  
-66.64,688,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT  
-71.45,628,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT  
-68.88,651,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT  
-67.95,671,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT  
-61.3,559,M,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,16.7,0,0,0,16.7,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT  
-54.9,568,M,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,16.7,0,0,0,16.7,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT  
-69.31,504,M,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,0,0,0,0,16.7,0,0,16.7,0,0,33.3,0,16.7,16.7,0,0,LT

## 8.2 Apjungtų ATP\_LT ir ATP\_SP atpažintuvų lifto modelio valdymo balso komandų atpažinimo tikslumo faile „ltest.arff“ požymiai:

```
@RELATION LT_SP
@ATTRIBUTE LT_prob NUMERIC
@ATTRIBUTE SP_prob NUMERIC
@ATTRIBUTE gender {M,F}
@ATTRIBUTE lt_a NUMERIC
@ATTRIBUTE lt_b NUMERIC
@ATTRIBUTE lt_ch NUMERIC
@ATTRIBUTE lt_d NUMERIC
@ATTRIBUTE lt_e NUMERIC
@ATTRIBUTE lt_g NUMERIC
@ATTRIBUTE lt_i NUMERIC
@ATTRIBUTE lt_k NUMERIC
@ATTRIBUTE lt_m NUMERIC
@ATTRIBUTE lt_n NUMERIC
@ATTRIBUTE lt_o NUMERIC
@ATTRIBUTE lt_p NUMERIC
@ATTRIBUTE lt_r NUMERIC
@ATTRIBUTE lt_s NUMERIC
```





-60.71,477,F,14.3,0,0,0,0,0,14.3,14.3,0,0,14.3,0,0,0,0,0,14.3,14.3,14.3,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT  
-72.81,509,F,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,14.3,0,0,0,0,0,14.3,14.3,0,0,14.3,0,0,0,0,14.3,14.3,SP  
-59.22,444,F,14.3,0,0,0,0,0,14.3,14.3,0,0,14.3,0,0,0,0,14.3,14.3,14.3,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT  
-61.29,479,F,14.3,0,0,0,0,0,14.3,14.3,0,0,14.3,0,0,0,0,14.3,14.3,14.3,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT  
-67.16,426,F,14.3,0,0,0,0,0,14.3,14.3,0,0,14.3,0,0,0,0,14.3,14.3,14.3,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT  
-71.6,478,F,14.3,0,0,0,0,0,14.3,14.3,0,0,14.3,0,0,0,0,14.3,14.3,14.3,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT  
-68.93,447,F,14.3,0,0,0,0,0,14.3,14.3,0,0,14.3,0,0,0,0,14.3,14.3,14.3,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT  
-67.94,475,F,14.3,0,0,0,0,0,14.3,14.3,0,0,14.3,0,0,0,0,14.3,14.3,14.3,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,14.3,14.3,0,14.3,0,0,0,LT  
-62.36,465,F,14.3,0,0,0,0,0,14.3,14.3,0,0,14.3,0,0,0,0,14.3,14.3,14.3,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT  
-60.01,449,F,14.3,0,0,0,0,0,14.3,14.3,0,0,14.3,0,0,0,0,14.3,14.3,14.3,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,14.3,14.3,0,14.3,0,0,0,LT  
-62.34,389,F,14.3,0,0,0,0,0,14.3,14.3,0,0,14.3,0,0,0,0,14.3,14.3,14.3,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,14.3,14.3,0,14.3,0,0,0,LT

