



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

GRAFINIŲ SISTEMŲ KATEDRA

Artūras Krivenkis

**ŽMOGAUS BALSO ATPAŽINIMO SISTEMŲ, PAGRĮSTŲ
PASLĖPTAIS MARKOVO MODELIAIS IR DIRBTINIŲ NEURONŲ
TINKLAIS, TYRIMAS**

**RESEARCH OF HUMAN'S VOICE RECOGNITION SYSTEMS BASED
ON HIDDEN MARKOV MODELS AND ARTIFICIAL NEURAL
NETWORKS**

Baigiamasis magistro darbas

Informacinių technologijų studijų programa, valstybinis kodas 6211BX016

Multimedijos informacinių sistemų specializacija

Informatikos inžinerijos studijų kryptis

Vilnius, 2020

Vilniaus Gedimino technikos universitetas
Fundamentinių mokslų fakultetas
Grafinių sistemų katedra

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

Antrosios pakopos studijų **Informacinių technologijų** programos magistro baigiamasis darbas 4

Pavadinimas **Žmogaus balso atpažinimo sistemų, pagrįstų paslėptais Markovo modeliais ir dirbtinių neuronų tinklais, tyrimas**

Autorius **Artūras Krivenkis**

Vadovas **Juozas Kamarauskas**

Kalba: lietuvių

Anotacija

Magistro darbo metu yra analizuojami žmogaus balso atpažinimo metodai, problemos ir jų sprendimo būdai bei atliekamas jau egzistuojančių sistemų įvertinimas. Pagal literatūros analizę iš apžvelgtų algoritmų išrinkti PMM ir DNT balso identifikavimo metodai, realizuoti Python programavimo kalba. Sukurti du žmogaus balso atpažinimo programos Spider aplinkoje, kurie naudoja pasirinktus klasifikatorius ir MSKK požymius. Programoms atliekamas eksperimentinis įvertinimas ir palyginimas tarpusavyje, siekiant sužinoti, kuris iš metodų yra geriausias.

Darbą sudaro vienuolika dalių: paveikslų, lentelių, schemų ir grafikų sąrašai, terminų žodynas, įvadas, literatūros analizė, žmogaus balso atpažinimo sistemų analizė, kuriamų žmogaus balso atpažinimo sistemų realizavimo eiga, eksperimentinis tyrimas, išvados ir literatūros sąrašas. Darbo apimtis 79 p. teksto be sąrašų ir terminų žodyno, 28 paveikslai, 11 lentelių, 2 grafikai, 11 schemos ir 55 literatūros šaltiniai.

Prasminiai žodžiai: atpažinimas, identifikavimas, žmogus, balsas, sistema, metodas, klasifikatorius, PMM, DNT, paliginimas, požymiai, MSKK, Python, Spyder.

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

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

Master Degree Studies **Information Technologies** study programme Master Graduation Thesis 4

Title **Research of Human's Voice Recognition Systems Based on Hidden Markov Models and Artificial Neural Networks**

Author **Artūras Krivenkis**

Academic supervisor **Juozas Kamarauskas**

Thesis language: Lithuanian

Annotation

This Master thesis deals with voice recognition methods, problems, their solutions and the evaluation of existing systems. According to literature, from reviewed algorithms HMM and DNN voice identification systems were selected and implemented with a Python programming language. Two voice recognition programs were created in Spyder environment, which uses selected classifiers and MFCC features. Experimental evaluation of the programs and assimilation with each other was performed to find out which method is better.

Thesis consists of 11 chapters: lists of pictures, tables, charts and graphs, vocabulary, introduction, analysis of literature, analyze of human voice recognition systems, realization process of human voice recognition systems, experimental research, conclusions and references. Work coverage – 79 p. text without lists and vocabulary, 28 images, 11 tables, 2 graphs, 11 schemes and 55 references.

Keywords: recognition, identification, human, voice, system, method, classifier, HMM, DNN, comparison, features, MFCC, Python, Spyder.

TURINYS

PAVEIKSLŲ SĄRAŠAS.....	6
LENTELIŲ SĄRAŠAS.....	7
GRAFIKŲ IR SCHEMŲ SĄRAŠAS.....	8
TERMINŲ ŽODYNAS.....	9
ĮVADAS.....	10
1. LITERATŪROS ANALIZĖ.....	12
1.1.Garsas.....	12
1.2.Garso atpažinimo sistemos.....	13
1.2.1. Garso atpažinimo sistemos raida.....	14
1.2.2. Garso atpažinimo sistemos realizavimo metodai.....	16
1.2.3. Signalų požymių nustatymo būdai.....	20
1.2.4. Signalų atpažinimo klasifikatoriai.....	24
1.3. Garso atpažinimo sistemos realizavimo problemos.....	31
1.4.Skyriaus apibendrinimas.....	32
2. ŽMOGAUS BALSO ATPAŽINIMO SISTEMŲ ANALIZĖ.....	33
2.1. Įvesties signalo modeliavimas ir kūrimas.....	33
2.2. Pirminis signalo filtravimas ir dalijimas į kadrus.....	35
2.3. Signalų segmentavimas.....	39
2.4. Signalų požymių išskyrimas.....	41
2.5. Signalų modelio sukūrimas ir lyginimo metodai.....	44
2.5.1. Paslėptasis Markovo modelis.....	45
2.5.2. Giliojo mokymosi metodas.....	48
2.6.Skyriaus apibendrinimas.....	53
3. KURIAMŲ ŽMOGAUS BALSO ATPAŽINIMO SISTEMŲ REALIZAVIMO EIGA	54
3.1.MSKK ir PMM atpažinimo sistemos įgyvendinimas.....	55
3.2.MSKK ir giliojo mokymosi atpažinimo sistemos kūrimas.....	63
3.3.Skyriaus apibendrinimas.....	68

4. EKSPERIMENTINIS TYRIMAS.....	69
4.1. Programų testavimas ir palyginimas.....	69
4.2. Skyriaus apibendrinimas.....	84
IŠVADOS.....	86
LITERATŪROS SĄRAŠAS.....	89

PAVEIKSLŲ SĄRAŠAS

1.1 pav. Standartinis kairės-dešinės <i>PMM</i> modelis.....	27
1.2 pav. (a) Vieno sluoksnio perceptronas, (b) daugiasluoksnis perceptronas...	30
2.1 pav. Garso signalai prieš ir po pradinės filtracijos (Loveimi, 2013).....	35
2.2 pav. Stačiakampio lango funkcija (Poularikas, 1999).....	36
2.3 pav. Hanningo lango funkcija (Poularikas, 1999).....	37
2.4 pav. Hemingo lango funkcija (Poularikas, 1999).....	37
2.5 pav. Gauso lango funkcija (Poularikas, 1999).....	38
2.6 pav. Trikampio lango funkcija (Poularikas, 1999).....	38
2.7 pav. Blackmano lango funkcija (Poularikas, 1999).....	39
2.8 pav. Melų skalė (Athira, 2017).....	42
2.9 pav. Normalizuotas trikampio filtras (Singh, 2014).....	43
2.10 pav. MSKK koeficientai po DKT skaičiavimo (Singh, 2014).....	44
2.11 pav. Ergodinis (a) ir grandininis (b) <i>PMM</i> modelis.....	45
2.12 pav. Tiesioginio leidimo rekursija (Schickhofer, 2017).....	46
2.13 pav. Atgalinio leidimo rekursija (Schickhofer, 2017).....	48
3.1 pav. <i>MSKK</i> su <i>PMM</i> atpažinimo programos veikimo principas.....	55
3.2 pav. <i>MSKK</i> su <i>PMM</i> atpažinimo programos <i>UML</i> diagrama.....	56
3.3 pav. <i>PMM</i> programos <i>_future_</i> modelio kodas.....	56
3.4 pav. <i>PMM</i> programos reikalingų bibliotekų kodas.....	56
3.5 pav. <i>PMM</i> programos pradžios kodas.....	58
3.6 pav. Melų spektrogramos pavyzdys.....	61
3.7 pav. Garso signalo amplitudės ir laiko grafiko pavyzdys.....	61
3.8 pav. <i>PMM</i> programos <i>train_GaussianPMM()</i> funkcijos iškvietimo kodas..	61
3.9 pav. <i>PMM</i> programos išvesties rezultato pavyzdys.....	63
3.10 pav. <i>MSKK</i> su <i>DNT</i> atpažinimo programos veikimo principas.....	64
3.11 pav. <i>MSKK</i> su <i>DNT</i> atpažinimo programos <i>UML</i> diagrama.....	65
3.12 pav. <i>DNT</i> programos <i>tensorflow</i> bibliotekos prijungimo kodas.....	65
3.13 pav. <i>DNT</i> programos išvesties rezultato pavyzdys.....	67

LENTELIŲ SĄRAŠAS

1.1 lentelė. <i>DNT</i> pranašumai ir trūkumai.....	30
6.1 lentelė. Atpažinimo rezultatas ir laikas, didinant neuronų skaičių ir apmokymo žingsnių skaičių.....	70
6.2 lentelė. Atpažinimo rezultatas ir laikas, didinant neuronų skaičių ir apmokymo signalų skaičių.....	71
6.3 lentelė. Atpažinimo rezultatas ir laikas, didinant apmokymo žingsnių skaičių ir apmokymo signalų skaičių.....	73
6.4 lentelė. Atpažinimo rezultatas ir laikas, didinant neuronų skaičių ir mažinant apmokymo žingsnių skaičių.....	74
6.5 lentelė. Atpažinimo rezultatas ir laikas, didinant neuronų skaičių ir mažinant apmokymo signalų skaičių.....	75
6.6 lentelė. Atpažinimo rezultatas ir laikas, didinant apmokymo žingsnių skaičių ir mažinant apmokymo signalų skaičių.....	76
6.7 lentelė. Atpažinimo tikimybė ir laikas, didinant neuronų skaičių.....	78
6.8 lentelė. Atpažinimo rezultatas ir laikas, didinant apmokymo žingsnių skaičių.....	79
6.9 lentelė. Atpažinimo rezultatas ir laikas, didinant apmokymo signalų skaičių.....	80
6.10 lentelė. Bendras gautų rezultatų apibendrinimas kiekvienam klasifikatoriui.....	83

GRAFIKŲ IR SCHEMŲ SĄRAŠAS

6.1 grafikas. Atpažinimo sistemų laiko vidurkiai kiekvienu testavimo atveju...	82
6.2 grafikas. Programų atpažinimo tikslumas kiekvienu testavimo atveju.....	83
1.1 schema. Garso signalo atpažinimo proceso schema.....	13
1.2 schema. Garso atpažinimo sistemos raida.....	14
1.3 schema. Šabloninis atpažinimo metodas.....	18
1.4 schema. <i>TPKK</i> skaičiavimo eiga.....	21
1.5 schema. <i>TSP</i> skaičiavimo eiga.....	22
1.6 schema. <i>GMM</i> bruožų skaičiavimo eiga.....	26
3.1 schema. <i>PMM</i> programos <i>buildDataSet()</i> funkcijos diagrama.....	59
3.2 schema. <i>PMM</i> programos <i>isskyrti_mfcc()</i> funkcijos diagrama.....	60
3.3 schema. <i>PMM</i> programos <i>train_GaussianPMM()</i> funkcijos diagrama.....	62
3.4 schema. <i>DNT</i> programos <i>buildDataSet()</i> funkcijos diagrama.....	66
3.5 schema. <i>DNT</i> programos <i>train_DNT()</i> funkcijos diagrama.....	67

TERMINŲ ŽODYNAS

1. **Programinė įranga** (angl. *Software*) – informacijos apdorojimo procedūrų ir taisyklių visuma arba tos visumos dalis. Kompiuterio vykdomų instrukcijų seka, skirta tam tikriems veiksams atlikti.
2. **UML** (angl. *Unified Modeling Language diagram*) – modeliavimo ir specifikacijų kūrimo kalba, skirta objektiškai orientuotų programų dokumentams specifiuoti, atvaizduoti ir konstruoti.
3. **Modelis** (angl. *Model*) – tai konkrečią tikslinę paskirtį (analizė, konstravimas, valdymas) turintis supaprastintas kokio nors originalo (realios sistemos) analogas, atspindintis to originalo struktūrą arba elgseną.
4. **Gilusis mokymasis** (angl. *Deep Learning*) – platesnės mašinų mokymosi metodų šeimos dalis, remiantis mokymosi duomenų vaizdavimu, o ne specifiniais užduoties algoritmais. Jos architektūros, pvz., gilus neuroninius tinklas, gilus pasitikėjimo tinklas ir pasikartojantis neuroninis tinklas, pritaikomos laukuose, įskaitant kompiuterinę viziją, kalbos atpažinimą ir pan.
5. **Fonema** (angl. *Phoneme*) – minimalus skiriamasis kalbos vienetas.
6. **Perceptronas** (angl. *Perceptron*) – tai yra paprasčiausias tiesioginio sklidimo neuroninis tinklų tipas, naudojamas „mokymasis su mokytoju“ algoritme. Jis susideda iš vieno sluoksnio d neuronų su n įvesčių.
7. **Programėlių programavimo sąsaja** (angl. *Application Programming Interface, API*) – tai sąsaja, kurią suteikia kompiuterinė sistema, biblioteka ar programa tam, kad programuotojas per kitą programą galėtų pasiekti jos funkcionalumą ar apsiektų su ja duomenimis.
8. **Unikodas** (angl. *Unicode*) – standartas, apibrėžiantis beveik visų kalbų abėcėlių bei papildomų simbolių kodavimą kompiuteriuose.

IVADAS

Mūsų laikais technologijos sparčiai tobulėja ir duomenų saugumas tampa vis labiau prioritetiniu dalyku. Žmonės sugalvoja ir bando realizuoti skirtingus metodus, kaip lengvai ir užtikrintai saugoti savo turtą arba kontroliuoti prietaisus, kad jie galėtų veikti tik taip, kaip nori jų savininkas.

Vienas iš tokių biometrijos metodų yra garso arba balso atpažinimas. Šios technikos panaudojimo idėja išpopuliarėjo septintojo ir aštuntojo dešimtmečio fantastiniuose filmuose. Vienas iš tokių filmų – „STAR WARS“, kuriame robotai „R2D2“ ir „C3PO“ galėjo lengvai atpažinti savininko bei kitų komandos narių balsus. Tokie filmai atskleidė naujas technologijų tobulinimo ir jų pritaikymo kasdieniame gyvenime galimybes.

Dabar garso atpažinimo technologijos leidžia daugeliui prietaisų reaguoti į žmogaus balsą ir pagal tai teikti naudingas paslaugas. Be to, kiekvieno žmogaus balsas yra unikalus ir garso atpažinimo technologijas galima naudoti telefono, automobilio arba net namo apsaugos sistemose. Šios technologijos iki šiol tobulėja ir turi skirtingus realizavimo algoritmus bei metodus.

Vieni iš populiariausių klasikinių ir iki šiol daugumoje garso atpažinimo sistemų taikomų metodų yra: PMM (*paslėptasis Markovo modelis*), GMM (*Gauso mišinių modelis*) ir naujas, neseniai patobulintas, DNT (*giliojo mokymosi metodas*). Šie trys garso atpažinimo metodai naudojami kaip klasifikatoriai po garso apdorojimo su kažkuriuo iš garso požymių nustatymo algoritmų, tokių kaip MSKK (*melų skalės keistro koeficientai*), TPK (*tiesinės prognozės kodavimas*) ir daugybė kitų. Tačiau iki šiol nėra visai tikslių garso atpažinimo sistemų. Yra problemų arba metodų trūkumų, neleidžiančių sukurti sistemos, kuri galėtų visada atpažinti balso savininką ir pateikti tinkamą rezultatą. Garso atpažinimo sistemai nustatyti savininko balsą gali trukdyti daugybė veiksnių: aplinka, kurioje pasakomas žodis, žodžių panašumas, skirtingos žmonių kalbos, savininko balso pasikeitimas dėl įvairių priežasčių. Visi šie dalykai lemia sistemos veikimo kokybę ir teisingo rezultato gavimą. Dėl to yra svarbu sukurti sistemą, kuri galėtų kuo greičiau ir tiksliau nustatyti balso savininką.

Išanalizavus kilusią žmogaus balso atpažinimo problemą, jos paklausą rinkoje bei sistemų įvairovę, galima padaryti išvadą, kad identifikavimo sistemų palyginimas, nustatant geriausiai veikiančią metodą, yra aktuali tema.

Baigiamojo magistro darbo tikslas – išanalizuoti žmogaus balso atpažinimo metodus ir algoritmus, pritaikyti juos programose bei palyginti dvi populiariausias sistemas. Šiam tikslui pasiekti suformuluoti 4 pagrindiniai **darbo uždaviniai**:

1. Atlikus literatūros analizę nustatyti dažnai naudojamus garso atpažinimo algoritmus, kylančias problemas ir jų sprendimo būdus.
2. Plačiau išanalizuoti pasirinktų identifikavimo sistemų kūrimo eigą ir veikimo principus.
3. Sukurti garso atpažinimo programas, naudojant atrinktus klasifikavimo metodus.
4. Atlikti sukurtų programų testavimą, jų palyginimą ir aptarti gautus rezultatus.

1. LITERATŪROS ANALIZĖ

Analizuojant literatūrą, nagrinėjama esama situacija, susijusi su atpažinimo sistemomis ir garsu kaip duomenų šaltiniu. Šiame skyriuje yra aprašoma atpažinimo sistemos istorija ir sistemą sudarančios dalys. Plačiau yra analizuojamos skirtingos garso atpažinimo sistemos ir jų veikimo principai. Pagal pateiktą informaciją iš garso atpažinimo sistemos kūrimo proceso, jų problemų ir jų sprendimo metodų, bandoma atrinkti ir įgyvendinti automatizuotas žmogaus balso savininko identifikavimo sistemas.

1.1. Garsas

Garsas turi daugybę fizikinių ir psichologinių apibūdinimų. Fizikos mokslas garą apibrėžia kaip elastingas bangas, kurios sklinda dujose, skysčiuose arba net kietose medžiagose. Psichologijoje tai yra žmogaus smegenų girdimų bangų suvokimas. Tačiau programuojant garsas išreiškiamas kaip duomenų rinkinys, kurį galima apdoroti, lyginti ir gauti pageidaujamą rezultatą (Corvin, 2009; Station Apparatus, 1969).

Garso signalas dažniausiai kuriamas iš šaltinio, vibruojančio atitinkamo dažnio diapazone. Norint apdoroti garso signalą, jis iš garso bangų paverčiamas į skaitmeninį signalą specialiais prietaisais. Garso bangos yra tolydinio laiko ir tolydinės amplitudės analoginis signalas, o skaitmeninis signalas yra diskretinio laiko ir diskretinės amplitudės, kurie naudojami garsui apdoroti. Dirbant su signalu dažniausiai jis dalijamas į vienodo atstumo kadrus ir su ryškiausiais matomais akustiniais elementais (Corvin, 2009; Station Apparatus, 1969):

1. Garsumas – šis elementas išreiškia garso signalo amplitudės pasikeitimą laiko periodu arba jo intensyvumą.
2. Pikas – tai yra signalo vibracijų sparta arba garso signalo bangos maksimumas ir minimumas (De Cheveigne, 2005) .
3. Tembras – tai yra garso bangos forma, atspindinti signalo turinį.

Plačiau kalbant apie garso signalo elementus, jie gali būti suskirstyti į dvi kategorijas – laiko ir dažnio (Yen, 2009).

Laiko garso signalo savybės yra vidutinė energija, nulių kirtimų skaičius ir signalo / triukšmo santykis. Vidutinė garso energija rodo signalo garsumą per tam tikrą laiko intervalą. Rasti energiją galima padalijus vidutinę signalo kvadratinę vertę iš bendro atskaitų skaičiaus. Nulių kirtimų skaičius yra dažnis, kiek kartų signalas kerta laiko ašį. Dažniausiai jis yra

naudojamas pagrindiniam dažniui įvertinti arba neišreikštų garsų pradžiai, pabaigai sužinoti. Signalų / triukšmo santykis yra santykis tarp intervalo, kuriame absoliuti amplitudė yra mažesnė už tam tikrą ribą, ir bendro signalo ilgio (Yen, 2009).

Dažnio garso signalo elementai yra spektras, pralaidumas, harmonika, pikas ir tembras. Spektras garso signalu parodo, kaip kinta amplitudė kintant dažniui ir kaip pasiskirstoma signalo energija visame dažnių diapazone. Pralaidumas yra skirtumas tarp didžiausio ir mažiausio dažnio visame signalo diapazone. Harmonika yra sudėtinės bangos dažnio dalis, kuri yra neatsiejama pagrindinio dažnio kartotinė. Pikas ir tembras, kaip buvo minėta pirmiau, yra pagrindiniai signalo elementai, kurie dažniausiai naudojami apdorojant garso signalą. Pavyzdžiui, tembras naudojamas MSKK skaičiavimo algoritme kaip viena iš esminių klasifikavimo nustatymo reikšmių (Yen, 2009).

Visos garso signalo savybės leidžia atskirti jas vieną nuo kitos ir tinkamai apdoroti. Pavyzdžiui, skirtumas tarp kalbėjimo ir dainavimo signalo yra toks, kad signalo / triukšmo santykis dainuojant yra mažesnis. Be to, kalbėjimas turi siauresnį dažnių pasiskirstymą tarp žmogaus girdimo diapazono ir yra mažiau harmoningas negu dainavimas. Kiekviena iš signalo savybių padeda nustatyti paties signalo išskirtinumą iš kitų garso signalų (Yen, 2009).

Taigi, garsas yra ne tik tai, ką girdi ir suvokia žmogus, bet ir skirtingų duomenų rinkinys. Jis leidžia apdoroti, perdaryti ir atpažinti garso signalą. Esant tokiems ypatumams ir tinkamiems algoritmams, metodus galima panaudoti garsui kaip sąveikai su prietaisais ir žmogui identifikuoti. Siekdami tai įgyvendinti, programuotojai ir mokslininkai bando realizuoti skirtingas balso atpažinimo sistemas.