### 8.3 Apjungtų ATP\_LT ir ATP\_SP atpažintuvų lifto modelio valdymo balso komandų atpažinimo tikslumo faile „ltrain.arff“ požymiai:

```
@RELATION LT_SP
@ATTRIBUTE LT_prob NUMERIC
@ATTRIBUTE SP_prob NUMERIC
@ATTRIBUTE gender {M,F}
@ATTRIBUTE lt_a NUMERIC
@ATTRIBUTE lt_b NUMERIC
@ATTRIBUTE lt_ch NUMERIC
@ATTRIBUTE lt_d NUMERIC
@ATTRIBUTE lt_e NUMERIC
@ATTRIBUTE lt_g NUMERIC
@ATTRIBUTE lt_j NUMERIC
@ATTRIBUTE lt_k NUMERIC
@ATTRIBUTE lt_m NUMERIC
@ATTRIBUTE lt_n NUMERIC
@ATTRIBUTE lt_o NUMERIC
@ATTRIBUTE lt_p NUMERIC
@ATTRIBUTE lt_r NUMERIC
@ATTRIBUTE lt_s NUMERIC
@ATTRIBUTE lt_sh NUMERIC
@ATTRIBUTE lt_t NUMERIC
@ATTRIBUTE lt_u NUMERIC
@ATTRIBUTE lt_v NUMERIC
@ATTRIBUTE lt_z NUMERIC
@ATTRIBUTE sp_a NUMERIC
@ATTRIBUTE sp_b NUMERIC
@ATTRIBUTE sp_ch NUMERIC
@ATTRIBUTE sp_d NUMERIC
@ATTRIBUTE sp_e NUMERIC
@ATTRIBUTE sp_g NUMERIC
@ATTRIBUTE sp_i NUMERIC
@ATTRIBUTE sp_k NUMERIC
@ATTRIBUTE sp_m NUMERIC
@ATTRIBUTE sp_n NUMERIC
@ATTRIBUTE sp_o NUMERIC
@ATTRIBUTE sp_p NUMERIC
@ATTRIBUTE sp_r NUMERIC
@ATTRIBUTE sp_s NUMERIC
@ATTRIBUTE sp_sh NUMERIC
@ATTRIBUTE sp_t NUMERIC
@ATTRIBUTE sp_u NUMERIC
@ATTRIBUTE sp_v NUMERIC
@ATTRIBUTE sp_z NUMERIC
@ATTRIBUTE class {LT,SP}
```

```
@DATA
-67.97,711,F,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,33.3,0,0,0,0,0,0,0,16.7,0,0,16.7,16.7,0,16.7,0,0,0,SP
-68.86,713,F,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,33.3,0,0,0,0,0,0,0,16.7,0,0,16.7,16.7,0,16.7,0,0,0,SP
-65.83,673,F,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,14.3,14.3,0,14.3,0,0,0,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,SP
-70.34,613,F,14.3,0,0,0,14.3,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,0,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT
-72.51,631,F,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,0,0,0,0,0,0,16.7,0,0,16.7,0,0,33.3,0,16.7,16.7,0,0,LT
-77.77,603,F,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,0,0,0,0,0,0,16.7,0,0,16.7,0,0,33.3,0,16.7,16.7,0,0,LT
-72.24,590,F,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,0,0,0,0,0,0,16.7,0,0,16.7,0,0,33.3,0,16.7,16.7,0,0,LT
-74.26,566,F,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,0,0,0,0,0,0,16.7,0,0,16.7,0,0,33.3,0,16.7,16.7,0,0,LT
-58.9,666,F,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT
-62.64,674,F,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT
-60.67,656,F,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,LT
-54.24,563,F,16.7,0,0,0,0,16.7,0,16.7,0,0,16.7,16.7,0,0,0,0,0,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,LT
-48.64,558,F,16.7,0,0,0,0,0,16.7,0,16.7,0,0,16.7,16.7,16.7,0,0,0,0,0,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,LT
-55.37,573,F,16.7,0,0,0,0,0,16.7,0,16.7,0,0,16.7,16.7,16.7,0,0,0,0,0,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,LT
-52.05,589,F,16.7,0,0,0,0,0,16.7,0,16.7,0,0,16.7,16.7,16.7,0,0,0,0,0,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,LT
-55.44,481,F,16.7,0,0,0,0,0,16.7,0,16.7,0,0,16.7,16.7,16.7,0,0,0,0,0,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,LT
-50.8,530,F,16.7,0,0,0,0,0,16.7,0,16.7,0,0,16.7,16.7,16.7,0,0,0,0,0,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,LT
```