1.2. Garso atpažinimo sistemos

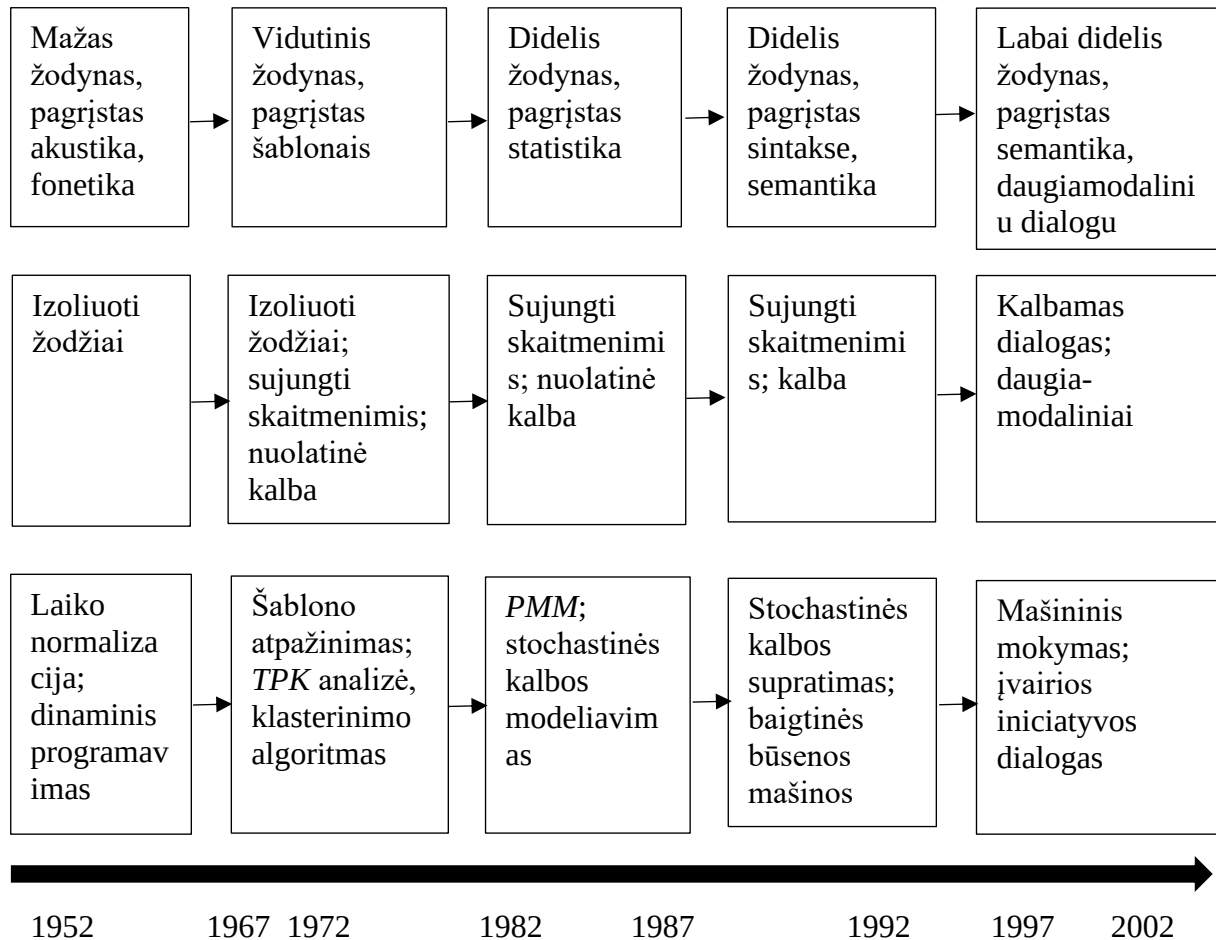
Dar praeitame amžiuje mokslininkai pradėjo domėtis, kaip atpažinti žmogų, naudojant jo balsą kaip šaltinį. Atlikus tyrimus buvo nustatyti pagrindiniai garso atpažinimo etapai, kurie pavaizduoti 1.1 schemoje. Pirmiausia garso bangos paverčiamos skaitmeniniu signalu ir išskiriami jų požymiai, kurie reikalingi tolesniam darbui. Nustačius požymius, jie klasifikuojami, priskiriant juos kategorijoms pagal tam tikrą savybę. Naudojant tuos požymius, galima lyginti garsą su požymiais, esančiais duomenų bazėje.



1.1 schema. Garso signalo atpažinimo proceso schema

1.2.1. Garso atpažinimo sistemos raida

Kalbos identifikavimo sistemos programavimas ir praktinis realizavimas prasidėjo prieš 60 metų (1.2 schema). Visą laiką žmonės bandė išspręsti pagrindines garso identifikavimo problemas: sistemos kalbos žodyno apimtį, kalbos žodžių apdorojimo kiekį, gebėjimą atpažinti skirtingų žmonių kalbą bei daugelį kitų garso atpažinimo trūkumų. Dabar dauguma sprendimų yra rasta, nors tam reikėjo daugelio žmonių pastangų ir nemažai technologijų ir algoritmų tobulinimo.



1.2 schema. Garso atpažinimo sistemos raida

Kuriant balso atpažinimo sistemas buvo nustatyta keletas problemų, kurios trukdė sukurti visiškai realizuotą sistemą. Pavyzdžiui, nepalankios aplinkos sąlygos neleisdavo kokybiškai „pagauti“ norimo garso signalo arba nustatyti tinkamų jo požymių. Pirmąjį žodžių skaitmeninį atpažinimą sukūrė Davisas, Biddulphas ir Balashekas dar 1952 metais „Bell Laboratories“. Jie kiekvienam garsiakalbiui pastatė sistemą, kuri, naudojant išmatuotus dažnius, atpažįsta izoliuotą skaitmeninį signalą (Vacher, 2010; Juang, 2004). Tais pačiais

metais daugelis kitų pasaulio mokslininkų pradėjo tobulinti garso atpažinimo technologijas (Juang, 2004) .

Olsonas ir „RCA Laboratories Belar“ sukūrė sistemą, kuri sugebėjo identifikuoti dešimt vieno žmogaus žodžių. Tą patį padarė ir „MIT Lincoln Lab“, tačiau jų sistema buvo nepriklausoma nuo pašnekovo kalbėto teksto. Japonijoje keletas tyrimų laboratorijų ir universitetų bandė sukurti balso identifikavimo programą. Pavyzdžiui, didelės įtakos nuolatinės kalbos atpažinimo sistemai turėjo Sakai ir Doshito darbas (Juang, 2004). Jie realizavo programą, kuriai nereikėdavo aiškos „nuopjovos“, kad atpažintų žodžius. Anglijoje Fry ir Denes buvo pirmieji, panaudoję statistinę sintaksę automatinėje kalbos identifikavimo programoje. Toks veikimo metodas padidino foneminių žodžių identifikavimo tikslumą (Juang, 2004) .

1960 metais buvo išspręsta laikino netolygumo problema pakartotiniuose kalbos renginiuose. Tomasa Martisn RCA laboratorijoje pasiūlė sprendimą, kur galiniai signalo taškai būtų aptikti. Sakoe ir Chiba, savo leidiniuose papildę Tomo Martino idėjas, sugalvojo dinaminio laiko skalės išlyginimo metodą, kuris tapo pagrindu automatinio kalbos atpažinimo technikoje. Nepriklausomai nuo kitų, Atal ir Itakura realizavo naują signalo požymių nustatymo metodą *TPK*. Ši technologija žymiai supaprastino kalbos atpažinimo sistemas, nes galėjo lengviau įvertinti balso trakto atsaką iš šnekos bangų (Juang, 2004).

Po 1980 metų buvo sugalvotas naujas garso atpažinimo klasifikatorius *PMM*. Šis naujas klasifikatorius dėl beveik tinkamų rezultatų gavimo tapo iki šiol dažniausiai naudojamu algoritmu garso identifikavimo sistemoje. *PMM* pagrįstas tuo, kad jis sukuria modelius kiekvienam žodžiui, kombinuojant juos visame sakinyje. Kuo geriau bus išmokyta ši technika, tuo geresnis rezultatas bus gaunamas. Len Baumas naudodavo *PMM* kaip modeliavimo ir įvertinimo klasifikatorių, kuris veikdavo pagal Markovo grandinės tikimybių funkcijas. Jie yra susiję su tikimybių pasiskirstymo rinkiniu ir pačia Markovo grandine. Be to, paslėptasis Markovo modelis gali spręsti nefiksuotas sekos funkcijas ir keisti jas pagal Markovo grandinę. Tačiau tai nebuvo galutinė *PMM* metodo realizacija. 1980 metais „Bell Laboratories“ techniką praplėtė mišinių tankiu. Toks patobulinimas žymiai pagerino atpažinimo tikslumą ir suteikė galimybę naudoti sistemą jau dideliems kalbos žodynams identifikuoti (Juang, 2004) .

Be *PMM*, dar 1950 metais buvo pristatyta dirbtinių neuroninių tinklų panaudojimo idėja. Tačiau jos įgyvendinimas praktikoje buvo realizuotas tik 1980 metais. Dėl laikinųjų variacijų problemų, pradinė neuroninių tinklų forma buvo prasta ir netikdavo sudėtingai kalbai atpažinti. Jie buvo naudojami tik paprastiems tikslams, pavyzdžiui, porai žodžių identifikuoti.

Siekiant pagerinti situaciją su neuroniniais tinklais, buvo bandoma apjungti juos ir *PMM* struktūrą. Iki šiol technologijos šioje srityje yra tobulinamos ir naujausia jos versija tampa vis populiarese už kitų panašių technikų (Juang, 2004) .

DNT sistema arba tiesioginio sklidimo neuroninis tinklas žymiai patobulėjo 2010 metais. Kalbos atpažinimo programos tyrėjai panaudojo netiesioginius paslėptus blokus ir didelius išvesties sluoksnius. Pagrindinis šio metodo tikslas yra panaikinti rankų darbo funkciją ir panaudoti neapdorotos spektrogramos savybes. Neseniai buvo įrodyta, kad tokia atpažinimo technika pateikia gerų kalbos identifikavimo rezultatų (Yu, 2010).

Vadinasi, žmogaus balso atpažinimo sistemos tobulinimas buvo ir yra vystomas iki šiol, siekiant rasti geriausią techniką, kuri galėtų turėti daugiau galimybių, veikti greičiau ir be jokių klaidų identifikuoti žmogų. Dauguma mokslininkų ir tyrėjų per 60 metų įdėjo didžiulį indėlį į tai, kad ateityje žmonės galėtų tokias sistemas naudoti kasdieniame savo gyvenime. Kalbos identifikavimo sistemos realizacijos galimybių yra daug, nes yra ne vienas metodas ar prieiga, kaip galima tai padaryti. Kiekvienam šiam būdui yra savų algoritmų, leidžiančių apdoroti garso signalą.

1.2.2. Garso atpažinimo sistemos realizavimo metodai

Kaip buvo minėta, yra daugybė metodų, kaip identifikuoti žmogų, naudojant specialius algoritmus ir technikas. Be to, galima paskirstyti kalbos atpažinimo sistemą skirtingais būdais. Pavyzdžiui, kaip technika identifikuoja žmogaus kalbą, galima nustatyti pagal tris tipus:

1. **Izoliuoti žodžiai.** Tai yra pati pirmoji sugalvota kalbos atpažinimo technika, kuri buvo realizuota dar 1952 metais. Tokio tipo reikalavimai yra gana griežti ir turi daug trūkumų. Kalbantis žmogus turi pasakyti kiekvieną žodį atskirai ir laukti tam tikrą laiką, kol galės pasakyti kitą žodį. Per tą „žodžių pauzės“ laiką dažniausiai ir vykdomas garso apdorojimas. Sistemos realizavimas yra lengvas, nes aiškiai išstartų žodžių ribos yra paprastai nustatomos (Bhabad, 2013; Arora, 2012).
2. **Sujungti žodžiai.** Šis atpažinimo tipas turi daug panašumų su izoliuotų žodžių tipu, bet žmogus gali padaryti minimalią pauzę tarp jo pasakytų žodžių (Bhabad, 2013; Arora, 2012).
3. **Nuolatinė kalba.** Paskutinis ir pats populiariausias metodas leidžia žmogui praktiškai natūraliai kalbėti, be jokių pauzių. Tokia technika naudoja specialius algoritmus arba metodus tam, kad nustatytų žodžio ribas ir dėl to sunkiausiai

kuriama, palyginti su visais kitais tipais. Vienas iš pagrindinių tokio tipo trūkumų yra tai, kad jis neturi žodžių ribų informacijos, dėl to kalbos greitis daro didelę įtaką tinkamam rezultatui gauti (Bhabad, 2013; Arora, 2012; Jelinek, 1976).

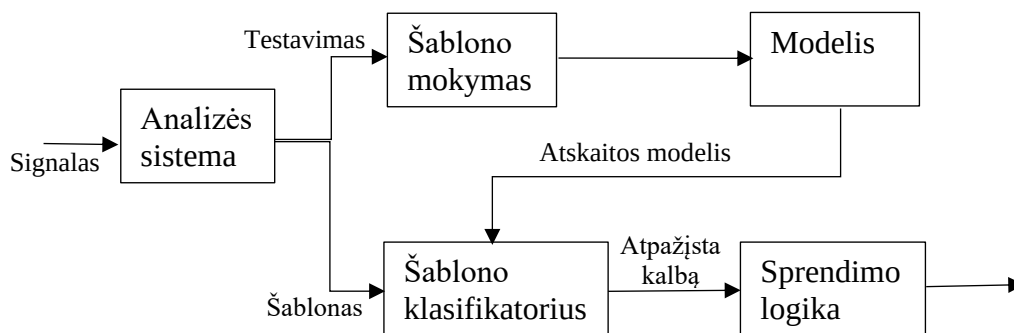
Kiekvienas iš šių tipų turi savo veikimo principus, pranašumus ir trūkumus. Be atpažinimo tipo, garso identifikavimo sistemą galima skirstyti pagal priėjimo prie signalo apdorojimo metodus (Arora, 2012):

1. akustinis fonetinis metodas;
2. šabloninis atpažinimo metodas;
3. dinaminio laiko skalės kraipymo metodas;
4. vektorinio kvantavimo metodas;
5. dirbtinio intelekto metodas;
6. dirbtinio neuroninio tinklo metodas;
7. atraminių vektorių metodas.

Akustinis fonetinis metodas pagal Hemdalą ir Hughesą apibrėžiamas kaip fonetinių vienetų akstinių savybių rinkinių panaudojimas atpažįstant garsą (Arora, 2012). Fonemų akustinės savybės yra kintamos, bet dėl to, kad jos keičiasi tolygiai, jų galima išmokyti kompiuterį. Šis metodas išreiškiamas trimis žingsniais. Pirmiausiai reikia atlikti spektrinę signalo analizę, kurioje spektriniai matavimai tampa funkcijų rinkiniais. Tada tie funkcijų rinkiniai suskirstomi į izoliuotas sritis ir pridedamas vienas ar daugiau fonetinių žymeklių kiekvienam iš jų. Šis akustinio fonetinio metodo etapas dar vadinamas segmentavimu ir ženkliniu. Paskutinis žingsnis yra žodžių radimas jau iš gautų fonetinių žymeklių sekos, suskirstant segmentuotą signalą į ženklinį. Šiais laikais toks garso atpažinimo metodas retai naudojamas, ypač komerciniais tikslais (Bhabad, 2013; Arora, 2012).

Šablono atpažinimo technikos veikimo principas susideda iš dviejų etapų – šablono mokymas ir lyginimas. Šablono mokymas vyksta taip, kad iš pradžių pagal matematinės-struktūrinės sistemos nustatomos signalo matavimo savybės. Jos siunčiamos į mokymo algoritmą arba klasifikatorių, kur visi duomenys sugrupuojami pagal klases į vieną bendrą modelį. Tada gautas modelis lyginamas su kiekvienu kitu galimu atskaitos modeliu, nustatant, ar jis sutampa, ir apskaičiuojant atstumo tarp jų matą. Toks balso atpažinimo metodas tapo viena iš pagrindinių šiuolaikinių technikų, naudojamų daugelyje sričių. Dėl savo skaičiavimo

ypatumų šis metodas gali būti suskirstytas į du realizavimo požiūrius – šabloninį ir stochastinį (Bhabad, 2013; Arora, 2012).



1.3 schema. Šabloninis atpažinimo metodas

Šabloninis realizavimo požiūris pagrįstas tuo, kad žmogaus pasakyti žodžiai lyginami su šablonų rinkiniu, randant balso (signalų požymių) atitikimus (1.3 schema). Prototipiniai modelių rinkiniai saugomi sistemoje kaip orientaciniai ir naudojami palyginti su kiekvienu nežinomu žodiniu pasakymu, randant labiausiai atitinkantį. Toks metodas yra efektyvus ir dažniausiai gaunamas tinkamas rezultatas, bet naudojant šablonus kiekvienam žodžiui, gali būti apsunkintas kompiuterio darbas. Siekiant išvengti šio trūkumo, bandoma, naudojant tam tikrus algoritmus, nustatyti kalbos kadrų sekas modeliui per tam tikrą vidurkinimo procedūrą arba panaudoti specialią dinaminio programavimo formą (Bhabad, 2013; Arora, 2012).

Stochastinis požiūris yra tikimybių modelių pritaikymas atpažįstant žmogaus balsą. Ši technika labiausiai tinka garsui atpažinti, nes ji išsprendžia daugybę problemų, pavyzdžiui, neišsamią informaciją, painius garsus, kontekstinę poveikį ir t. t. Vienu iš populiariausių stochastinės atpažinimo sistemos klasifikatorių laikomas *PMM*. *PMM* veikimo principas – naudojami du esminiai kalbos identifikavimo kintamieji – išvesties pasiskirstymo rinkiniai ir ribotos būsenos Markovo modelis. Tačiau net toks būdas turi savo trūkumų projektavimo etape:

- tinkamiausios ribotų būsenų sekos nustatymas;
- modalinių kriterijų kontroliavimas;
- konkrečiam Markovo modeliui pateiktas stebėjimo eilės tikimybės įvertinimas.

Tik kai visos šios pagrindinės problemos išsprendžiamos, galima taikyti *PMM* atpažinimo sistemą. Dauguma dabartinių garso atpažinimo programų naudoja šį, jau klasikiniu laikomą, kalbos identifikavimo metodą (Bhabad, 2013; Arora, 2012).

Kitas algoritmas, naudojamas vietoj šabloninio arba akustinio fonetinio metodo, yra dinaminio laiko skalės kraipymo technika. *DLSK* galima realizuoti bet kokiais atpažinimo

sistemos, kurios įvesties duomenis galima konvertuoti į tiesinę formą. Jos esmė yra tokia, kad lyginami dviejų sekų panašumai laiko ir greičio atžvilgiu. Laiko dimensijos yra netiesiogiai iškreiptos, todėl *DLK* dažniausiai taikomas su *PMM* (Arora, 2012).

Vektorinio kvantavimo metodas yra paprastesnis signalo analizės metodas dėl savo duomenų sumažinimo. Per testavimą įvertinamos visos kodinės knygos ir balso atpažinimo sistema pasirenka žodį, turintį mažiausią kodų knygoje atstumo matą. Kodų knyga – tai duomenys, gaunami naudojant domeno transformavimo metodus, pavyzdžiui, „Linde-Buzo-Grey“ algoritimą arba paprastus klasterizavimo algoritmus. Funkcijų mokymo etape atliekami kodų knygų mokymai ir įrašytų žodžių vektorių pasirinkimai. Per testavimą vyksta nežinomo žodžio ir saugomų žodinių modelių lyginimas. Kai žodis turi mažiausią atitikimo rezultatą, jis išvedamas. Kodų knyga neturi aiškos laiko informacijos ir jos įrašai gali būti imami iš bet kurios mokymo žodžio dalies. Toks atpažinimo metodas yra gana populiarus dėl savo galimybės suspausti duomenis (Arora, 2012; Kekre, 2011).

Viena kalbos atpažinimo technika naudoja dviejų kitų metodų veikimo principus. Tai vadinama dirbtinio intelekto metodu, kuris yra sujungtas fonetinis akustinis ir šabloninis metodas. Kalbos požymių nustatymo taisyklės taikomos, naudojant akustines fonetines kalbos identifikavimo koncepcijas, o iš šablonų sistemos – klasifikatoriai, kurie galėtų apdoroti pateikiamus duomenis. Šis metodų hibridas dėl savo specifinio apskaičiavimo turi nemažai sunkumų ir problemų, pavyzdžiui, žmonių kalbos taisyklių integravimą į sistemą, tinkamų įvesties duomenų įterpimą ir ekspertų žinių skirtumą. Dirbtinio intelekto metodas padarė didelę įtaką atpažinimo dizainui ir kalbos kintamųjų apibrėžimo vystymuisi (Arora, 2012).

Dirbtinis neuroninis tinklas, kaip buvo pasakojama pirmiau, kai tik buvo sukurtas, negalėjo konkuruoti su kitais atpažinimo metodais dėl mokymosi algoritmų prastumo ir kompiuterių galingumo tais metais. Tik prieš keletą metų šioje garso atpažinimo srityje buvo atlikta reikšmingų pasiekimų, tada jis tapo viena iš perspektyviausių technikų. *DNT* turi ne vieną paslėptą sluoksnį tarp jo įvesčių ir išvesčių. Tai suteikia galimybę tiksliau apskaičiuoti didelius duomenų kiekius. Paskutiniai šio metodo tyrimai parodė, kad jis gali efektyviau dirbti su akustiniais modeliais negu tokie atpažinimo modeliai kaip Gauso mišinių. *DNT* mechanizuoja akustinių savybių rinkinių analizavimą ir žinių integravimą į sistemą. Dėl apskaičiavimo stiliaus ir vienetų tiesinio sujungimo, *DNT* panašus į nervų sistemą. Tokie modeliai dar vadinami lygiagrečiu arba masyviai paskirstytu apdorojimu. Apdorojimo elemento paprastumas ir modelio vienodumas suteikia galimybę efektyviau kontroliuoti tinklą ir diegti jį aparatinėje įrangoje. Šio metodo pagrindinis dabartinis trūkumas yra didelis iteracijų

kiekis mokymų duomenyse, kuris gali pakenkti ir būti per daug sudėtingas apskaičiuoti (Arora, 2012; Hinton, 2012).

Paskutinis iš plačiai naudojamų kalbos atpažinimo metodų yra atraminių vektorių klasifikavimas (AVK). Šis identifikavimo metodas naudoja diskriminacinį požiūrį, turintį daug apribojimų ir reikalavimų. Tai reiškia, pavyzdžiui, tai, kad klasifikuotiems duomenims reikia panaudoti apibendrintą tiesinį klasifikatorių su maksimalios ribos funkcijomis. Su šiuo algoritmu atraminis vektorius gali naudoti kintamo ilgio duomenis, kurie buvo konvertuoti į fiksuotus. AVK turi problemų su paslėptų vienetų skaičiais ir dimensiškumu, kurie padidina modelio sudėtingumą. Tam išvengti dažniausiai taikomi neuroninio tinklo arba klasikiniai statistiniai metodai. AVK mokymo etapo metu perkelia visus netiesiškumus į ypatybes, panaudojant tinkamus algoritmus, kurie gali sukurti daug netiesinių funkcijų ir atlikti adaptyvaus skaičiavimo pasirinkimą. Tokiu atveju tai padės išvengti lėto mokymo proceso. Galų gale, nors toks kalbos atpažinimo metodas ir turi savo sunkumų, jis labiausiai tinka vidutinio arba didelio žodyno nuolatinei kalbai identifikuoti (Arora, 2012; Zhang, 2014).

Kiekvienas iš šių garso identifikavimo metodų skirtas vienam tikslui – jie atpažįsta pateikiamą signalą. Jų pagrindiniai skirtumai yra kalbos žodyno apimtis, algoritmų sudėtingumas, įvedamos kalbos tipas ir požymiai, kurie gaunami iš to signalo. Dėl to, kad nėra bendro susitarimo, kokius kalbos požymius reikia imti iš signalo, metodai turi skirtingus veikimo principus ir požiūrius. Ne visi jie gali visada teisingai pateikti pageidaujamą rezultatą. Tačiau jie turi ir bendrų bruožų, vertinant veikimo principą. Pirmiausiai kiekvienas iš jų atlieka kalbos analizę tam, kad nustatytų garso signalo požymius. Jų gavimas yra vienas iš svarbiausių momentų atpažįstant garsą, nes jie pašalina signalo triukšmo komponentus ir nuo jų priklausys tolesni darbai su signalu.

1.2.3. Signalų požymių nustatymo būdai

Vienas iš esminių etapų atpažįstant kalbą yra jos požymių nustatymas. Dažniausiai jis remiasi dažnine signalų analize. Yra daugybė skirtingų algoritmų, kurie tai įvykdo, pavyzdžiui, tiesinės prognozės kodavimas (TPK), tiesinės prognozės kepstro koeficientai (TPKK), tiesinė suvokimo prognozė (TSP), melų skalės kepstro koeficientai (MSKK), Barkų skalės kepstro koeficientai (BSKK) ir daugelis kitų požymių išskyrimo būdų. Bendrai šis procesas ir signalo bruožų vektorių sekos apskaičiavimas suteikia galimybę gauti kompaktišką kalbos signalo atvaizdavimą. Kalbant apie visus šiuos algoritmus, jų darbą galima suskirstyti į tris etapus:

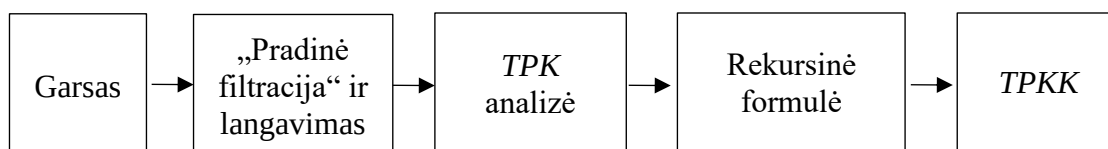
1. spektrinė analizė ir neapdorotų savybių generavimas (Dave, 2013);

2. išplėstinių funkcijų vektoriaus apskaičiavimas (Dave, 2013);
3. išplėstinių funkcijų vektoriaus konvertavimas į kompaktiškesnius vektorius (Dave, 2013).

Nors jų tikslas ir skaičiavimo metodai panašūs, kiekvienas iš jų turi savo skirtumų ir ypatumų bei rezultato gavimas skiriasi. Požymiai, kuriuos jie išgauna ir perduoda klasifikatoriams, gali būti pateikti tik specifiniams algoritmams. Dėl to kiekvienas bruožo nustatymo metodas tinkamas tik tam tikram tolesniam skaičiavimo būdui.

Tiesinio prognozavimo kodavimas arba formančių analizės metodas yra viena iš populiariausių garso įvertinimo technikų. Jis plačiau naudojamas vidutinio arba mažo greičio koderiui ir reikalauja, kad signalas būtų suspaustas prieš pradedant apdoroti. Šis etapas dar vadinamas „pradinė filtracija“, kuriame signalas išlyginamas spektriniu būdu ir perduodamas kaip mažos eilės skaitmeninė sistema. Tai daroma, kad atliekant tolesnį apdorojimą ir analizę, garso bangos nebūtų labai jautrios ribinio tikslumo efektams. Po šio etapo eina lango funkcijos taikymas, kur „pradinės filtracijos“ išvestis paskirstoma pagal tam tikrus kadrus. Tada jiems taikoma Hemingo, Heningo arba stačiakampio lango funkcija, sumažinant signalo nutraukimą. Prieš galutinį etapą vyksta autokoreliacijos analizė, kurioje išgaunamas maksimalus autokoreliacijos kintamasis. Tas kintamasis konvertuojamas į tiesinės prognozės kodavimo parametrus, naudojant Durbino metodą. Šis procesas dar vadinamas *TPK* analize, iš kurios tada gaunami *TPK* parametrai. Tokio garso požymių nustatymo plusai yra perdavimo signalo dydžio sumažinimas ir jo šifravimas (Vijoyo, 2011; Dave, 2013).

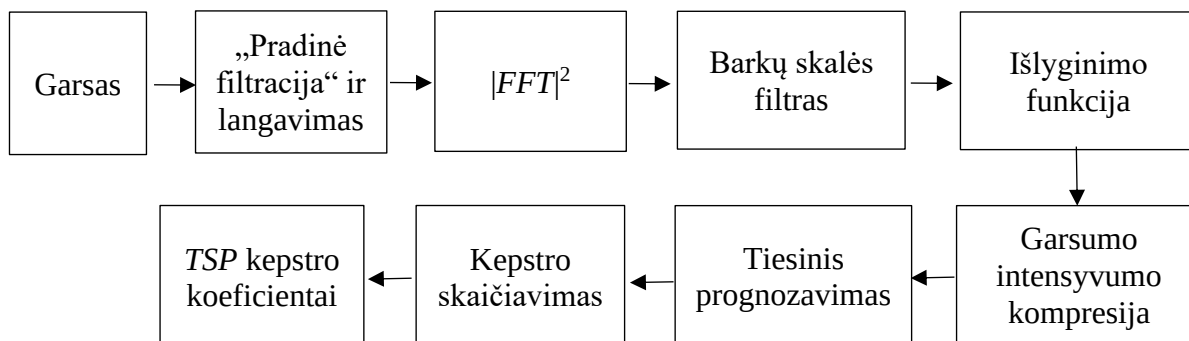
TPKK yra koeficientai, gaunami konvertuojant *TPK* parametrus. Naudojant rekursijos formulę galima apskaičiuoti kepstro koeficientus, apdorojant signalą tokiu pat būdu kaip ir skaičiuojant *TPK* metodą (1.4 schema). *TPKK* laikomas vienu iš geriausių požymių nustatymo metodų, pagal kurį galima identifikuoti žmogų. Jis suteikia stabilų signalo atvaizdą ir sklandesnę spektrinę apvalkalą, bet kaip ir *TPK*, jis turi tokių pat trūkumų (Dave, 2013; Athira, 2017).



1.4 schema. *TPKK* skaičiavimo eiga

Tiesinė suvokimo prognozė (angl. *Perceptual Linear Prediction*) 1982 metais sugalvojo Hynek Hermanskiu. Jo pagrindinė esmė – taikant klausos psichofizikos sampratą modeliuojama žmogaus kalba. *TSP* turi daug panašumų su *TPK* metodu, bet jo atpažinimo

greitis yra geresnis. Taip yra todėl, kad šioje požymių nustatymo technikoje nereikšminga informacija apie kalbą atmetama ir spektriniai parametrai atitinka žmogaus klausos sistemos charakteristikas. Skaičiavimo etapai vyksta pagal 1.5 schemą (Dave, 2013).



1.5 schema. TSP skaičiavimo eiga

Iš pradžių garsas, kaip ir taikant *TPK* metodą, išlyginamas spektriniu metodu, perduodamas į mažos eilės skaitmeninę sistemą, languojant ją į tam tikrus kadrus. Tada galios spektras apskaičiuojamas iš lango signalo, naudojant greitąją Furjė transformaciją, pakeltą kvadratu. Gautas signalas integruojamas į vieną filtrą, dar vadinamą Barkų skalės dažnių filtru. Filtro parametrai išlyginami ir pagal Stevensono galios dėsnį garsumo intensyvumas pakeliamas 0,33. Per tiesinį prognozavimą koeficientai apskaičiuojami kaip galios spektras. Galutiniame etape, kaip ir per *TPK*, siekiant surasti kepstro koeficientus, vyksta gautų signalo duomenų rekursija. Tokiu metodu apskaičiuojami *TSP* kepstro koeficientai, kurie tada taikomi tolesniam žmogaus balso atpažinimui. Šio metodo trūkumas toks, kad *TSP* nesuderinamas su žmogaus klausa, nes dažnis viršija jo klausos diapazoną (Dave, 2013).

Vienas iš populiariausių ir dažniausiai naudojamų požymių nustatymo būdų, ypač mobiliesiems telefonams, yra Males skalės kepstro koeficientai. Pagrindinis jo skirtumas nuo paprasto kepstro yra tai, kad netiesinės dažnių juostos yra lygiai išdėstytos, suteikiant artimesnį atsaką žmogaus klausai. Be to, parametrai, kuriuos *MSKK* ištraukia iš signalo, yra panašūs į charakteristikas, pagal kurias žmonės klausosi kitų kalbos. Šio metodo skaičiavimą galima suskaidyti į penkis žingsnius (Kaur, 2015; Athira, 2017):

1. **„Pradinė filtracija“, kadravimas ir langavimas.** Garso signalas, kaip ir kituose požymių nustatymo metoduose, suskirstomas į laiko kadrus, kurie sudaryti iš mėginių skaičiaus. Kiekvienas kadras tam tikru laiko periodu tada sujungiamas Hammingo, Hanningo, Blackmano arba Gauso lange. Tokiu būdu išnyksta visi nelygumai kraštuose ir gaunama geresnė dažnio raiška (Dave, 2013).

2. **DFT skaičiavimas.** Signalu kadrai apskaičiuojami su greitąja Furjė transformacija, konvertuojant kiekvieną iš jų į dažnio domeną. Tokiu būdu žymiai paspartinamas signalo apdorojimas (Kaur, 2015; Dave, 2013).
3. **Melų filtrų generavimas.** Kiekvienas dažnio domenas praeina pro trikampių juostų filtrų seriją, kuri taip pat vadinama melų skalės filtru. Šis filtras remiasi netiesinių dažnių skale. Daugelio tyrimų metu buvo nustatyta, kad 1000 Hz tonas yra lygus 1000 melų. Viršijus šį reiškinį, signalas taps mažiau jautrus, o melų skalės ir tiesinio dažnio skalės santykis taps logaritminis ir netiesinis. Mažinant hercus santykis taps tiesinis. Filtro darbo veikimo principas yra panašus į žmogaus ausies garsų filtravimą, kai dėmesys sutelkiamas tik į tuos garsus, kurie turi tam tikrą dažnių juostą (Athira, 2017).
4. **Diskretinė kosinusų transformacija.** Paskutinis žingsnis skaičiuojant melų skalės kepstro koeficientus yra paimtų melų skalių konversija į laiko domeną. Šis transformavimas atliekamas naudojant *DKT*. Rezultatas, gaunamas apdorojus, vadinamas *MSK* koeficientais ir išsiskiria aukšto identifikavimo tikslumu ir našumu (Athira, 2017; Ramljak, 2013; Bhattacharjee, 2013).

Pagrindinis šio metodo pranašumas, kaip buvo minėta, yra požymių nustatymas, kurie leidžia beveik be klaidų atpažinti garsą. Be to, jo nustatytos signalo charakteristikos yra panašios į žmogaus klausos atsakus. Tačiau būna, kai net naudojant tokį beveik tobulą metodą būna skaičiavimo klaidų. Tai atsitinka apdorojant signalą su triukšmu, ir siekiant išvengti netinkamo rezultato buvo patarta normalizuoti garso signalą prieš pradedant skaičiavimą. Tokiu būdu sumažinamas klaidų skaičius.

Dar vienas signalo charakteristikų nustatymo metodas yra Barkų skalės kepstro koeficientai. Jo skaičiavimo pradžia turi daug panašumų su *MSKK*, vienintelis skirtumas yra toks, kad naudojamas Barkų skalės filtras. Tolesni veiksmai yra kaip ir su *MSKK*, signalo kadrai suspaudžia atitinkamai vienas kitą pagal garso intensyvumą ir stiprumą, kad nebūtų prarastas ryšys. Paskutinis žingsnis yra signalo praėjimas pro logaritminę funkciją ir *DKT*, po kurio išgaunamas Barkų skalės kepstro koeficientai. Kaip ir melų skalės kepstro koeficientai, *BSKK* gauti kintamieji atitinka žmogaus klausos jautrumo charakteristikas (Gulzar, 2014).

Visi šitie požymių nustatymo metodai turi savo panašumų bei skirtumų. Kiekviename iš jų pirmasis žingsnis yra garso signalo suskaidymas ir langavimas į keletą dalių. Tolesnė eiga jau skiriasi ir nuo jos priklauso, kaip ir kokie žmogaus balso parametrai bus gaunami. Vienas iš geriausių ir dažniausiai naudojamų požymių nustatymo metodas yra *MSKK* skaičiavimas. Jo išvestis geriausiai atitinka žmogaus klausos garsų filtravimą, palyginti su kitomis panašiomis technikomis. Tačiau požymių nustatymas – ne vienintelis procesas, reikalingas garso

atpažinimo sistemai. Be jo, dar svarbus ir reikšmingas etapas yra signalo klasifikacija ir lyginimas.

1.2.4. Signalų atpažinimo klasifikatoriai

Klasifikatoriai garso identifikavimo programoje atlieka apskaičiavimą, kalbančio žmogaus modeliavimo ir požymių vektorių palyginimą. Pirmiausiai sukuriamas modelis, kuris turi signalo požymių vektorių rinkinį, skirtą juos toliau mokyti. Galiausiai atliekamas lyginimas, kurio metu nustatoma, ar yra panašumų tarp nežinomo asmens kalbos charakteristikų parametrų ir sukurto modelio. Kaip buvo minėta pirmiau, yra du būdai, kaip galima lyginti signalus. Pirmas – atsižvelgiant į patį ištartą tekstą, o antras – nepriklausomai nuo jo. Šios dvi technikos dar vadinamos etalonine ir stochastine. Iš jų daug efektyvesnis yra stochastinis metodas, kuris sukuria modelį, nurodantį laike kintančius garso požymius, ir pagrįstas tikimybėmis (Kare, 2015).

Per daugelį metų padaugėjo pačių klasifikatorių ir žmonės vis bando surasti idealų metodą, kad galėtų gauti visiškai tobulą rezultatą, atpažįstant žmogaus balsą. Populiariausi iš jų yra dinaminis laiko skalės kraipymas (*DLSK*), Gauso mišinių modelis (*GMM*), paslėptasis Markovo modelis (*PMM*) ir dirbtiniai neuroniniai tinklai (*DNT*) arba *gilusis mokymasis*.

Pirmiausiai plačiau bus aptartas ne toks populiarus, bet taip pat turintis nemažą panaudojimo sritį, *DLSK* klasifikatorius. *DLSK* dažniausiai taikomas mobiliesiems telefonams dėl savo paprasto ir greito mokymosi proceso. Kaip buvo minėta, šis metodas naudojamas tik tada, kai signalo požymius galima konvertuoti į tiesinę formą. Jis pagrįstas tuo, kad lyginamas dviejų signalų laikas ir greitis. Klasikinis metodo variantas susideda iš vieno modelio kiekvienam identifikuotam žodžiui. Tačiau pagrindinė jo skaičiavimo problema yra rasti labiausiai atitinkantį žodį (Kare, 2015).

Dinaminis programavimas ir šablonas, kuriame yra skirtingi sutampantys tarimo ilgiai, yra *DLSK* algoritmo esmė. Jo tikslas – panaudojus tinkamas formules surasti optimalių panašumų tarp dviejų kalbos funkcijų vektorių sekų. Siekiant nustatyti signalų atitikimą skaičiavimo metu, reikia laikytis tam tikrų taisyklių (Sun, 2014):

1. **Monotoninė sąlyga.** Apribojimo reikšmė yra tokia, kad pirmenybės tvarka sekoje po sukūrimo turi būti pastovi.
2. **Nuolatinė sąlyga.** Visi sekų elementai neturi būti praleisti ir pirmosios sekos indeksas bei bet kokie kiti sekos indeksai privalo atitikti vienas kitą.
3. **Ribinė sąlyga.** Sąlygos esmė yra tokia, kad pirmieji ir paskutiniai abiejų sekų elementai turi būti suderinti vienas su kitu.

4. Reguliavimo lango sąlyga. Ji apibrėžia skaičiavimą, kai ne visi taškai naudojami deformacijos funkcijoje. Taškai, esantys už reguliavimo lango, ignoruojami ir nenaudojami skaičiuojant.

Išlaikius šiuos apribojimus, galimas *DLSK* panaudojimas ir teisingų rezultatų gavimas. Klasikinio *DLSK* formulę galima išreikšti taip:

$$D(i, j) = \min \begin{pmatrix} D(i-1, j) + d(i, j) \\ D(i-1, j-1) + 2d(i, j) \\ D(i, j-1) + d(i, j) \end{pmatrix} \quad (1)$$

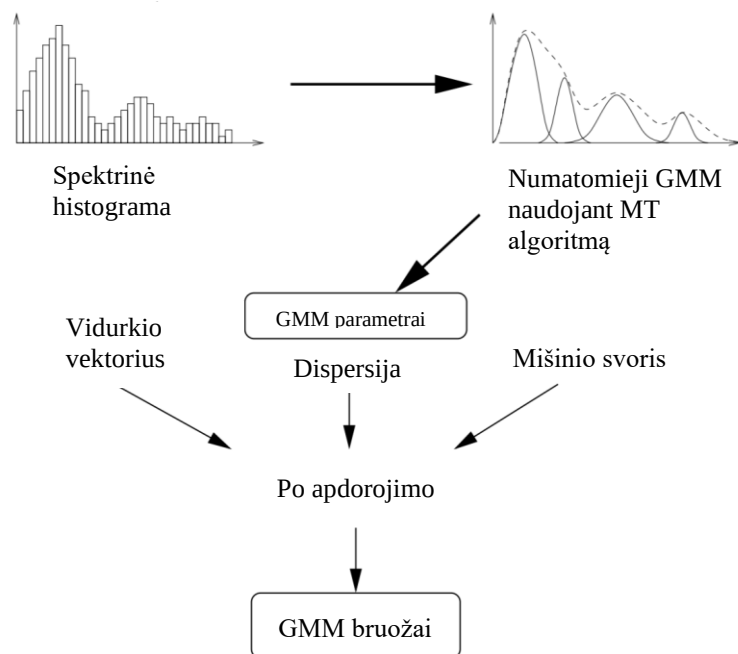
Šioje 1 formulėje $D(i, j)$ yra minimalus arba optimalus kelias nuo pirmosios sekos iki paskutiniosios, i ir j yra X ir Y ašis, o d yra iškraipyto kelio distancija nuo pirmojo elemento iki paskutiniojo (Sun, 2014).

Per daugelį metų žmonės sugalvojo daugybę įvairių šios technikos variantų ir patobulinimų, pavyzdžiui, daugiafunkcis *DLSK* arba *dDLSK*. Jis veikia greičiau ir skaičiavimo būdas yra lengvesnis negu originalo, bet identifikavimo tikslumas tampa prastesnis. Kitas, geresnis, metodas yra šabloninis *DLSK* arba *šDLSK*, kuris ne tik geriau atpažįsta žmogaus balsą, bet ir jo greitis nėra daug lėtesnis už originalaus dinaminio laiko skalės kraipymą (Sun, 2014). Apskritai *DLSK* metodas nėra tobulas ir galingas atpažinimo signalo klasifikatorius. Jis negali būti naudojamas didelėms automatinėms identifikavimo sistemoms. Be to, jis yra priklausomas nuo žodžio tarimo ir turi nemažai sąlygų atliekant skaičiavimą.