-60.89,620,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,0,14.3,14.3,0,14.3,0,0,0,0,SP  
-66.47,641,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,0,14.3,14.3,0,14.3,0,0,0,0,SP  
-60.4,672,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,0,14.3,14.3,0,14.3,0,0,0,0,SP  
-62.61,667,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,0,14.3,14.3,0,14.3,0,0,0,0,SP  
-61.09,635,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,0,14.3,14.3,0,14.3,0,0,0,0,SP  
-61.8,641,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,0,14.3,14.3,0,14.3,0,0,0,0,SP  
-60.01,667,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,0,14.3,14.3,0,14.3,0,0,0,0,SP  
-66.77,405,M,27.3,18.2,0,9.1,0,9.1,18.2,0,0,0,0,0,9.1,0,9.1,0,0,0,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,0,LT  
-73.91,663,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,LT  
-66.09,694,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,LT  
-78.11,566,M,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,0,0,0,0,0,0,0,0,16.7,0,0,16.7,0,0,33.3,0,16.7,16.7,0,0,0,LT  
-77.8,570,M,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,0,0,0,0,0,0,0,0,16.7,0,0,16.7,0,0,33.3,0,16.7,16.7,0,0,0,LT  
-56.55,541,M,14.3,0,0,0,0,14.3,14.3,0,0,14.3,0,0,0,0,0,14.3,14.3,14.3,0,0,0,0,0,0,0,16.7,0,0,16.7,0,0,33.3,0,16.7,16.7,0,0,0,LT  
-55.7,555,M,14.3,0,0,0,0,14.3,14.3,0,0,14.3,0,0,0,0,0,14.3,14.3,14.3,0,0,0,0,0,0,0,16.7,0,0,16.7,0,0,33.3,0,16.7,16.7,0,0,0,LT  
-67.39,612,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,LT  
-69.3,617,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,LT  
-68.58,686,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,LT  
-65.42,612,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,LT  
-69.91,613,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,LT  
-66.64,688,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,LT  
-71.45,628,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,LT  
-68.88,651,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,LT  
-67.95,671,M,11.1,0,0,0,11.1,0,11.1,11.1,0,0,0,0,11.1,11.1,0,22.2,0,11.1,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,LT  
-61.3,559,M,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,LT  
-54.9,568,M,14.3,0,0,0,14.3,0,0,14.3,0,14.3,0,14.3,0,14.3,0,14.3,0,0,0,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,LT  
-69.31,504,M,16.7,0,0,0,16.7,0,0,0,0,0,0,0,16.7,33.3,16.7,0,0,0,0,0,0,0,0,0,0,0,0,16.7,0,0,16.7,0,0,33.3,0,16.7,16.7,0,0,0,LT

## Priedas 9. Auto-WEKA rezultatas

Geriausio rasto klasifikatoriaus ir optimizuotų parametrų Auto-WEKOS rezultatų failas:

==== Run information ====

Scheme: weka.classifiers.meta.Auto-WEKAClassifier -seed 123 -timeLimit 15 -memLimit 1024 -nBestConfigs 1 -  
metric errorRate -parallelRuns 1

Relation: LT\_SP

Instances: 264

Attributes: 42

LT\_prob  
SP\_prob  
gender  
lt\_a  
lt\_b  
lt\_ch  
lt\_d  
lt\_e  
lt\_g  
lt\_i  
lt\_k  
lt\_m  
lt\_n  
lt\_o  
lt\_p  
lt\_r  
lt\_s  
lt\_sh  
lt\_t  
lt\_u  
lt\_v  
lt\_z  
sp\_a  
sp\_b  
sp\_ch  
sp\_d  
sp\_e  
sp\_g  
sp\_i  
sp\_k  
sp\_m  
sp\_n  
sp\_o  
sp\_p  
sp\_r  
sp\_s  
sp\_sh  
sp\_t  
sp\_u  
sp\_v  
sp\_z  
class

Test mode: 10-fold cross-validation

==== Classifier model (full training set) ====

best classifier: weka.classifiers.trees.RandomForest

arguments: [-I, 10, -K, 0, -depth, 0]

attribute search: null

```

attribute search arguments: []
attribute evaluation: null
attribute evaluation arguments: []
metric: errorRate
estimated errorRate: 0.007575757575757576
training time on evaluation dataset: 0.042 seconds

```

You can use the chosen classifier in your own code as follows:

```

Classifier classifier = AbstractClassifier.forName("weka.classifiers.trees.RandomForest", new String[]{"-I", "10", "-K",
"0", "-depth", "0"});
classifier.buildClassifier(instances);

```

```

Correctly Classified Instances      260      98.4848 %
Incorrectly Classified Instances    4        1.5152 %
Kappa statistic                    0.9646
Mean absolute error                0.0572
Root mean squared error            0.141
Relative absolute error            13.1675 %
Root relative squared error        30.2764 %
Total Number of Instances         264

```

=== Confusion Matrix ===

```

a b <-- classified as
180 0 | a = LT
4 80 | b = SP

```

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1,000	0,048	0,978	1,000	0,989	0,965	0,999	0,999	LT
	0,952	0,000	1,000	0,952	0,976	0,965	0,999	0,997	SP
Weighted Avg.	0,985	0,032	0,985	0,985	0,985	0,965	0,999	0,998	

For better performance, try giving Auto-WEKA more time.  
Tried 317 configurations; to get good results reliably you may need to allow for trying thousands of configurations.

Time taken to build model: 798.9 seconds