Kitas metodas, kuris yra plačiai naudojamas ir beveik vienas iš geriausių technikų, yra vadinamasis Gauso mišinių modelis arba *GMM*. Šis požymių lyginimo skaičiavimas nepriklauso nuo tariamų žodžių ir statistškai parodo, kaip kalbantis žmogus sako garsus. Skaičiavimas pagrįstas tikimybės tankio funkcija ir maksimalaus tikėtimumo (*MT*) algoritmu. *MT* yra vienas iš Gauso modelio optimizavimo algoritmų, kurio pagrindinis tikslas – keletą kartų atnaujinti modelio parametrus. *GMM* naudojimo sritis yra platesnė negu *DLSK*. Dažniausiai jis taikomas garsiakalbių atpažinimo sistemoje kaip biometrinių sistemų funkcija. Jo skaičiavimo procesas prasideda nuo numatomų Gauso parametrų gavimo iš spektrinės histogramos, taikant *MT* algoritmą. Tada Gauso mišinių tankis yra M komponentų tankio suma ir gali būti išreikšta pagal 2 formulę (Stuttle, 2003; Chauhan, 2016).

$$p(\vec{x}|\lambda) = \sum_{i=1}^M p_i \frac{1}{(2\pi)^{D/2} |\Sigma_i|^{1/2}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu}_i)' \Sigma_i^{-1}(\vec{x}-\vec{\mu}_i)} \quad (2)$$

Pagal ją $\vec{x}$ yra dimensinis atsitiktinis vektorius, p_i , $i=1, 2, \dots, M$ – mišinių svorių koeficientai, $\vec{\mu}_i$ – vidurkio vektorius, Σ_i – kovariacinė matrica ir λ – parametrų rinkinys. Taip gaunami *GMM* bruožai, kurie vėliau naudojami modelių lyginimo etape (1.6 schema).



1.6 schema. GMM bruožų skaičiavimo eiga

Kaip ir visos atpažinimo technikos, *GMM* turi savo trūkumų ir problemų, trukdančių kokybiškai ir be klaidų atpažinti žmogaus balsą. Viena iš tokių yra pradinių parametrų, kurie bus taikomi *MT* algoritme, inicializavimas bei mišinių eilės *M* parinkimas. Šių problemų sprendimas gali būti atliktas tik eksperimentiniu būdu, nes pastovių formulių ar technikos nėra. Be to, *GMM* yra jautrus triukšmui ir be slopinančių ar šalinančių filtrų apskaičiuojant bus gaunamas prastesnis rezultatas (Stuttle, 2003; Chauhan, 2016).

Klasikinis garso atpažinimo metodas, kuris iki šiol daug kur taikomas, yra vadinamasis paslėptasis Markovo modelis (*PMM*). Tai yra statistinis modelis, kuriame generuojama sistema su paslėptais parametrais. Jis buvo sukurtas dar prieš 40 metų Lenu Baumu ir paprasčiausias jo realizavimas gali būti išreikštas dinamišku Bajeso tinklu. *PMM* kartais dėl paslėptųjų kintamųjų, kurie valdo kiekvienam stebėjimui atrinktus derinio komponentus, laikomas derinio modelio apibendrinimu. Kiekviename diskrečiame laiko žingsnyje vyksta perėjimas į naują būseną. Pereinamojo ir išvesties komponento pasirinkimai yra atsitiktiniai. Jų reguliavimas vyksta pagal tikimybių pasiskirstymą. Taip visos būsenos generuoja seką, kuri pateikia duomenis apie tam tikrus signalo bruožus. Kitais žodžiais tariant, kiekvienas kalbėtojo žodis tampa tam tikra būsena, sauganti garso bruožus. *PMM*, laikui einant, kuriamos naujos pastebimos išvesties būsenų sekos, bet kitos aplankytos sekos yra paslepiamos. Bendrai paslėptieji Markovo elementai susideda iš penkių elementų (Deng, 1994; Juang, 1991; Stamp, 2018; Levinson, 1983; Kaur, 2015; Chavan, 2013; Trivedi, 2014):

1. *N* yra paslėptųjų specifinių būsenų skaičius.
2. *M* yra stebėjimo simbolių skaičius atskiroje būsenoje.

3. A yra būsenos perėjimo tikimybė, kuri skaičiuojama pagal 3 formulę, kur q_t užpildo dabartinę būseną.

$$A = \{a_{ij}\} = p\{q_{t+1} = j | q_t = i\}, 1 \leq i, j \leq N_n \quad (3)$$

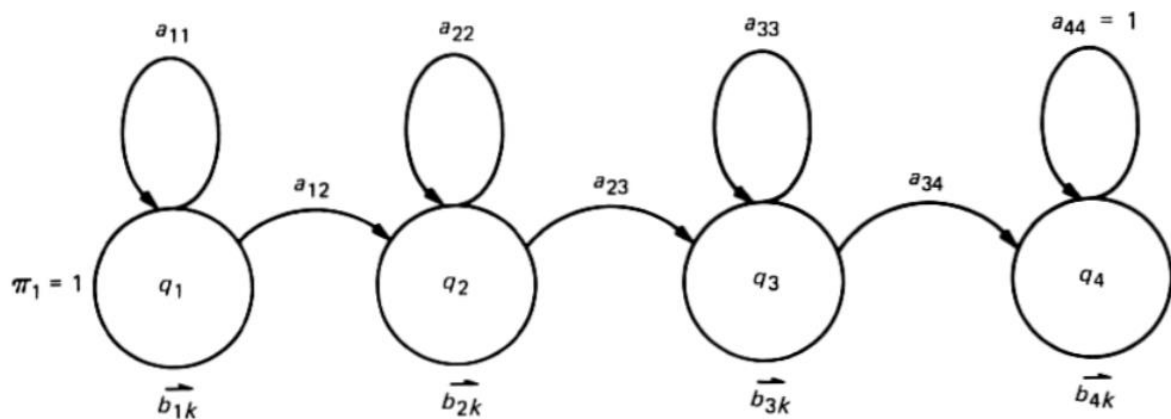
4. B yra stebėjimo simbolio tikimybės išdėstymo matrica. Jos apskaičiavimas vyksta naudojant dabartinį parametrų vektorių Q_t ir V_k kintamąjį, kur k yra stebėjimo simbolis abėcėlėje (4 formulė).

$$B = b_j(k) = p\{Q_t = V_k | Q_t = j\}, 1 \leq j \leq N, 1 \leq k \leq M \quad (4)$$

5. Π yra pradinės būsenos pasiskirstymas, apskaičiuojant pagal 5 formulę.

$$\Pi = \pi_i = P[q_1 = S_i] \quad 1 \leq i \leq N \quad (5)$$

Naudojant visus šiuos elementus, gali būti generuojamos modelio stebėjimo sekos. Pati paprasčiausias Markovo modelio forma yra vadinamasis kairės-dešinės modelis, kuris pavaizduotas 1.1 paveikslėlyje (Chavan, 2013; Trivedi, 2014).



1.1 pav. Standartinis kairės-dešinės PMM modelis

Pagal iliustruotą pavyzdį, veikiant PMM technikai, kiekvieną kartą žingsnis pereina iš dabartinės būsenos į vieną iš prijungtų. Perėjimo tikimybė A nustatoma, kai vyksta tam tikras žingsnis iš būsenos į būseną. Kai būseną įveda, sukuriamas funkcijų vektorius. Jis gaunamas iš pasiskirstymo, susijusio su įvesta B reikšme. Pagal šią PMM formą galima apibūdinti keletą standartinių sąlyginai nepriklausomų prielaidų:

- būsenos, nes neturi įtakos visoms kitoms būsenoms, turinčioms ankstesnę būseną;
- stebėjimai, nes veikia savarankiškai nuo visų kitų stebėjimų, atsižvelgiant į tai, kokią būseną sukūrė.

Apskaičiuojant paslėptąjį Markovo modelį, yra trys pagrindiniai algoritmai, veikiantys priklausomai nuo atpažinimo sistemos rūšies. Atskiram žodžiui identifikuoti pakanka „Forward“ algoritmo, bet galingesnei atpažinimo sistemai reikėtų papildyti algoritmą, kad jis galėtų išmokyti save. Toks skaičiavimo metodas vadinamas „Forward-backward“ algoritmu. Jo

pagrindinis tikslas – surasti garso imties tikimybę, išmokyti ir įvertinti nenustatytus parametrus. Kai atpažįstant šneką naudojamas didelės imties žodynas, generuojamas statinis arba dinaminis žodžių tinklas. Šis tinklas yra sudarytas iš daugelio PMM. Atlikus skaičiavimą mokymosi proceso metu gaunamas rezultatas yra PMM modelio bruožų didžiausios tikimybės ir galinio režimo įvertinimas. Kitas standartinis algoritmas yra k vidurkio algoritmas, kuris reikalingas siekiant sugrupuoti stebėjimus į k skaidinius, generuojant viso balso mėginio funkcijų kodo knygą. Šį algoritmą galima apibrėžti klasterių priskyrimo ir naujų vidurkių apskaičiavimo žingsniais. Procesas vyksta labai greitai ir rezultatas yra geriausias grupavimas, atliekant keletą iteracijų. Paskutinis, ne mažiau svarbus algoritmas, yra „Viterbi“, kuris dažniausiai naudojamas žmogaus balsui nuolat identifikuoti sistemoje. Jo esmė – iš gautų modelio parametrų ir stebėjimo vektorių apskaičiuoti vertę, kuri derintųsi su visais specifiniais žodininiais modeliais. Tam dažniausiai taikomas žetono siuntimo principas (Chavan, 2013; Trivedi, 2014).

PMM dažniausiai naudojamas kartu su kitais klasifikatoriais, siekiant pagerinti rezultatą. Tai daroma išvengiant apribojimų ir trūkumų, atsirandančių naudojant šį klasifikatorių. Vieni, galbūt reikšmingiausi, šio metodo trūkumai yra būtinas didelis kiekis mokymų duomenų, lėtas modeliavimas, sąlyginis nepriklausomumo prielaidos ir modelio topologijos formalus pasirinkimas. Dažniausi taikomi klasifikatoriai, kompensuojantys PMM, yra pirmiau aptarti GMM, DLSK metodai. Be to, yra dar vienas klasifikatorius DNT, kuris gali būti naudojamas su PMM, tačiau gali būti naudojamas ir atskirai (Trivedi, 2014).

Per keletą paskutiniųjų metų DNT atpažinimo sistemos klasifikatorius tapo vienu iš perspektyviausių ir greitai populiarėjančių metodų. Dažniausiai jis naudojamas komplikuotiems signalo požymiams statiskai sugrupuoti į paprastą rezultatą. Jis dar vadinamas statinio modelio identifikavimo procesu. Šio klasifikatoriaus sudedamosios dalys yra panašios į žmogaus smegenų neuroninį tinklą. DNT taip pat padarytas iš paprastų procesorių neuronų, kurie po duomenų apdorojimo siunčia juos toliau per neuronų išeinantį ryšį. Išeinantis ryšys perduoda tą patį signalą per keletą šakų iš vieno neurono į kitą (Trivedi, 2014; Lippmann, 1990; Tebelskis, 1995).

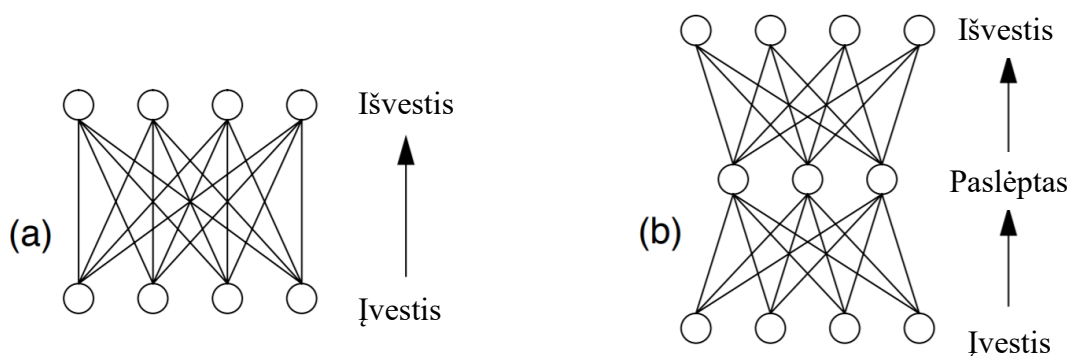
Nuo 1986 metų, kai Rumelhartas sugalvojo „atgalinio sklidimo“ algoritmą, DNT buvo realizuotas daugelyje sėkmingų atpažinimo sistemų – pavyzdžiui, foneminėje, išreikštoje skaitmeniniame identifikavime, bei balso / suduslėjusioje diskriminacijoje. Šio algoritmo tikslas – išmokyti neuronus ir išspręsti prognozavimo, klasifikavimo problemas. Tinklo svoriai proporcingai modifikuojami pagal įėjimo duomenis ir pastebėtas gauto rezultato vieneto aktyvinimo paklaidas. „Atgalinio sklidimo“ algoritmas nėra esminis DNT metode, be to, yra daugelis kitų mokymo algoritmų, kurie padeda gauti geresnį rezultatą. Pagrindiniai ir dažniausiai naudojami duomenims treniruoti skaičiavimai yra trys (Trivedi, 2014; Tebelskis,

1995):

1. **Mokymasis su mokytoju.** Tai skaičiavimo technika, kai kiekvienam signalo požymio bruožų vektoriui pateikiami rezultato tikslai ir ištaisomos tinklo klaidos. Jis gali būti naudojamas pasikartojančiuose ir pranašesniuose tinkluose (Trivedi, 2014; Tebelskis, 1995).
2. **Mokymasis be mokytojo.** Tai metodas, kai tinklas be jokios kontrolės pats atlieka duomenų apdorojimą. Siekiant atlikti šį skaičiavimą, dažniausiai taikomi kvantavimo, atvaizdavimo arba klasifikavimo algoritmai. Nekontroliuojamo mokymosi esmė garso atpažinimo sistemoje – veikti kaip klasterizavimui. Rasti mėginiai su vienodomis savybėmis sugrupuojami į bendras klases. Tokiu būdu daugiau informacijos išsaugoma paslėptame sluoksnyje ir atliekamas matmenų mažinimas. Pats žymiausias šio mokymo būdo pavyzdys yra Hopfieldo tinklas. Savo 1982 m. tyrime jis parodė, kad naudojant pakartotinio tipo tinklą su Hebbo taisykle, sugadinti mokymo modeliai gali būti pataisomi artimiausioje būsenos erdvėje. Tai padeda stabilizuoti *DNT* modelius ir perduoti geresnius rezultatus (Trivedi, 2014; Tebelskis, 1995).
3. **Sustiprintas arba pusiau prižiūrimas mokymas.** Šio mokymo modelio duomenys nepateikiami, bet paties tinklo elgesys yra įvertinamas ir prižiūrimas pagal tam tikrą dinamiką. O tinklai skiriasi ir mokymosi veikimo principu, ir topologijos forma. Per šį duomenų treniravimo procesą bus numatomos kaupiamosios išlaidos ir mažinamos sąnaudos, atradus politiką, kurioje gali būti atrenkami veiksmai (Trivedi, 2014; Tebelskis, 1995).

Šie mokymo procesai, vertinant gautų duomenų rūšį, jų eiliškumą tinkluose ir topologijos formą, treniruoja modelius taip, kad būtų gautas pageidaujamas rezultatas. Be pasiskirstymo pagal mokymo procesą, *DNT* gali būti paskirstytas pagal du tinklo tipus – tiesioginio sklidimo ir pasikartojimo.

Tiesioginio sklidimo tinklas – tai tinklas, paskirstomas į vieno arba kelių sluoksnių perceptronus, kuriuose nėra jungčių tarp vienetų (1.2 pav.).



1.2 pav. (a) Vieno sluoksnio perceptronas, (b) daugiasluoksnis perceptronas

Vieno sluoksnis perceptronas veikia pagal „Perceptron“ taisyklės, kurios esmė yra svorio rinkinio padidinamas arba mažinamas, jeigu įvestis yra aktyvi, o išvestis neteisinga. Tai daroma siekiant gauti norimą išvestį, pagrįstą arba 1, arba 0. Tokiu būdu užtikrinamas svorių rinkinio radimas ir modelių klasifikavimo veikimas bet kokiame mokymo etape. Be to, reikalaujama, kad patys modeliai būtų išdėstyti pagal liniją. Jei tai neįmanoma, tada vieno sluoksnio perceptronas netinka kaip *DNT* tipas modelio apdorojime – tada reikalingas daugiasluoksnis perceptronas (Hinton, 2012; Trivedi, 2014; Tebelskis, 1995).

Kelių sluoksnių perceptronas (*KSP*), kitaip vadinamas giliuoju neuronų tinklu (*GNT*), yra neuronų tinklas, sudarytas iš kryptinio ciklo junginių. Tinklas turi ne tik įvestį ir išvestį, bet ir paslėptus vienetų, pagrįstus logistine funkcija, siekiant konvertuoti pradinius duomenis į skaliarinę būseną. Ta skaliarinė būsena vėliau siunčiama į aukščiau esantį sluoksnį arba į galutinį sluoksnį. Toks mokymo procesas yra daug sudėtingesnis už paprastą vieno sluoksnio perceptroną. Tačiau gaunami tokio duomenų mokymo rezultatai yra tikslesni. Naudojant *GNT* su „atgalinio sklaidimo“ algoritmu, tinklo klaidų skaičius sumažėja, o tikimybė gauti pageidaujamą rezultatą – padidėja. Yra daugybė architektūrinių būdų, kaip pagerinti *KSP*, pavyzdžiui, laiko uždelsimo tinklas arba išmokyto vektoriaus kiekio nustatymas. Visi jie atlieka tam tikrą modelių suvaržymą, siekiant apriboti svorio vertes, jų ryšius arba susieti reikalingus svorius (Hinton, 2012; Trivedi, 2014; Tebelskis, 1995).

DNT yra vykstantis atpažinimo bei kitų sričių duomenų apdorojimo metodas. Tik paskutiniaisiais metais šioje technikoje buvo padaryta žymių algoritmų pakeitimų ir patobulinimų. Pagrindiniai šio klasifikatoriaus pranašumai ir trūkumai, palyginti su kitais, pavaizduoti 1.1 lentelėje.

1.1 lentelė. *DNT* pranašumai ir trūkumai

Pranašumai	Trūkumai
Ne parametrinis modelis, nereikalauja aukšto statistikos lygio	Reikalauja daug pranašumų ir apribojimų

Lengvai suprantamas netiesinis modelis, kuriuo galima paprastai manipuluoti	Lėta konvergencija naudojant kartu su „grįžtamojo dauginimo“ algoritmu
Netiesinis modeliavimas	Sudėtingesnis skaičiavimas ir kartais veikimo principas
Gera išmokyta metodas duoda tikslius rezultatus	

Pagal pateiktą lentelę galima teigti, kad *DNT* turi nemažai pranašumų, bet turi ir trūkumų. Tačiau rezultatai, gaunami naudojant šį modelį, laikomi aukštais. Kartais šio klasifikatoriaus tikslumas gali būti geresnis už klasikinį *PMM* metodą (Trivedi, 2014; Tebelskis, 1995).

Per daugelį metų klasifikatorių skaičius žymiai padidėjo ir jų signalo požymių apdorojimo ir palyginimo tikslumas yra skirtingas. Naudojant kai kuriuos iš jų, gaunami tikslesni rezultatai kaip *PMM* ir *DNT*. Yra ir tokių, kurie pritaikyti tik tam tikram duomenų, nustatytų požymių ar signalo apdorojimo būdai. Siekiant pagerinti rezultatą ir sumažinti klaidų skaičių, žmonės sujungia juos ir tada gaunami hibridiniai klasifikatoriai. Patys populiariausi iš jų yra *DLK PMM*, *GMM PMM*, *DNT PMM*, *GMM DNT* ir t. t. Visi šie klasifikatoriai bet koku atveju atlieka kelias bendras funkcijas: pateiktų požymių apdorojimas, palyginimas ir atpažinimo rezultato gavimas. Ne visais atvejais klasifikatoriai gali atpažinti žmogaus balsą – yra tam tikrų apribojimų, metodų trūkumų arba sistemos problemų, neleidžiančių gauti norimo rezultato, nežiūrint į tai, kaip gerai veikia klasifikatoriai.

1.3. Garso atpažinimo sistemos realizavimo problemos

Jau seniai domimasi garso atpažinimo sistemomis, tačiau nepaisant tuo, kad buvo sukurta daug naujos garso užrašymo aparatūros, apdorojimo algoritmų ir technikų, iki šiol vis tiek yra apribojimų arba trūkumų. Žmogaus balso atpažinimo sistema, kaip biometrijos rūšis, nėra visiškai patikima. Daugybė faktorių lemia, koks rezultatas bus gautas atlikus garso apdorojimą ir lyginimą.

Pagrindinė ir pati svarbiausia žmogaus identifikavimo pagal balsą problema yra pats žmogaus faktorius. Balsas nelaikomas pastovia biometrijos rūšimi, nes laikui bėgant kinta dėl aplinkos, dėl psichologinių veiksnių, sveikatos arba paties žmogaus balso tembro pasikeitimo, pradinis įrašytas garso signalas gali neatitikti to paties žmogaus dabartinio balso. Visada yra skirtumas tarp sveiko žmogaus balso ir sergančio, nors jis gali būti ir nežymus. Kartais, net jeigu žmogus sveikas, jo balsas gali skirtis jau kitą dieną nuo įrašyto – dėl svorio ar net amžiaus

kitimo. Būna atvejų, kai žmonės specialiai pakeičia savo balsą norėdami suklaidinti atpažinimo sistemą. Jie keičia toną, garsumą, kalbėjimo stilių, apgaudami identifikavimo programą (Kamarauskas, 2009).

Garso įrašymo įranga bei patalpos, taip pat turi įtakos pageidaujamo rezultato gavimui žmogaus balso atpažinimo sistemoje. Užrašant signalą su nekokybiška įranga, spektre gali atsirasti trikdžių, iškraipymų ar papildomų formančių, prailginančių dažnių juostą. Be to, aplinka, kurioje įrašomas žmogaus balsas, turi didelę įtaką. Dėl papildomų triukšmų, mobiliųjų telefonų patalpose esanti įranga (kompiuteris, mikrobangų krosnelė ir pan.) arba pačių patalpų architektūros žmogaus balso signalas papildomas pašaliniais garsais arba vėluoja originalaus garso kopija, pavyzdžiui, girdimas aidas (Kamarauskas, 2009).

Kalbant apie pačius atpažinimo sistemos metodus ir algoritmus, didžiausia jų problema yra tinkamų garso požymių nustatymas bei tai, kad jie visi priklauso ne nuo kalbančiojo žmogaus, bet nuo kalbos. Per 60 metų, kai prasidėjo domėjimasis žmogaus balso identifikavimo sistemos kūrimu, nebuvo rasta garso požymių, kurie galėtų be jokių klaidų ar apribojimų nustatyti žmogaus balsą. Bet kokie dabartiniai garso signalo bruožai šiek tiek skirsis nuo pradinio įrašyto to pačio žmogaus balso. Klasifikatoriai, naudojami tiems požymiams apdoroti ir jiems lyginti, taip pat turi savo apribojimų ir trūkumų, neleidžiančių gauti visaverčio balso atpažinimo rezultato (Kamarauskas, 2009).

Žmogaus balso bei paties garso identifikavimo programos iki šiol tobulinamos, siekiant išspręsti daug problemų. Yra pakankamai daug skirtingų faktorių, lemiančių tinkamo rezultato gavimą. Visiškai teisingas išvesties suradimas kiekvienu realizavimo atveju iš skirtingų atpažinimo sistemų negalimas. Ši mokslinių tyrimų ir programavimo sritis vis dar vystoma, siekiant, kad iš nepastovios biometrijos rūšies ji taptų viena iš patikimiausių.

1.4. Skyriaus apibendrinimas

Iš aptartų žmogaus balso atpažinimo sistemos problemų, galima padaryti išvadą, kad tinkamų garso požymių ir klasifikatorių pasirinkimas lemia, koks rezultatas bus gaunamas. Ši programavimo sritis yra aktuali dėl didelės paklausos (duomenų bei kitų dalykų saugojimo ir prietaisų kontroliavimo balsų). Garso atpažinimo procesas turi daugybę skirtingų realizavimo būdų ir metodų, todėl svarbu išrinkti geriausią iš jų. Problema ir noras turėti pačią patikimiausią garso atpažinimo sistemą yra didelė, todėl iš visų žinomų garso klasifikatorių išskiriamas klasikinis *PMM* ir dabartinis *DNT* metodas. Lyginant abi sistemas bus nustatyta, kuri iš jų yra pranašesnė atliekant garso požymių apdorojimą. Šiai užduočiai atlikti bus naudojamas *Python* programavimo kodas, realizuojant abi sistemas ir jas lyginant. Atlikus tyrimus bus nustatyta, kaip tinkamai veikia programos.

2. ŽMOGAUS BALSO ATPAŽINIMO SISTEMŲ ANALIZĖ

Magistro darbe buvo analizuojama ir apžvelgiama keletas algoritmų ir metodų, susijusių su žmogaus balso atpažinimo procesu. Buvo pasirinktas melų skalės kepstro koeficientai kaip požymių nustatymo modelis. Jis buvo pasirinktas todėl, kad jo garso savybių diagnozavimas yra panašiausias į žmogaus ausies garsų filtravimą ir tai padidina tikimybę atpažįstant gautą signalą. Kiti pasirinkti garsų lyginimo metodai – paslėptasis Markovo modelis ir giliojo mokymosi būdas. Paslėptasis Markovo modelis yra klasikinis ir dažniausiai naudojamas identifikavimo procese dėl savo didelio atpažinimo tikslumo. Tai yra statistinis modelis, kuris buvo tobulinamas daugelį metų, kol tapo vienu iš geriausių, gebančių lengvai lyginti nuolatinį žmogaus kalbėjimą su savo surinktais duomenimis. Tačiau, kaip buvo pastebėta, kiti populiarėjantys atpažinimo sistemos klasifikatoriai – giliojo mokymosi procesas arba dirbtinis neuroninis tinklas, kurio metu gauto signalo savybės apdorojamos ir statistiškai sugrupuojamos. Be to, gerai išmokytas modelis suteikia didesnę tikimybę atpažinti tinkamą žmogaus balsą. Šie du lyginimo klasifikatoriai buvo pasirinkti, siekiant nustatyti, kiek geresnis rezultatas gaunamas kiekvieno iš jų atveju identifikuojant žmogaus balsą.

Šiame skyriuje plačiau nagrinėjama signalo atpažinimo eiga: kaip vyksta balso modeliavimas ir apdorojimas, kokiomis savybėmis jie pasireiškia. Aptariami procesai, kaip atskiriamas garsas nuo triukšmo. Toliau aprašomas melų skalės kepstro koeficientų skaičiavimo procesas ir balso požymių išskyrimas. Galiausiai analizuojami veikimo principai ir skaičiavimo būdai, tokie kaip *PMM* ir *DNT*, naudojant gautus signalo požymius bandoma identifikuoti, ar tas signalas sutampa su kitu garsu.

2.1. Įvesties signalo modeliavimas ir kūrimas

Žmogaus balsas yra viena iš svarbiausių jo sąveikos su išorine aplinka priemonių. Kasdien žmonės girdi skirtingus garsus ir jų pačių kūnai skleidžia garsą, naudojant garso generavimo sistemą – viršutinius kvėpavimo takus. Ši biologinė sistema suteikia žmonėms galimybę išreikšti savo nuomonę, jausmus ir bendrauti su kitais, naudojant balsą. Pagrindiniai komponentai, sudarantys žmogaus kalbos sistemą, yra (Schickhofer, 2017; Deller, 2005):

- plaučiai;
- ryklės ertmė;
- burnos traktas;
- nosies traktas;

- trachėja.

Visi šie žmogaus balso generavimo komponentai dar vadinami artikulatoriais. Pradedant jiems judėti, kuriama kalba, filtruojant iš plaučių išeinantį orą į skirtingus garsus. Gerklėje esančių balso stygų veikimo principas yra slėgio kritimo per gerklę reguliavimas, siekiant gauti kvaziperiodinę slėgio bangą. Balso dažnis žmogaus kalbėjimo metu gali svyruoti nuo 80 iki 200 Hz, tai priklauso nuo fizinių žmogaus savybių. Raumenų įtempis, masė, ilgis bei žmogaus lytis ir amžius – visi šitie faktoriai lemia, koks bus žmogaus balsas. Pavyzdžiui, vyrų gerklės matmenys yra didesni, todėl jų dažnių diapazonas yra žemesnis negu moterų ar vaikų. Be to, žmogaus liežuvis, lūpos, burna, žandikaulis veikia kaip garso rezonatoriai ir jų judėjimo rezultatas – skirtingi garsai. Balso generavimą galima padalyti į keletą etapų (Schickhofer, 2017; Deller, 2005):

1. **Šaltinio generavimas**, kur gali sklisti vokalizuoti arba nevokalizuoti garsai. Vokalizuotas garsas, dar vadinamas kvaziperiodiniu signalu, generuojamas einant oro srautui iš plaučių, virpant balso stygomis. Nevokalizuotas, arba aperiodinis signalas, gaunamas iš šaltinio einant turbulentiniam garso triukšmui.
2. **Garso sklidimas nuo lūpų iki nosiaskylių**. Šiame etape keičiant lūpų formą sklindantis signalas tampa tam tikru garsu.
3. **Balso trakto artikuliacija**. Tai yra procesas, kai iš plaučių sklindantis garsas filtruojamas pagal tam tikrą vokalinę formą.

Balso trakto modeliavimui apibūdinti dažniausiai naudojamas diskretinio laiko balso trakto modelis, kuris dar vadinamas šaltinio filtro modeliu. Diskretinio laiko balso modelio šaltinis yra baltas triukšmas arba periodinė impulsų seka. Tokio filtro pavyzdys gali būti žmogus, pučiantis saksofoną. Šiuo atveju saksofonas yra filtras, o žmogus yra garso šaltinis. Pučiant orą į saksofoną bei keičiant saksofono vožtuvų padėtį gaunamas skirtingas garsas. Tokiu būdu filtruojamas ir kuriamas visai naujas signalas, turintis savo ypatingų garso duomenų rinkinį (Schickhofer, 2017; Deller, 2005).

Pagal ištirtus kalbos generavimo principus buvo sukurta daug skirtingų modelių, gebančių generuoti dirbtinius garsus praktiškai. Pavyzdžiui, kalbantys ir gebantys bendrauti su žmonėmis robotai, kompiuteryje esantis dirbtinis intelektas bei daugelis garso atpažinimo sistemų. Siekiant tinkamai sukurti gerą žmogaus balso identifikavimo kalbą, reikia panaudoti garso generavimo modelį bei įvertinti gaunamus parametrus. Tai padeda nustatyti balso požymius ir pagal juos identifikuoti, ar nors vienas iš naudojamų kalbos garsų atitinka nežinomą

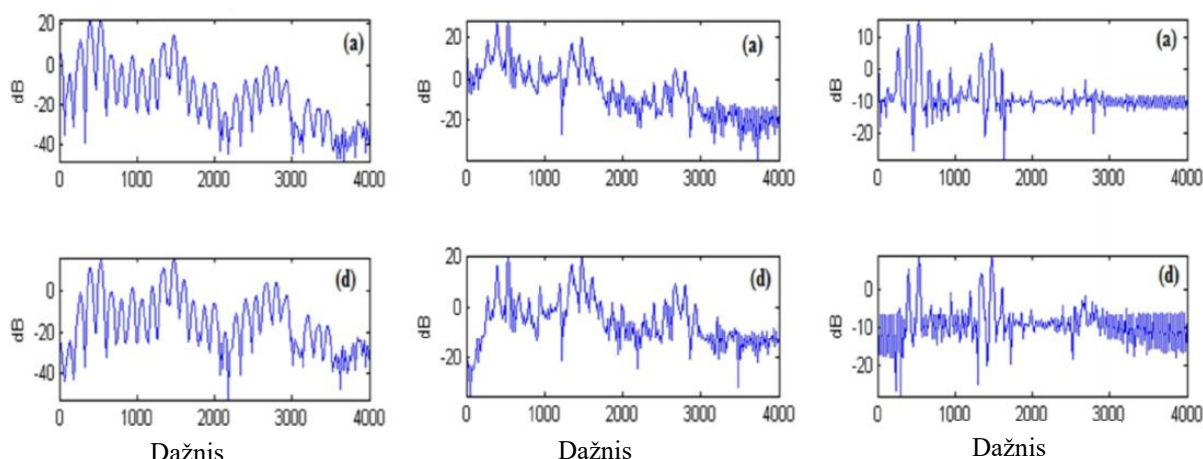
žmogaus balsą. Vadinasi, žmogaus oro filtravimą ir jo balso generavimą galima sutapatinti su skaitmeniniu filtravimu, kurio išvesties signalo parametrus galima panaudoti skaičiavimuose (Schickhofer, 2017; Deller, 2005).

2.2. Pirminis signalo filtravimas ir dalijimas į kadrus

Pirmas žingsnis beveik visose kalbos identifikavimo sistemose yra pradinis signalo filtravimas arba „pradinė filtracija“. Po šio veiksmo jau eina signalo kadravimo ir langavimo etapai. Visi šitie žingsniai dar vadinami pirminiu kalbos apdorojimu. „Pradinė filtracija“ skirtas siekiant subalansuoti aukšto ir žemo dažnio komponentus ir sulyginti spektro plokštumas. Žmogaus balso signalo kritimas yra 6dB per oktavą. Dėl šios priežasties ne visi signalo požymiai bus gauti kokybiškai ir bus prarasta svarbi informacija. Siekiant pagerinti garso signalą, naudojamas ribotos impulsinės reakcijos filtras (*RIR*), slopinantis žemo dažnio komponentes ir iškeliantis aukštus dažnius (2.1 pav.). *RIR* filtrą galima išreikšti pagal 6 formulę (Kamarauskas, 2009; Chauhan, 2015; Loveimi, 2013).

$$\mathfrak{s}(n) = s(n) - as(n - 1) \quad (6)$$

Joje filtruotas garso įrašas gaunamas atimant prieš tai ėjusį diskretizuotą kalbos signalą su a koeficientu iš pradinio signalo. Koeficientas a šiuo atveju yra spektro išlyginimo laipsnis, nuo kurio priklausys garso dažninės amplitudės charakteristikos tiesės pasvirimo kampas (Kamarauskas, 2009).



2.1 pav. Garso signalai prieš ir po pradinės filtracijos (Loveimi, 2013)

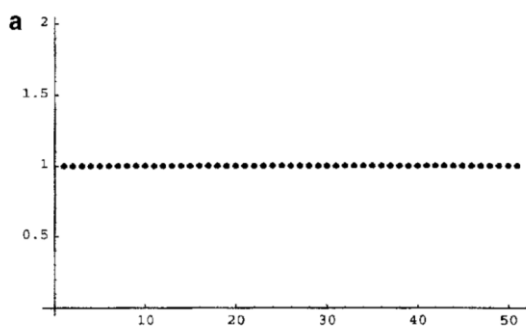
Po pradinio signalo filtravimo gautas naujas įrašas tinkamesnis tolesniam apdorojimui ir požymių išskyrimui negu prieš tai buvęs. Jo energijos kiekis yra paskirstytas ne tik žemesniuose dažniuose, bet ir aukštesniuose.

Kitas pirminio apdorojimo etapas yra garso įrašo dalijimas į kadrus. Kalbos signalas trumpame intervale yra stacionarus. Tai reiškia, kad žmogaus balsas per labai mažą laiko intervalą nespėja pasikeisti ir jo parametrus galima laikyti pastoviais. Tada signalas skaidomas į mažesnius, po $1/3 \sim 1/2$ reimo dydžio persidengiančius kadrus. Dažniausiai tų kadrų trukmės intervalas yra nuo 20 ms iki 30 ms, tačiau garso parametrų išskyrimo etape jie pasislenka po 10–20 ms. Vienas toks kadras gali būti išreikštas pagal formulę (Kamarauskas, 2009; Chauhan, 2015):

$$s(j, n) = s \quad (6)$$

Čia $s(j, n)$ – j -tasis kadras. Jis gaunamas naudojant N – kadro ilgį atskaitomis, O – gretimų kadrų persidengimo ilgį atskaitomis ir patį $s(n)$ – originalų signalą (Kamarauskas, 2009).

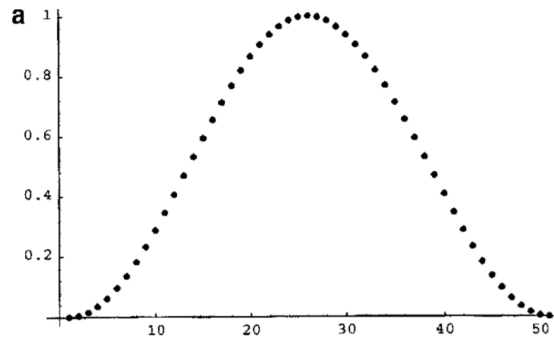
Paskutinis žingsnis prieš tolesnį signalo apdorojimą – lango funkcijos taikymas, kuris gali būti išreikštas dauginant originalų signalą iš lango funkcijos $w(n)$. Pagal šią funkciją pasirenkamas tam tikras intervalas, kuris grąžina baigtinę ne nulio vertę ir nulinę vertę už jo ribų. Šiuo metu yra daugybė skirtingų lango funkcijų, besiskiriančių savo skaičiavimo ypatumais, spektrinėmis charakteristikomis ir surastų viršūnių skaičiumi. Taip pat jas galima sugrupuoti į dvi kategorijas: reguliuojamos ir fiksuojamos lango funkcijos. Pačios populiariausios ir dažniausiai naudojamos lango funkcijos yra fiksuotos (Kamarauskas, 2009; Aparna, 2017; Poularikas, 1999):



2.2 pav. Stačiakampio lango funkcija (Poularikas, 1999)

1. **Stačiakampio lango funkcija** – pati paprasčiausia lango funkcija, skaičiuojama tolygiai dauginant signalo kadrą ir stačiakampio funkcijos (2.2 pav.). Jo esmė yra duomenų sekos vertės keitimas, kad už lango esančios reikšmės būtų paverstos nuliais. Gaunamas rezultatas labai prastas, signalo spektras yra iškraipytas, o surastų viršūnių skaičius yra mažas, palyginti su kitomis lango funkcijomis (Kamarauskas, 2009; Aparna, 2017; Poularikas, 1999). Jos reikšmė apskaičiuojama pagal 7 formulę:

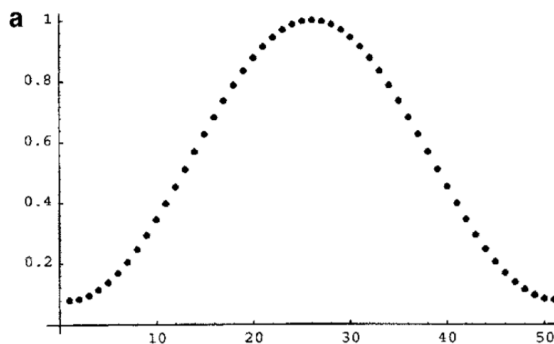
$$w(n) = \begin{cases} 1, & \text{kai } 1 \leq n \leq N; \\ 0, & \text{kitur.} \end{cases} \quad (7)$$



2.3 pav. Haningo lango funkcija (Poularikas, 1999)

2. **Haningo lango funkcija** yra viena iš populiariausių lango funkcijų, naudojama daugelyje programų (2.3 pav.). Pavyzdžiui, jos gaunamas mėginių serijos pogrupis geriausiai tinka, norint atlikti Furjė transformacijos skaičiavimus. Ji skaičiuojama pagal 9 formulę (Kamarauskas, 2009; Aparna, 2017; Poularikas, 1999).

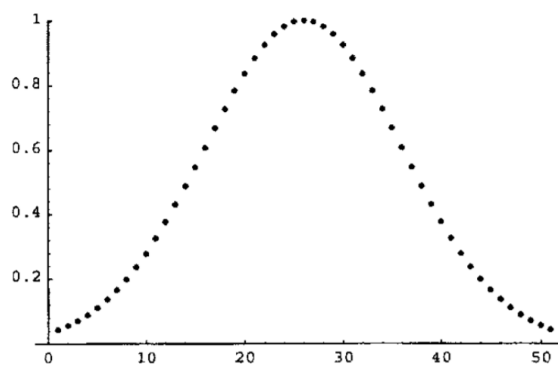
$$w(n) = \begin{cases} 0.5 - 0.5 * \cos \left(2\pi \frac{n}{N} \right), & \text{kai } 1 \leq n \leq N; \\ 0, & \text{kitur.} \end{cases} \quad (9)$$



2.4 pav. Hemingo lango funkcija (Poularikas, 1999)

3. Pagal **Hemingo lango funkciją** kadru suteikiama maždaug penktadalis Haningo lango aukščio (2.4 pav.). Tokiu būdu sumažinama maksimali šoninė skiltis (Kamarauskas, 2009; Poularikas, 1999; Aparna, 2017).

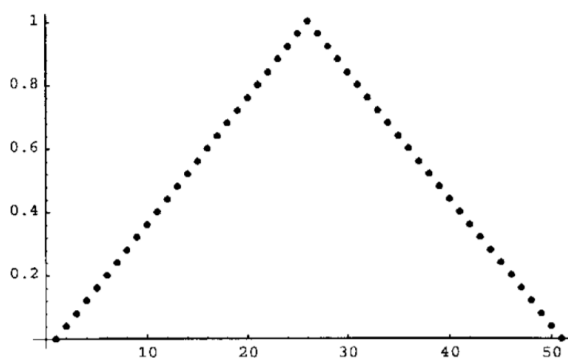
$$w(n) = 0.54 + 0.46 \cos \frac{2\pi n}{N}, \text{ kai } 1 \leq n \leq N \quad (8)$$



2.5 pav. Gauso lango funkcija (Poularikas, 1999)

4. **Gauso lango funkcija** yra varpinės kreivės formos langas, kurio logaritmas sudarytas iš eksponentinės funkcijos su kvadratine funkcija (2.5 pav.). Ši funkcija tinka visiems garso identifikavimo metodams. Rezultatas gaunamas pagal 10 formulę (Aparna, 2017; Poularikas, 1999).

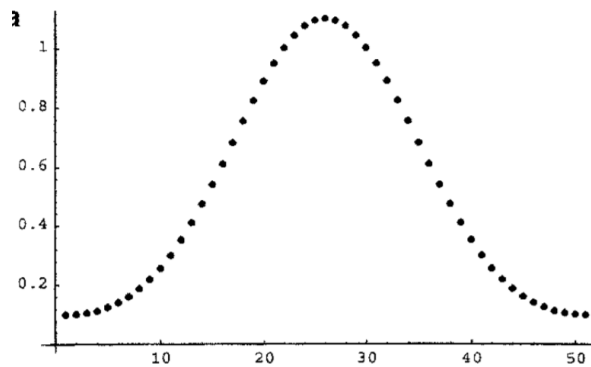
$$w(n) = \begin{cases} e^{\frac{-1}{2} \left(\frac{n}{N} \right)^2}, & \text{kai } 1 \leq n \leq N; \\ 0, & \text{kitur.} \end{cases} \quad (10)$$



2.6 pav. Trikampio lango funkcija (Poularikas, 1999)

5. **Trikampio lango funkcija**, dar vadinama *Barletto*, skiriasi nuo stačiakampio funkcijos tuo, kad signalas dauginamas iš trikampio lango (2.6 pav.). Pagal 11 formulę apskaičiuojama $w(n)$ (Kamarauskas, 2009; Aparna, 2017; Poularikas, 1999):

$$w(n) = 1 - n \vee \frac{|n|}{N-1} \quad (11)$$



2.7 pav. Blackmano lango funkcija (Poularikas, 1999)

6. **Blackmano lango funkcija** pagal 12 formulę gauna geresnius rezultatus su įvairiais dažniais, nes slopina aukštas šonines skiltis ir turi platesnį pagrindinį skilties plotį (2.7 pav.)(Aparna, 2017; Poularikas, 1999).

$$w(n) = 0.42 + 0.5\cos\left(\frac{2\pi}{N}n\right) + 0.08\cos\left(\frac{4\pi}{N}n\right), \text{ kurn} = \frac{-N}{2}, \dots, -1, 0, 1, \dots, \frac{N}{2}; \quad (12)$$

Lango funkcijos daugiausiai skirtos skaitmeniniam filtrui ir pasirinkus tam tikrą intervalą nuo kadro iki kadro grąžinama ne nulinė baigtinė vertė tame intervale ir nulinė už to intervalo ribų. Tik po visų šių pirminio apdorojimo etapų signalas yra paruoštas tolesniems skaičiavimams. Tačiau net po pirminio signalo filtravimo, kadravimo ir langavimo, jis nėra tinkamas geriausiam rezultatui gauti. Kitas veiksmas, reikalingas geriams garso įrašo požymiams gauti, yra segmentavimas.

2.3. Signalo segmentavimas

Segmentavimas yra vienas iš balso atpažinimo sistemos kūrimo etapų. Jis leidžia surasti ribas žodžiuose, fonemose ar skiemenyse, net ir natūraliojoje kalboje, jei tinkamai nustatyti parametrai. Dažniausiai segmentavimas naudojamas duomenų išmokymo etape atliekant kalbos identifikavimą, duomenų bazių kūrimą ir kalbos sintezėje. Segmentavime žodis yra natūralus vienetas, fonema yra mažiausias signalo vienetas, o jų derinys sudaro pagrindinį kalbos vienetą – skiemenį. Yra du segmentavimo tipai – rankinis ir automatinis. Rankinis segmentavimas turi daugybę netikslumų ir trūkumų. Jis reikalauja nemažai laiko, atliekamas tik su specialistu fonetiku ir yra nenuoseklus. Automatinis tipas yra patogesnis ir tikslesnis metodas, suskirstantis kalbą į akustiškai apibrėžtus žodinius vienetus. Automatinis „kalbos detektorius“ gali būti klasifikuojamas priklausomai nuo požymių ir nuo požymių klasifikatoriaus. Patys populiariausi ir dažniausiai naudojami skaičiavimo algoritmai yra aklas ir pagalbinis segmentavimas (Kamarauskas, 2009; Sakran, 2017).

Taikant aklo segmentavimo metodą, iškart nėra žinoma iš anksto egzistuojančių arba išorinių žinių apie bet kokią kalbinę informaciją. Atliekant pirmąjį šio segmentavimo skaičiavimo veiksmą dėl kalbinių žinių trūkumo viskas priklauso nuo akustinių savybių. Tada naudojamas „iš apačios į viršų“ apdorojimas. Ši procedūra nustato optimalų subžodžių segmentų skaičių konkrečiame kalbos pavyzdyje ir taip pat nustato šių subžodžių segmentų ribas. Jis dažniausiai taikomas kalbos identifikavimo programose ir garsiakalbio patikros sistemose (Kamarauskas, 2009; Sakran, 2017).

Vienas iš lengviausių ir populiariausių metodų, naudojamų segmentavime, yra triukšmo slenksčio nustatymas. Jį naudojant gaunamas rezultatas ne visada tikslus ir labai priklauso nuo įrašo kokybės. Jam apskaičiuoti naudojamas tam tikras triukšmo signalo ilgis N_{tr} , kuris įstatomas į 13 formulę (Kamarauskas, 2009; Sakran, 2017).

$$Thr = \frac{1}{N_{tr}} \sum_{n=1}^{N_{tr}} |s_{tr}(n)| \quad (13)$$

Pagalbinis segmentavimas naudoja tam tikrą kalbinę informaciją, siekiant sujungti ją su norimo tipo segmentu ir atskirti žmogaus kalbą nuo nereikalingo triukšmo. Dažniausiai atpažįstant kalbą jis taikomas duomenų mokymo etape su tokiais klasifikatoriais kaip *PMM*, *DNT* ir *DLK*. Siekiant, kad tinkamai veiktų ir būtų gautas geras rezultatas, yra svarbu, kad kalbos išrinkimas gautų tik kalbos signalo požymius, o ne triukšmo (Kamarauskas, 2009; Sakran, 2017).

Segmentavimo skaičiavime taikomos laiko ir dažnio domeno savybės. Naudojant laiko domeno savybes, skaičiuojama trumpalaikė signalo energija ir vidutinis nulinio kirtimų dažnis. Naudojant trumpalaikę signalo energiją, skaičiuojama pagal 14 formulę, kai signalas siunčiamas tam tikrą laiką, lyginant jo mėginius ir imant vidurkį (Kamarauskas, 2009; Sakran, 2017).

$$E_n = \frac{1}{N} \sum_{m=-\infty}^{\infty} [x(n-m)w(m)]^2 \quad (14)$$

Vidutinis nulinio kirtimų dažnis apskaičiuoja, kiek kartų tam tikrame kadre mėginiai keičia savo algebrinį ženklą arba, kitaip tariant, kiek kartų signalo amplitudė eina per nulinę vertę (15 formulė).

$$Z_n = \frac{1}{2} \sum_{m=-\infty}^{\infty} |sgn[x(n-m)] - sgn[x(n-m-1)]| w(n) \quad (15)$$

Dažnio domeno savybės skaičiuojamos, naudojant greitosios Furjė transformacijos reprezentaciją, kuri rodo signalo spektrinę sudėtį. Spektrui apibūdinti apdorojant garsą taikomas matas, vadinamas spektriniu centroidu. Jis parodo, kurioje vietoje spektro aukščio

centras pasiekia maksimalią reikšmę, atitinkančią ryškesnį signalą. i kadro spektrinis centroidas skaičiuojamas pagal toliau pateiktą 16 formulę (Kamarauskas, 2009; Sakran, 2017).

$$SC_i = \frac{\sum_{n=0}^{N-1} f_i(n)x_i(n)}{\sum_{n=0}^{N-1} x_i(n)} \quad (16)$$

Pagal ją $f_i(n)$ yra i kadro centrinis dažnis GFT spektre, o $x_i(n)$ yra svertinis dažnio dydis.

Kitas metodas, kuris skaičiuojamas, naudojant dažnio domeno savybes, yra spektrinis srautas. Jis apibrėžia greitį, per kurį keičiasi garso galios spektras, kai lyginamas vieno kadro galios spektras su praėjusiu kadro galios spektru. Be to, apskaičiuojant šią reikšmę, galima nustatyti signalo trukmę. Spektrinis srautas skaičiuojamas pagal 17 formulę, kur $X_i(k)$ yra i -ojo trumpalaikis kadras, kurio ilgis N , pateikiamas lygtyje GFT k koeficientas (Kamarauskas, 2009; Sakran, 2017).

$$SF_i = \sum_{k=1}^{N/2} (|X_i(k)| - |X_{i-1}(k)|)^2 \quad (17)$$

Garso segmentavimas yra svarbi žmogaus balso identifikavimo dalis. Jis reikalingas, siekiant gauti kuo tikslesnius rezultatus po išmokymo etapo iš kalbos signalo požymių ir jų palyginimo. Tiek aklas, tiek pagalbinis segmentavimas leidžia išvengti nereikalingų signalo požymių, tokių, kaip nepalankiomis sąlygomis įrašytas garso signalas. Jei „kalbos detektorius“ bus netinkamai sukonfigūruotas, padidės klaidų tikimybė. Todėl prieš išskiriant požymius, labai svarbu tinkamai atlikti segmentavimo etapą.

2.4. Signalų požymių išskyrimas

Kitas kalbos atpažinimo etapas po segmentavimo yra garso signalo požymių išskyrimas. Šioje darbo dalyje bus aprašyti melų skalės keistro koeficientai, diskrečiosios kosinuso transformacijos skaičiavimai ir požymiai, gaunami ir siunčiami į klasifikatorių. Kalbos signalo požymiai yra esminis dalykas, iš kurio bus nustatytas žmogaus balso savininkas. Po pirminio signalo apdorojimo ir segmentavimo eina melų filtro apdorojimas, kurio metu apdorojamas garso įrašas, gaunant iš jo reikalingus signalo bruožus.

Pats žodis „mel“ reiškia garso aukščio vienetą, kurio pagrindas – žmogaus klausos organų garso suvokimas. Žmogaus ausies dažnio atsakas nėra tikslus parametras, todėl ir buvo sugalvoti empiriškai atrinkti garsumo vienetai. Tai reiškia, kad tikslumas nėra šimtu procentų, bet naudojant dažnio domeno savybes galima gauti tikslesnį rezultatą (Kamarauskas, 2009; Dave, 2013; Athira, 2017; Ramljak, 2013; Bhattacharjee, 2013; Singh, 2014).



2.8 pav. Melų skalė (Athira, 2017).

2.8 paveiksle matoma, kaip logaritmiškai keičiasi dažnis. Jis buvo išvestas iš sinusoidinio signalo, atliekant eksperimentus. Tokia melų skalės priklausomybė nuo dažnio gali būti išreikšta pagal 18 formulę, kur ji išgaunama naudojant apskaičiuotą Furjė transformaciją ir lango funkcijos reikšmę (Bhattacharjee, 2013).

$$B(f) = 1127.01 \ln(1 + f/700) \quad (18)$$

Melų skalės kepstro koeficientų skaičiavimo eigą galima suskirstyti į keletą etapų (Ramljak, 2013):

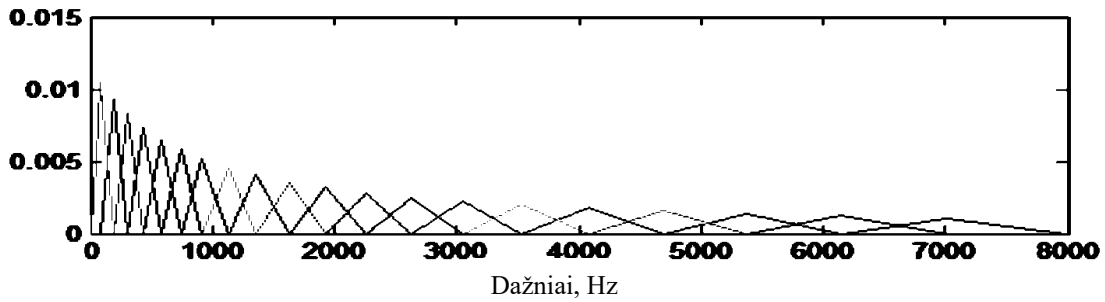
1. Furjė transformacijos panaudojimas.
2. Trikampio filtro skaičiavimas, kuris apibūdina spektro galias melų skalėje.
3. Paimama kiekviena rasta spektro galia iš kiekvieno melo dažnio.
4. Diskrečiojo kosinuso transformacijos taikymas melų spektro konversijai į laiko domeną. Gauta *MSKK* yra spektro amplitudė.

Kaip buvo minėta pirmiau, siekiant gauti dažnio domeno savybes, reikia panaudoti Furjė transformaciją. Furjė transformacija laikoma viena iš geriausių signalo manipuliavimo būdų. Ji gali būti naudojama naujiems signalo skambėjimo efektams gauti, taip pat triukšmui mažinti arba labai tiksliai jį išlyginti. Apdorojant signalą *FFT* filtras padeda lengvai padidinti arba sumažinti tam tikrus dažnius, nes sugeba greitai analizuoti informaciją. Kitaip tariant, tai yra daug tikslesnis išlyginimo tipas. Šis efektas padeda sukurti plačios aukšto arba žemo dažnio juostos filtras, žiedų filtras arba siauros dažnio juostos filtras. Jo veikimo principas yra N laiko srities taško išskaidymas į N laiko domeno signalus. Tada atliekamas N dažnio spektrų apskaičiavimas, atitinkantis to N laiko domeno signalus. Paskutinis žingsnis apskaičiuojant *FFT* yra N spektrų sintezavimas į vieną dažnių spektrą. Prieš *FFT* atliekant žmogaus balso

apdorojimą pirmiausiai reikia užrašyti garso įrašo signalą į diskreitinį $x[n]$, $0 \leq n < N$. Tik po šio veiksmo geriausiai taikyti *FFT*, apskaičiuojamą taip (Kamarauskas, 2009; Dave, 2013; Athira, 2017; Ramljak, 2013; Bhattacharjee, 2013; Singh, 2014):

$$X_a[k] = \sum_{n=0}^{N-1} x[n] e^{\frac{-2\pi i}{N} kn}, 0 \leq k < N \quad (19)$$

Kitas veiksmas yra trikampio filtro skaičiavimas, sukurtas pagal netiesinę dažnių skalę (2.9 pav.).



2.9 pav. Normalizuotas trikampio filtras (Singh, 2014)

Su langų funkcija atliekamas filtrų derinimas, apibrėžiant m -ąjį filtrą:

$$H_m = \begin{cases} 0 & k < f[m-1] \\ \frac{(k-f[m-1])}{(f[m]-f[m-1])} f[m-1] & f[m-1] \leq k < f[m] \\ \frac{(f[m+1]-k)}{(f[m+1]-f[m])} f[m] & f[m] \leq k < f[m+1] \\ 0 & k \geq f[m+1] \end{cases} \quad (20)$$

Šis trikampio filtras suranda vidurkinį spektrą aplink kiekvieną centrinį dažnį, kurio dažnio juosta platėja. Jos $f[m]$ dažnis gaunamas iš pateiktos melų skales, skaičiuojant pagal 21 formulę.

$$f[m] = \left(\frac{N}{F_d}\right) B^{-1} \left(B(f_1) + m \frac{B(f_h) - B(f_1)}{M+1} \right) \quad (21)$$

Pagal pateiktą formulę N yra greitosios Furjė transformacijos dydis, F_d – diskretizacijos dažnis, B^{-1} yra melų į hercus transformacija ir f_h ir f_1 – tai aukščiausio ir žemiausio filtro dažnis. Kitas veiksmas yra energijos gavimas kiekvienam kadrui, taikant 22 formulę, kur

$$S[m] = \ln(\sum_{k=0}^{N-1} |X_a[k]|^2 H_m[k]), 0 \leq m < M \quad (22)$$

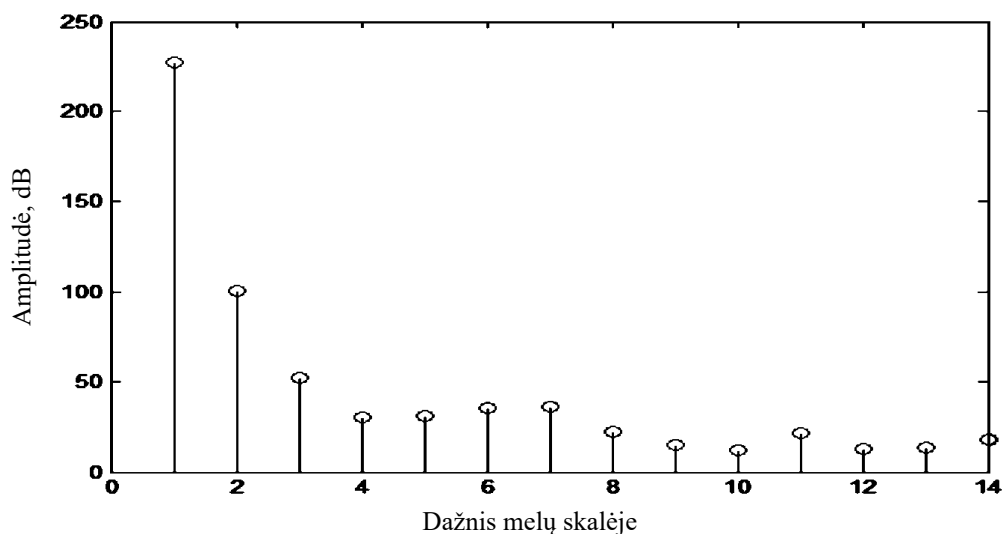
Šioje formulėje naudojama garso įrašo Furjė transformacijos koeficientų amplitudė $X_a[k]$.

Paskutinis etapas, siekiant gauti melų skalės kepstro koeficientus, yra diskrečiosios kosinusų transformacijos taikymas. Tai yra realius skaičius naudojanti Furjė transformacija,

labiau susitelkianti į žemesnius indeksus. Tokia duomenų transformacija suteikia galimybę aproksimuoti garsą su mažiau koeficientų skaičiumi. Apdorojant *DKT* garsą, apibrėžiamas melų arba barkų spektro duomenų pavertimas į laiko domeną (2.10 pav.). Šios lygties išraiška gali būti užrašyta pagal 23 formulę (Kamarauskas, 2009; Dave, 2013; Athira, 2017; Ramljak, 2013; Bhattacharjee, 2013; Singh, 2014).

$$c[n] = \sum_{m=0}^{M-1} S[m] \cos\left(\frac{\pi n(m-\frac{1}{2})}{M}\right), 0 \leq n < M \quad (23)$$

Tokiu būdu gaunamas melų skalės kepstro koeficientai. Šių koeficientų derinys vadinamas akustiniais vektoriais, o jų rinkinys – akustinių vektorių seka.



2.10 pav. *MSK* koeficientai po *DKT* skaičiavimo (Singh, 2014)

MSKK yra naudingas požymių išskyrimo metodas, leidžiantis pašalinti daug nereikalingų komponentų originaliame spektre. Dažniausiai identifikuojant žmogaus kalbą galima sumažinti vieno laiko langą iki 13 koeficientų, kurių kiekvienas daro didelę įtaką galutiniam spektrui (Kamarauskas, 2009; Dave, 2013; Athira, 2017; Bhattacharjee, 2013; Singh, 2014).

MSKK yra efektyvesnė ir patikimesnė technika garso požymiams išskirti. Analizuojant visus gautus duomenų rinkinius, galima juos perduoti išmokymo modeliams ir tolesniam lyginimui.

2.5. Signalo modelio sukūrimas ir lyginimo metodai

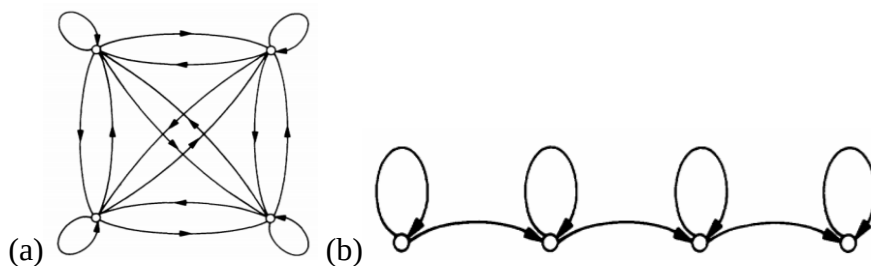
Šioje dalyje bus aptarta, kaip iš *MSKK* gautus požymius naudoja parinkti klasifikatoriai. PMM ir giliojo mokymosi metodas taiko šiuos požymių vektorius ir atlieka

išmokymo bei lyginimo procesą. Jie buvo parinkti, nes kiekvienas iš jų yra populiarus ir dažnai naudojamas daugelyje programų klasifikatorių. Jų identifikavimo tikslumas, palyginti su kitais, yra aukštesnis, o išmokymo procesas naudoja paslėptus vienetus. Pirmiausiai bus aprašytas klasikiniu klasifikatoriumi tapęs paslėptasis Markovo modelis.

2.5.1 Paslėptasis Markovo modelis

Paslėptasis Markovo modelis yra stochastinis, statistinis Markovo modelis su nematomais nariais, kurie sujungti perėjimais tarpusavyje. Vienas iš esminių šio metodo pranašumų yra tai, kad jis veiksmingai modeliuoja laike kintančias spektrines vektorines sekas. Be to, kadangi PMM nariai yra akustiniai modeliai, jie nurodo, kokie žmogaus balso garsai bus girdimi per tam tikrus segmentus, o tai žymiai priartina prie tinkamo rezultato gavimo (Deng, 1994; Juang, 1991; Stamp, 2018; Trivedi, 2014; Gales, 2007). Yra kelios PMM struktūros (2.11 pav.):

1. Ergodinis arba visiškai sujungtas modelis. Kiekviena tokio modelio tipo būseną pasiekama iš bet kurios kitos būsenos per vieną veiksmą. Tačiau dėl tokio didelio perėjimo skaičiaus naudingi atpažinti duomenys gali būti lengvai prarasti (Cuzzocrea, 2017).
2. Grandinės modelis. Šio tipo būsenos eina nuosekliai viena paskui kitą ir yra artimiausios kalboje vykstantiems procesams. Toks modelis dar vadinamas kairės-dešinės modeliu ir dažniausiai naudojamas daugelyje atpažinimo sistemų.



2.11 pav. Ergodinis (a) ir grandininis (b) PMM modeliai

PMM grandininis modelis yra vektorinių požymių lyginimo metodas, per kurį parametrai priskiriami tam tikrai klasei ir sudaro tų klasių seką. PMM stebimų įvykių seka gali būti aprašyta su $S = \{S_1, S_2, S_3, \dots, S_N\}$, kur N yra vektorių skaičius. Nario būseną laiko intervale keičiasi pagal atsitiktinę tikimybę ir faktinę laiko būseną, kuri gali būti išreikšta $q =$

$\{q_1, q_2, \dots, q_t\}$. Be to, atsižvelgiant į šiuos procesus, formos perėjimo tikimybių $A = \{a_{ij}\}$ rinkinys skaičiuojamas pagal formulę (Deng, 1994; Juang, 1991; Trivedi, 2014; Gales, 2007):

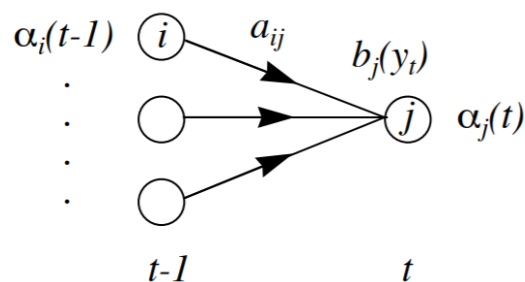
$$A = \{a_{ij}\} = P[q_t = S_i | q_{t-1} = S_j], \quad (24)$$

kur $a_{ij} \geq 0$, $\sum_{j=1}^N a_{ij} = 1$. Tokiame PMM procese rezultatas yra kiekvieno laiko momento stebimų įvykių seka, dar vadinama stebimu Markovo modeliu. Be šių bendrų paslėptųjų Markovo elementų, kaip buvo minėta pirmiau, dar yra pradinės būsenos pasiskirstymas, apskaičiuojamas pagal 5 formulę, ir stebėjimo simbolio tikimybės išdėstymo matrica, išreiškiama pagal 4 formulę. Tik sujungus visus šiuos elementus, galima sukurti modelio stebėjimo seką.

Vienas iš algoritmų, išpopuliarinusių PMM metodą, yra modelio parametrų įvertinimo Baumo ir Velčo algoritmas. Pagal Baumo ir Velčo skaičiavimą parametrų rinkiniai gaunami iš kiekvieno sudarančio fonetinio vieneto. Vykstant atpažinimui, nustatoma didžiausia ieškomo nežinomo žodžio tikimybė ir išmokyto modelio požymių panašumas. Tikimybių radimas ir kaupimas vyksta atliekant iteracinį „forward – backward“ skaičiavimą. Jį naudojant modelyje sukauptos N paeiliui einančių žodžių statistikos (Deng, 1994; Juang, 1991; Schickhofer, 2017).

Yra trys pagrindiniai algoritmai, naudojami PMM ir atliekantys tam tikrus skaičiavimus garsui atpažinti: tiesioginis, Viterbio ir jau minėtas „forward – backward“ algoritmas (Deng, 1994; Juang, 1991; Trivedi, 2014; Gales, 2007; Schickhofer, 2017).

Tiesioginis algoritmas skirtas garso atpažinimui tik izoliuotam žodžiui atlikti. Jo veikimo principo esmė yra tikimybių įvertinimas, kad tam tikras žodinis modelis sukūrė tam tikrą stebėjimo seką. Pagal gautus rezultatus galima nustatyti kiekvieno pateikto žodžio modelio balus. Algoritmas dėl PMM nepriklausomumo taisyklių gali atlikti skaičiavimus rekursyviai (2.12 pav.)(Deng, 1994; Juang, 1991; Schickhofer, 2017).



2.12 pav. Tiesioginio leidimo rekursija (Schickhofer, 2017)

Tikimybė $a_j(t)$ skaičiuojama pagal 25 formulę, kurioje t yra laikas, per kurį eina j būseną, y_t išvesties simbolis, o kartu su $b_j()$ sudaro stebėjimo simbolio būklės stebėjimo tikimybę, atsižvelgiant į dabartinę padėtį, a_{ij} – perėjimo tikimybė (Schickhofer, 2017).

$$a_j(t) = \sum_i a_i(t-1)a_{ij}b_j(y_t) \quad (25)$$

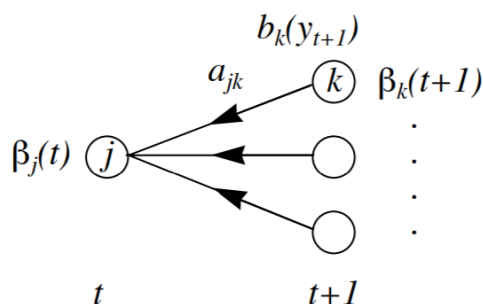
Tokia skaičiavimo strategija dar vadinama dinaminio programavimu, kuris padeda apskaičiuoti brutalaus jėgos įvertinimo tikimybę. Paskutinis veiksmas yra stebėjimo sekos tikimybės $P(O|\lambda)$ nustatymas, kuris lygus $\sum a_j(t)$. Šio algoritmo trūkumas yra tas, kad jį naudojant neįmanoma gauti PMM kiekvienai atskirai galimai sekai (Schickhofer, 2017).

Viterbio algoritmas taikomas nuolatinei kalbai atpažinti. Šis skaičiavimo būdas yra panašus į tiesioginį algoritmą, išskyrus tai, kad randamas maksimalus elemento įvertinimas vietoj kiekvieno. Jo dekodavimo atveju paieškos erdvė apribojama, palaikant tik keletą tikimybių lygiagrečiai. Tai padidina gero rezultato gavimą ir suteikia galimybę naudoti ne atskirus žodžius, o visą žmogaus kalbą. Tačiau vienas iš šio metodo trūkumų yra vykdymo laiko kontroliavimas, kuris gali būti sunkiai valdomas (Deng, 1994; Juang, 1991; Gales, 2007; Schickhofer, 2017). Norint apskaičiuoti Viterbio algoritmą, reikia sekti kiekvieną geriausią pirmtakio būseną matricos elemente, randant maksimaliausią elementą pagal 26 formulę (Schickhofer, 2017).

$$v_j(t) = \text{MAX}_i[v_i(t-1)v_{ij}b_j(y_t)] \quad (26)$$

Paskutinis iš dažniausiai naudojamų PMM algoritmų yra „forward – backward“ algoritmas, skirtas PMM modeliui išmokyti. Jį naudojant įmanoma surasti mėginių logaritminę tikimybę ir įvertinti PMM nenustatytus parametrus. „Forward – backward“ algoritmas yra vienas iš įvertinimo maksimizavimo metodų. Jis padeda išspręsti PMM problemą, kai nėra uždaros formos sprendimo. Jo atgalinio termino $\beta_j(t)$ skaičiavimas yra panašus į tiesioginio algoritmo formulę, bet veikia atvirkštiniu būdu (27)(2.13 pav.)(Deng, 1994; Juang, 1991; Gales, 2007; Schickhofer, 2017).

$$\beta_j(t) = \sum_k \beta_k(t+1)a_{jk}b_k(y_{t+1}) \quad (27)$$



2.13 pav. Atgalinio leidimo rekursija (Schickhofer, 2017)

Sujungiant tiesioginio ir atgalinio algoritmų skaičiavimus galima gauti tikimybės formulę, kad $P(O|\lambda) = \sum \alpha_j(t)\beta_j(t)$. Taip randama mokymo procese didžiausia PMM parametrų tikimybė ir galinis režimas. Po šio požymio vektoriais mokymo proceso galinių klasifikatoriaus rezultatų tikslumas gaunamas labai aukštas ir pakankamai stabilus (Deng, 1994; Juang, 1991; Gales, 2007; Schickhofer, 2017).

PMM garso atpažinimo klasifikatorius buvo pavadintas paslėptuoju, nes ne visos jo įvykių sekos yra matomos. Vidinė procesų seka gali būti stebima tik per atsitiktinius išorinio stebėjimo procesus. Tokia modelio struktūra leidžia laisvai valdyti kintamo ilgio įvestį ir tiesiogiai vykdyti mokymo procesą iš neapdorotų sekos duomenų. Tačiau netgi toks klasikinis ir plačiai naudojamas klasifikatorius turi savo apribojimų ir sunkumų norint gauti gerą rezultatą. Pavyzdžiui, PMM vienu metu gali nagrinėti tik vieną kalbą ir visiškai sujungta perėjimo sistema gali perkrauti programą. Be to, gali atsirasti akustinių duomenų diskriminacija dėl maksimalaus tikimybės mokymo. Nepaisant visų šių problemų su PMM, jis vis tiek lieka vienu iš geriausių garso atpažinimo sistemų klasifikatorių. Jo stiprus statistinis pagrindas, gebėjimas dirbti kartu su kitomis bibliotekomis ir aukšto tikslumo rezultato gavimas žymiai išskiria jį iš kitų duomenų palyginimo metodų. Kita vertus, kitas neseniai pradėtas plačiai taikyti atpažinimo sistemos klasifikatorius DNT gali visai lygiavertiškai ir net kai kuriais atvejais geriau veikti negu PMM.

2.5.2 Giliojo mokymosi metodas

Paskutiniaisia dešimtmečiais giliojo mokymosi dirbtinių neuroninių tinklų metodo taikymas daugelyje atpažinimo sistemų padidėjo dėl šio lyginimo proceso ir kompiuterinių technologijų tobulėjimo. DNT apdorojimo įrenginių skaičius labai panašus į žmogaus smegenų neuronus ir jų veikimo principą. Visus procesus jame atlieka požymių vektorių vienetai, skaičiuojant vietinių įvesčių skaliarinę funkciją ir perduoda duomenis kitiems savo

kaimyniniams vienetams. Kaip ir PMM klasifikatoriuje DNT vienetai skirstomi į tris grupes (Hinton, 2012; Yu, 2010; Vijoyo, 2011; Tebelskis, 1995; Schickhofer, 2017):

1. Įvesties vienetai – tai vienetai, gaunantys duomenis.
2. Paslėpti vienetai – skirti transformuoti duomenų atvaizdavimą.
3. Išvesties vienetai – parodo sprendimus.

Visi šitie tinklo vienetai gali būti struktūrizuoti pagal topologijos rūšį – svorį. Vieneto svoris, dar vadinamas jėga, apibūdinamas kaip reikšmė, koduojanti tinklo trumpalaikę atmintį. Jo vertė lygi, kiek vienas neuronas turi įtakos savo „kaimynams“. Teigiamas svoris sužadina neuronus „kaimynus“, o neigiamas – blokuoja.

Dirbtiniai neuroniniai tinklai turi daugybę skirtingų struktūrų, bet dažniausiai taikomi nestruktūriniai, moduliniai, sluoksniuoti ir pasikartojantys. Kiekviena iš šių DNT rūšių skirta tam tikrai užduočiai atlikti. Nestruktūriniai tinklai skirti modeliams saugoti ir naudoja dvipusę ryšį, o kiti naudoja tik vieną kryptį. Modeliniai tinklai gali būti integruojami į bet kokią kitą tinklo rūšį. Jie išskirtiniai tuo, kad iš paprastų komponentų sukuria sudėtingas sistemas. Sluoksniai sujungia vieną modelį su kitu ir gali kartotis kaip ir pasikartojantys tinklai. Patys pasikartojantys tinklai skirti tinklo modeliavimui sekti. Paties tinklo apribojimas yra gera praktika, gaunant geresnius rezultatus. Išmokymo etape neribojant DNT, jis mokysis iš jau išmokytų duomenų (Hinton, 2012; Yu, 2010; Vijoyo, 2011; Tebelskis, 1995; Schickhofer, 2017).

DNT skaičiavimas pradedamas nuo požymių vektorių pateikimo mokymo modeliui. Jame vyksta vienetų svorių modifikavimas ir jungčių pritaikymas visiems įvesties duomenims taip, kad tinklo elgesys atitiktų pageidaujamą skaičiavimą. Paprasčiausiam svorių skaičiavimui, kai reikia surasti įvesties vektorių seką išvesties vektoriams, taikoma 28 formulė, kur x yra įvestas požymio vektorius, t yra tinklinis vektorius ir b yra šablono indeksas (Hinton, 2012; Vijoyo, 2011), (Tebelskis, 1995).

$$w_{ji} = \sum_b \frac{x_i^p t_j^p}{\|x^p\|^2} \quad (28)$$

Kaip buvo minėta pirmiau, yra keli mokymosi tipai – mokymasis su mokytoju, mokymasis be mokytojo ir hibridinis mokymas. Šie mokymo procesai pagal įvesties vienetų, svorio vertes ir tinklo topologiją išmoko modelį taip, kad lyginimo etapo metu būtų gaunama tinkama rezultato tikimybė.

Per mokymosi su mokytoju procesą iškart ištaisomos tinklo klaidos, nes aišku, koks

turi būti gautas išvesties rezultatas. Šis procesas, kaip jau buvo minėta, naudoja „feedforward“ tinklą savo skaičiavimuose ir vienasluoksnius arba daugiasluoksnius perceptronus. Jo išvesties vertės skaičiuojamos kiekvienam įvesties vektoriui ir yra tiesiogiai susijusios su išvesties reikšme, todėl lengvai galima taikyti delta taisyklę (Hinton, 2012; Vijoyo, 2011; Tebelskis, 1995).

Delta arba Vidrov-Hoff taisyklės esmė yra sustiprinti ryšį tarp dviejų vienetų. Ją naudojant sumažinama santykinė paklaida, todėl didesnė tikimybė, kad nustatytas svoris atitiks norimą rezultatą. Daugiasluoksniame tinkle delta taisyklė yra mokymosi proceso pagrindas. Jis skaičiuojamas pagal 29 formulę (Tebelskis, 1995).

$$\Delta w_{ji} = \varepsilon x_i (t_j - x_j) \quad (29)$$

Kai tinklo išvesties reikšmės nėra žinomos, taikomas požymių išmokymo modelis be mokytojo. Per šį procesą požymio vektoriai suskirstomi į klases ir ieškoma panašumų tarp mokymo aibės įvesčių arba jų koreliacijų. Yra keli būdai, kaip galima realizuoti skirtingas neuroninių tinklų topologijas šiame mokymo procese (Hinton, 2012; Vijoyo, 2011; Tebelskis, 1995):

1. Varžytinių mokymas.
2. Hebbo mokymas.
3. Saviorganizuojantis mokymas.

Varžytinių mokymo eigoje įvesties duomenys grupuojami ir kvantuojami, siekiant juos klasifikuoti pagal tam tikrus parametrus. Svorio vektoriai perkeliama į hipersferinius įvesties grupių centrus, tokiu būdu sumažinamas atstumas. Jo mokymo proceso skaičiavimo taisyklė yra (Tebelskis, 1995):

$$\Delta w_{j'i} = \varepsilon * (x_i - w_{j'i}), \quad (30)$$

kur j' yra laimėjimo išvestis. Varžytinių mokymo trūkumas yra tas, kad kai kurie svorio vektoriai gali būti per toli vienas nuo kito ir jų mokymas tampa neįmanomas. Tokiu atveju reikia taikyti papildomas priemones arba skaičiavimus. Pavyzdžiui, vienas iš būdų, kaip galima išvengti šio trūkumo, yra svorių inicijavimas taip, kad jie galėtų atitikti įvesties mėginius.

Kitas mokymosi be mokytojo tipas yra Hebbo mokymas, kurio tikslas, kai išvesties aktyvacijos yra susietos, sustiprinti ryšį tarp dviejų vienetų. Jis labiausiai tinka linijiniams vienetams mokyti, suglaudinant jų išvesties sluoksnio įvestį. Hebbo taisyklė išreiškiama pagal 31 formulę (Hinton, 2012; Vijoyo, 2011; Tebelskis, 1995).

$$\Delta w_{ji} = \varepsilon * x_i * x_j \quad (31)$$

Šis mokymo procesas kai kuriais atvejais reikalauja pakeitimų skaičiavimuose, kad svorio vertės nebūtų per didelės. Tada papildomai taikomos Linkerio arba Sangerio taisyklės, apribojančios svorio reikšmes ir neleidžiančios jiems užtekti už ribų (Tebelskis, 1995).

Paskutinis ir dažniausiai daugelyje garso atpažinimo su DNT sistemų naudojamas mokymo metodas yra saviorganizuojantis mokymas. Šio tinklo mokymosi tikslas – išlaikyti vektoriinių vienetų topologiją ir rekonstruoti išvesties sluoksnio įvesties modelius, siekiant išsaugoti kuo daugiau požymių informacijos paslėptame sluoksnyje. Mokymosi šiuo būdu pradžia – vektorių komponentių atsitiktinis nustatymas ir skirtingų n -mačių vektorių pateikimas. Tada eina jų lyginimas ir atstumo tarp vektoriaus ir neurono skaičiavimas. Mažiausias atstumas tarp jų vadinamas neuronų nugalėtoju. Kai randamas neuronų nugalėtojas, naudojama iteracinė su kaimynystės funkcija formulė, keičianti visų tinklo neuronų komponentes. Taip vyksta visas mokymo procesas, per kurį nustatomi neurono nugalėtojai ir sudedami į pirmosios eilės kaimynus. Jeigu neuronas nugalėtojas yra šalia neurono su mažiausiu atstumu nuo jo, tada rezultatas užrašomas kaip 0, priešingu atveju – 1. Toks mokymo procesas dar vadinamas tinklo kodavimu. Duomenų matmenys mažinami, o paslėpti sluoksniai tampa suspaustu įvesties vektorių atvaizdu. Šio mokymo proceso pranašumai yra lengvas veikimo supratimas ir tiesioginiai atsakai. Tačiau jis turi nemažai trūkumų, pavyzdžiui, mokymo laiko pailgėjimas nerealiame laiko mastelyje ir jis turi blogas apibendrinimo savybes (Hinton, 2012; Yu, 2010; Vijoyo, 2011; Tebelskis, 1995; Schickhofer, 2017).

Be tiesioginio sklidimo mokymo, yra ir atgalinio sklidimo metodas, kuris taip pat vadinamas apibendrinta delta taisykle. Jis laikomas daug greitesne mokymo procedūra, jei tinkamai reguliuotas svorių dydis ir mokymosi tempas. Pirmasis žingsnis atliekant jo skaičiavimą yra tikrųjų tinklo atsakų radimas iš atsitiktinai parinktų mažo dydžio ryšių ir slenksčio svorių reikšmių. Reikšmių vektorius dedamas į įvesties sluoksnį, o norimo atsako pavyzdys – į DNT išvesties sluoksnį. Tada reikia didinti tinklo jautrumą atgaliniu būdu per visą tinklą ir naudojant rekursinį algoritmą skaičiuojamas ryšių svorių pokytis pagal 32 formulę (Hinton, 2012; Vijoyo, 2011; Tebelskis, 1995).

$$W_{ij}(t + 1) = w_{ij}(t) - a\delta_j x'_i, \quad (32)$$

kur a yra konstanta, δ_j – klaida, kurios skaičiavimas priklauso nuo to, kurioje vietoje yra j -tasis neuronas, ir x'_i – i -tojo neurono įvesties vertė. Vieneto poslinkis skaičiuojamas panašiai, bet be i -tojo neurono įvesties reikšmės.

$$p_{ij}(t + 1) = p_{ij}(t) - a\delta_j \quad (33)$$

Šie mokymo proceso skaičiavimo veiksmai vyksta tol, kol klaida tampa lygi nuliui arba yra patenkinamai maža. Atgalinio sklidimo mokymas gali išmokyti daug daugiau ir bet kokio sudėtingumo netiesinio sąryšio duomenis. Tačiau, jei jo veikimo principas nustatytas netinkamai, mokymo laikas gali pailgėti. Pagrindinis šio mokymo proceso trūkumas yra negebėjimas veikti realiu laiku (Hinton, 2012; Yu, 2010; Vijoyo, 2011; Tebelskis, 1995).

Hibridinio mokymo proceso veikimo principai – kaip ir mokymosi su mokytoju bei mokymosi be mokytojo. Dalis tinklo nustato svorius pagal vieną metodą, o kita dalis – pagal kitą. Dažniausiai jis taikomas kartu su atgalinio sklidimo mokymu. Kartu jie pasiskirto darbą taip, kad žemesniame tinklo sluoksnyje duomenys kaupiami, o tada aukštesniame susiejami su norimais išvesties modeliais. Toks požymių mokymo procesas yra greitesnis, bet tikslumas yra prastesnis negu kitų tipų mokymuose (Hinton, 2012; Yu, 2010; Vijoyo, 2011; Tebelskis, 1995).

Taip vyksta mokymo procesai DNT klasifikatoriuje. Tolesni veiksmai yra teisingas gautų duomenų rinkinio klasifikavimas pagal tam tikras kategorijas, kad vėliau būtų galima palyginti juos su nežinomo ieškomo signalo požymiais. Duomenų klasifikavimas vyksta taikant Bajeso taisyklę, kuris naudoja tikimybės pasiskirstymą siekiant gauti aposteriorinę tikimybę (Tebelskis, 1995).

$$P(c_i|x) = \frac{P(x|c_i)*P(c_i)}{\sum_i P(x|c_i)*P(c_i)} \quad (34)$$

Pagal Bajeso taisyklės formulę (34) c_i yra klasės, o x yra įvesties vektorius. Jį naudojant galima sužinoti aposteriorinę tikimybę, paprastai lyginant $P(x|c_i)P(c_i)$ vieną su kitu. Jeigu vienas iš jų turi didesnę reikšmę, jis priskiriamas kitam duomenų rinkiniui. Tokiu metodu mažinamas klasifikacijos klaidų lygis (Tebelskis, 1995).

DNT klasifikatorius po daugelio metų tampa vienu iš prastesnių ir nepatikimesnių atpažinimo metodų, vienu iš populiariausių ir dažniausiai naudojamų dėl algoritmų ir technologijų patobulinimo paskutiniaisiais dešimtmečiais. Jo tinkamo rezultato tikimybės gavimas tapo aukščiausio lygio ir gali konkuruoti su tokiais klasikiniiais klasifikacijos metodais kaip PMM. Savo netiesinio modeliavimo galimybe, gerai apmokyto modelio savybėmis DNT gali būti net pranašesnis už paslėptąjį Markovo modelį. Tačiau, kaip žinoma, visiško atpažinimo sistemos šiuo metu neegzistuoja ir DNT turi savo trūkumų, pavyzdžiui, ilgas išmokymo laikas, sudėtingas skaičiavimo procesas, o kai kada – ir apribojimų skaičius. Vadinasi, DNT nėra tobulas klasifikacijos metodas, bet turi daug daugiau pranašumų, palyginti su kitais klasifikatoriais.

2.6. Skyriaus apibendrinimas

Šioje magistro darbo dalyje detaliau aprašytas kiekvienas žmogaus balso atpažinimo sistemos kūrimo etapas. Tolesniam praktiniam darbui įgyvendinti buvo nagrinėjama, kaip teoriškai ir praktiškai realizuojami balso generavimo ir modeliavimo procesai. Nustatyta, kaip svarbu atlikti pirminį signalo apdorojimą ir jo segmentavimą. Kiekvienas signalas turi triukšmų, kurie gali būti įtraukti į požymių išskyrimo procesą ir dėl to norimo rezultato gavimo tikimybė prastėja. Be to, kiekviename atpažinimo procese garso įrašo signalas dalijamas į kadrus. Tik padalijus į kadrus ir pašalinus triukšmą, garsą galima siųsti į požymių išskyrimo metodą *MSKK*. Melų skalės kepstro koeficientų skaičiavimo rezultatas – parametrų vektorių seka, kuri siunčiama į mokymo modelį ir tada klasifikatorius lygina juos su nežinomo žmogaus balso signalu. Tolesnei klasifikatorių veikimo analizei ir tinkamo rezultato gavimui palyginti buvo pasirinkti ir plačiau aprašyti PMM ir DNT metodai, jų veikimo eiga ir naudojami algoritmai. Kiekvienas iš šių duomenų klasifikavimo būdų yra populiarus ir daugelyje atpažinimo sistemų taikomi metodai, kuriuos naudojant gautas rezultatas priklauso nuo išmokyto modelio ir tinkamų požymių pateikimo. Toliau darbe bus aprašytas praktinis žmogaus balso atpažinimo sistemų realizavimas su pasirinkta programavimo kalba ir realizavimo skirtumai bei panašumai.

3. KURIAMŲ ŽMOGAUS BALSO ATPAŽINIMO SISTEMŲ REALIZAVIMO EIGA

Magistro darbo praktinis įgyvendinimas buvo atliktas *Spyder* aplinkoje, kuriant sistemas su *Python* programavimo kalba. Programoms kurti buvo pasirinkta tokia interaktyvi kalba, nes ji turi daugybę galimybių, padedančių lengviau įgyvendinti paskirtą užduotį.

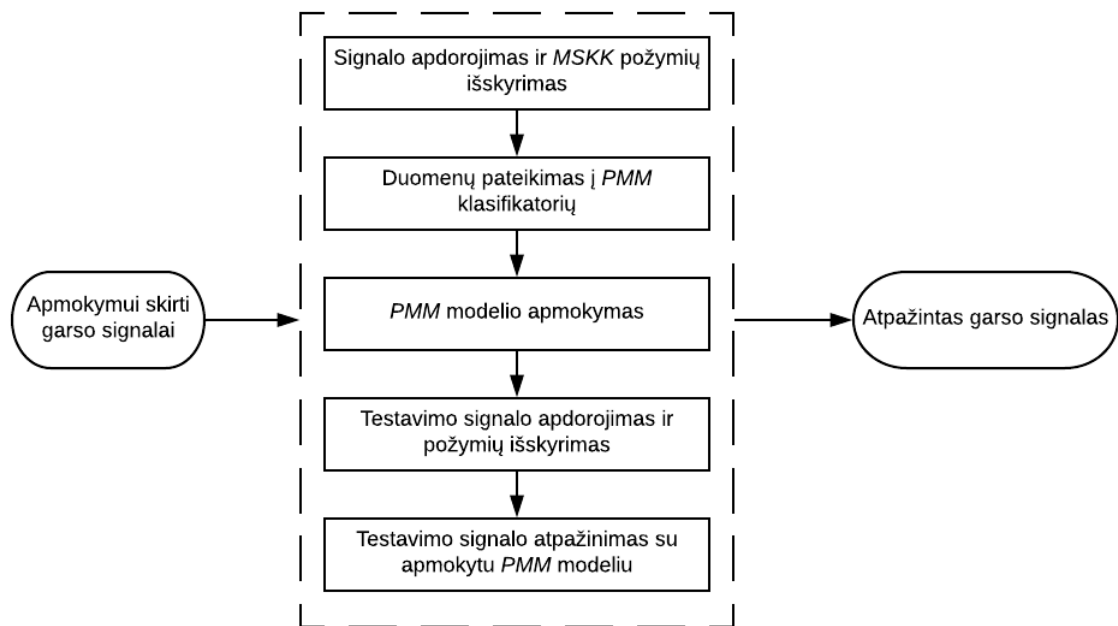
Python kalba yra atviro kodo, platinama pagal nemokamos programinės įrangos licencijas. Tai reiškia, kad programą, sukurtą naudojant šią kalbą, platinti galima net komerciniais tikslais. Ji taikoma kokybei optimizuoti, integruoti ir pernešti. Kitas *Python* programavimo kalbos pranašumas – ją galima naudoti kaip pseudokodą. Jo sintaksė yra lengvai suprantama ir tinka algoritmams mokytis. Tačiau *Python* kalba turi tam tikrų trūkumų. Pavyzdžiui, ji nėra tinkama programoms, kur labai svarbus komponentų vykdymo greitis arba reikia labai daug atminties. Nepaisant to, daugybė skirtingų programinių įrangų, kaip *Spyder*, *Blender* taiko ją naudotojo sąsajoje, projektuojant modelius, kuriant atpažinimo sistemas, dirbant su duomenų bazėmis ir pan. (Python Documentation contents, 2019).

Spyder programa yra *Python* programavimo kalbos atviro kodo platforma, skirta moksliniam programavimui. Pati platforma jau turi keletą integruotų naudingų paketų, pavyzdžiui, „Numpy“, „Pandos“, „Matplotlib“, leidžiančių lengviau atlikti tam tikrus algoritmų apskaičiavimus. Be to, jos galimybes galima išplėsti, naudojant papildomus *API*. Pavyzdžiui, atpažinimo sistemose, kad nereikėtų rašyti visų *PMM* algoritmų ir apdoroti duomenų, naudingiau prijungti „hmmlearn“ *API*. Su ja bus daug lengviau aprašyti kodą ir gauti norimą rezultatą (Spyder Docs, 2018).

Vadinasi, ši platforma ir programavimo kalba yra tinkamiausi magistro darbo užduočiai atlikti. Jos yra atviro kodo, su jomis lengvai galima programuoti, o jų trūkumai neturės jokios įtakos. Taikant *Python* programavimo kalbą ir *Spyder* platformą bei papildomus reikalingus *Python* paketus, bus kuriamos *PMM* ir *DNT* žmogaus balso atpažinimo programos. Šiame skyriuje plačiai nagrinėjama balso identifikavimo sistemos programavimo eiga. Buvo analizuojama, kaip vyksta balso apdorojimas su *MSKK* ir jo požymių išskyrimas. Aptariamieji kodo veikimo procesai ir analizuojami skaičiavimo būdai, taikant *PMM* ir *DNT*, kurie identifikuos signalus po mokymo etapo. Laukiamas rezultatas – abi programos turi išvesti paties identifikuoto žmogaus vardą ir nustatyti, kiek pateikti testavimo signalo požymiai sutampa su atitinkamų mokytis naudojamų signalų požymiais.

3.1. MSKK ir PMM atpažinimo sistemos įgyvendinimas

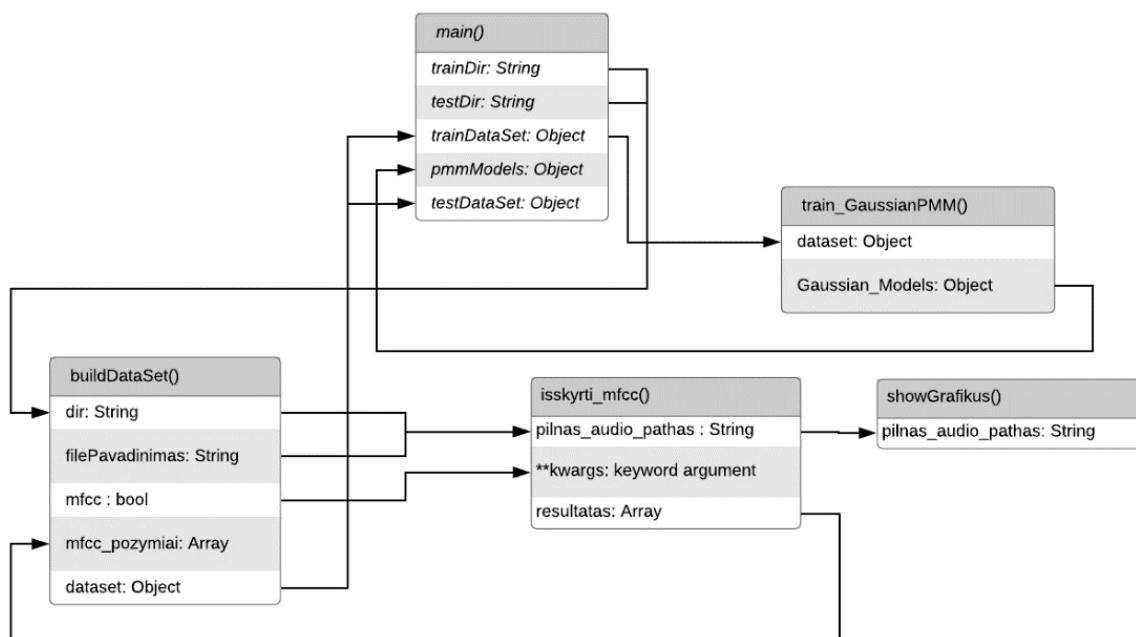
Kaip buvo rašyta ankstesniuose skyriuose, pirmieji žmogaus balso atpažinimo etapai yra duotųjų mokytį signalų apdorojimas ir *MSKK* požymių išskyrimas. Tada požymiai pateikiami *PMM* algoritmui, kuris apmokomas ir tada, kai suteikiami testavimo signalo *MSKK* požymiai, palyginamas su kiekvienu išmokyto modelio komponentu. Rezultatas – programa išveda testuoti pateikto signalo ir atpažinto signalo pavadinimus. Atliekant tokius veiksmus buvo kuriama *PMM* žmogaus balso atpažinimo sistema (3.1 pav.).



3.1 pav. *MSKK* su *PMM* atpažinimo programos veikimo principas

Duomenų perdavimas programoje vyksta nuo vienos klasės iki kitos ir atgal. Iš pradžių apmokytų signalų failų radimo vietos pavadinimas pateikiamas į *buildDataSet()* klasę. Iš visų failų, esančių aplanke, į *isskyrti_mfcc()* klasę siunčiami tik tie failai, kurie yra .wav formato. *isskyrti_mfcc()* klasėje iš kiekvieno signalo išskiriami *MSKK* požymiai, kurie grąžinami atgal į *buildDataSet()* klasę. Ten iš jų sukuriamas duomenų rinkinys. Tada duomenys siunčiami į *train_GaussianPMM()* klasę, kur vyksta *PMM* modelio apmokymas pagal nustatytus *GaussianHMM()* funkcijos parametrus. Kai mokymo modelis sukurtas, į *buildDataSet()* klasę siunčiamas atpažinti skirtas signalo failo radimo vietos pavadinimas. Kaip ir anksčiau su *buildDataSet()* ir *isskyrti_mfcc()* sukuriamas duotojo signalo duomenų rinkinys. Tada gauti duomenys lyginami su apmokytu *PMM* modeliu ir nustatomas atpažinto signalo pavadinimas pagal didžiausią modelio tikimybę. Be to, yra papildoma klasė *showGrafikus()*,

kuri iš *isskyrti_mfcc()* klasės paima signalo laikymo vietos pavadinimą ir naudojant specialias funkcijas sukuria signalo amplitudės diagramą ir *MSKK* spektrogramą (3.2 pav.).



3.2 pav. *MSKK* su *PMM* atpažinimo programos *UML* diagrama

Toliau darbe bus plačiau aprašytos visos programos ir funkcijų veikimas. Pirmiausiai *Spyder* platformoje kodo pradžioje buvo importuotas specialus *_future_* modulis. Norint, kad veiktų *Python* kodas, labai svarbu yra viršuje parašyti šį modulį, kad būtų atstatytos kitos aukštesnės *Python* versijų funkcijos dabartiniame interpretatoriuje. Pavyzdžiui, nors *print_function* jau pagal nutylėjimą įgalinta, ją vis tiek reikia aprašyti (3.3 pav.).

```
6 from __future__ import print_function
```

3.3 pav. *PMM* programos *_future_* modelio kodas

print_function bibliotekos esmė – paversti rašomą tekstą į unikodą ir pasiūlyti naują struktūrą ir funkcijos aprašą. Parašius šį modulį, galima importuoti kitas reikalingas bibliotekas (3.4 pav.)(Python Documentation contents, 2019).

```

7 import warnings
8 import librosa
9 import librosa.display
10 import os
11 from hmmlearn import hmm
12 import numpy as np
13 import matplotlib.pyplot as plt
14 import time

```

3.4 pav. *PMM* programos reikalingų bibliotekų kodas

Kiekvienas iš jų bus atskirai paaiškintas ir pateiktos priežastys, kodėl jie buvo prijungti prie programos ir kokias galimybes suteikia realizuojant sistemą.

Pirmiausiai *import warnings* modulis taikomas, kad programa galėtų pranešti naudotojui apie dabartinę sąlygą. Paprasčiausiai įspėti jį, kad kode taikomi moduliai yra arba pasenę, arba nebereikalingi. *Python* programavime įspėjimo pranešimai kontroliuojami dviem veiksmomis. Programa iš pradžių nustato, ar apskritai turi būti pateiktas įspėjimas. Tada, jeigu pranešimas visgi pateikiamas, jis atspausdinamas pagal naudotojo nustatytą būdą. Kode jis naudojamas, norint ignoruoti vienas kitą atitinkančius pranešimus. Tai atliekama kontroliuojant įspėjimą su filtru, kuris nurodo veiksmų seką. Kode tai parašoma taip: *warnings.filterwarnings('ignore')*. *filterwarnings()* šiuo atveju ir yra pranešimo taisyklių nustatymo funkcija (Python Documentation contents, 2019).

Kitos bibliotekos yra *import librosa* ir *import librosa.display*, kurios buvo prijungtos prie programos, siekiant išanalizuoti garso signalą. Be to, iš pateiktų elementų įmanoma suprogramuoti požymių paieškos algoritmą. *librosa.display* turi elementus, kurie vizualizuoja grafiškai apdorotą garsą. Apibendrinant, *librosa* suteikia galimybę patogiai sukurti signalo apdorojimą, nes turi jau automatiškai pagal nutylėjimą veikiančių nustatymų. Tačiau, prieš juos taikant, reikia įsitikinti, ar tikrai nustatyti nustatymai gerai suderinami su kito kodo dalimis. Kitu atveju tai gali sukelti klaidų ir nenumatytų atvejų (LibROSA Docs, 2018).

import os modulis taikomas, norint skaityti arba įrašyti failą. Turint prieinamumą prie operacinės sistemos funkcijų, jis suteikia galimybę manipuluoti failo keliais arba perskaityti komandų eilutes ir gauti informaciją apie procesą. Be to, naudojant *os* modulio funkcijas, susijusias su failo pavadinimais, gražinami ne tik failo baitai, bet ir patys eilutės objektai, kurie atitinka paties objekto tipą. Atpažinimo sistemoje jis reikalingas signalo failams skaityti ir jų pavadinimams saugoti tolesniems veiksmams.

Pati svarbiausia biblioteka, žymiai palengvinanti visos atpažinimo programos sukūrimą, yra *from hmmlearn import hmm*. Ji pateikia funkcijas, įgyvendinančias visą *PMM* funkcionalumą. Pagal nustatytą kintamųjų seką sukuriamos vidinės paslėptų būsenų sekos. Jos nėra tiesiogiai stebimos ir apibrėžiamos pagal pereinamosios tikimybes matricą ir tikimybės vektorius. Norint, kad tinkamai veiktų *hmmlearn* modulio funkcijos, labai svarbu tinkamai nustatyti visus parametrus. Tačiau, kaip buvo minėta pirmiau, *PMM* veikimas turi tam tikrų problemų, bet su dinaminio programavimo ir iteracijos algoritmais jos yra įveikiamos. „Viterbi“ ir „forward-backward“ algoritmai leidžia įvertinti optimalią paslėptųjų būsenų seką ir

apskaičiuoti modelio tikimybę. Taikant Baum'o ir Velčo modelio parametrų įvertinimo algoritmą su *PMM*, galima apskaičiuoti iš kiekvieno sudarančio fonetinio vieneto didžiausią tikimybę. Taip programos kodas galės įvertinti pateikto signalo požymių parametrus ir nustatyti jų panašumus (hmmlearn Tutorial, 2010).

Vienas iš pagrindinių „Python“ programavimo kalbos paketų, kuris taikomas kaip daugiamačių duomenų kaupiklis, vadinamasis „NumPy“. Be to, jis pats turi daugybę funkcijų ir algoritmų, padedančių lengviau atlikti skirtingas operacijas. Jo funkcijų veikimo greitis yra aukšto lygio. Pavyzdžiui, jis per trumpą laiką atlieka skirtingas masyvo manipuliacijas, tokias kaip rūšiavimas, formos keitimas, *DFT*, tiesinės algebros ir statistinės operacijos taikymai. „Numpy“ yra galinga biblioteka, suteikianti programuotojams daugybę būdų, kaip apdoroti masyvą. Tačiau ji turi tam tikrų apribojimų ir pranašumų. Visi masyvo dydžiai yra fiksuoti ir norint keisti jų elementų dydį, generuojamas naujas masyvas, ištrinant seną. Be to, visi masyvo elementai turi būti vienodo tipo. Kode ši biblioteka kviečiama su *import numpy as np*. Daugybė programuotojų naudoja ją dėl greitumo ir lankstumo, konvertuojant savo įvesties duomenis iškart į „NumPy“ masyvus. Atpažįstant balsą ji tam ir naudojama, kad duomenys būtų konvertuojami į reikalingą „NumPy“ masyvą ir į *PMM* modelį būtų pateikiama taisyklinga jo forma (NumPy, 2018).

Dar viena importuota biblioteka *PMM* atpažinimo sistemoje buvo *import matplotlib.pyplot as plt*, turinti funkcijas, atliekančias panašius veiksmus kaip ir *MatLab* programos funkcijos. Pavyzdžiui, ji taikoma brėžiniams ir figūroms piešti bei duomenims paryškinti. Programoje tai naudojama, siekiant parodyti signalo amplitudes ir paties *MSKK* požymių kitimą laike (Pyplot tutorial, 2012).

Paskutinė panaudota biblioteka buvo *import time*, suteikianti galimybę manipuluoti laiku. Ją naudojant buvo skaičiuojamas visos programos veikimo laikas bei apmokymo ir atpažinimo trukmė.

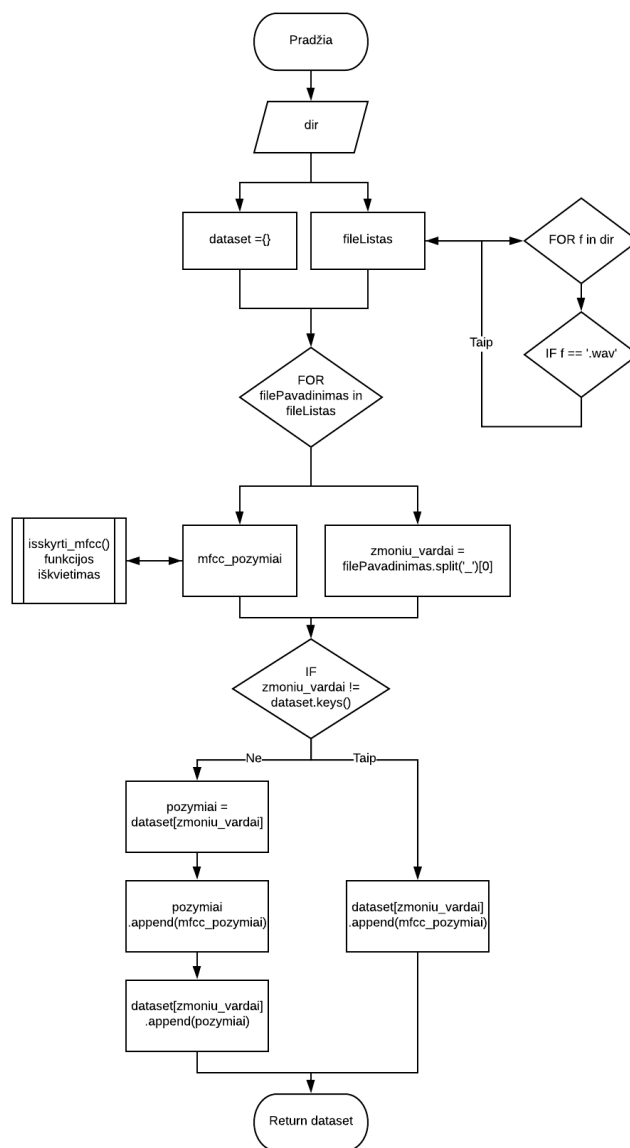
„Python“ programavimo kalboje paskutinėse eilutėse rašomas *if*, kuris patikrina, ar kodas neimportuojamas kaip modulis ir kokia funkcija turi būti kviečiama pirma kompiliavimo metu (3.5 pav.).

```
109 if __name__ == '__main__':  
110     main()
```

3.5 pav. *PMM* programos pradžios kodas

Nurodant *main()* funkciją, programa supranta tai kaip pradžios tašką ir nuo ko pradės veikti visa sistema. Iš pradžių sukuriamas *global_laiko_pradzia* kintamasis, kuris užfiksuoja laiką, kada pradėjo veikti programa.

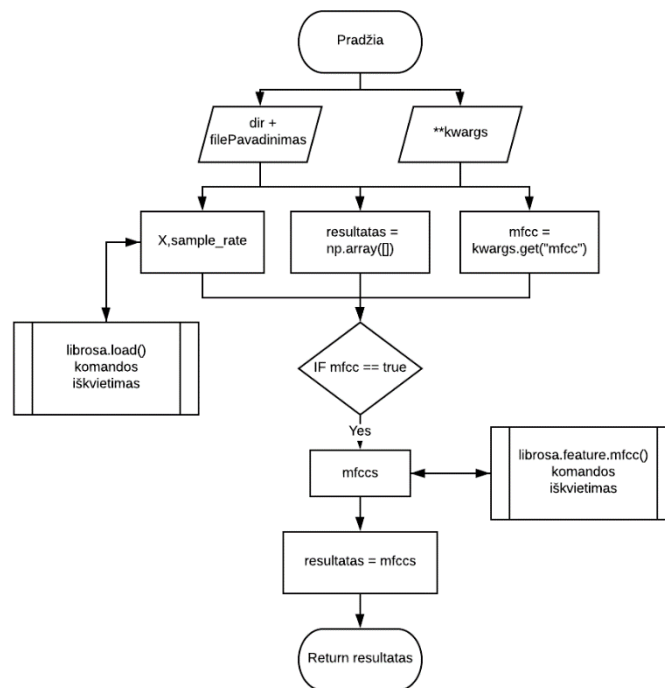
buildDataset() funkcijoje vyksta garso failų apdorojimas ir paruošimas *PMM* modeliui apmokyti, grąžinant visą reikalingą duomenų rinkinį (3.1 schema).



3.1 schema. *PMM* programos *buildDataSet()* funkcijos diagrama

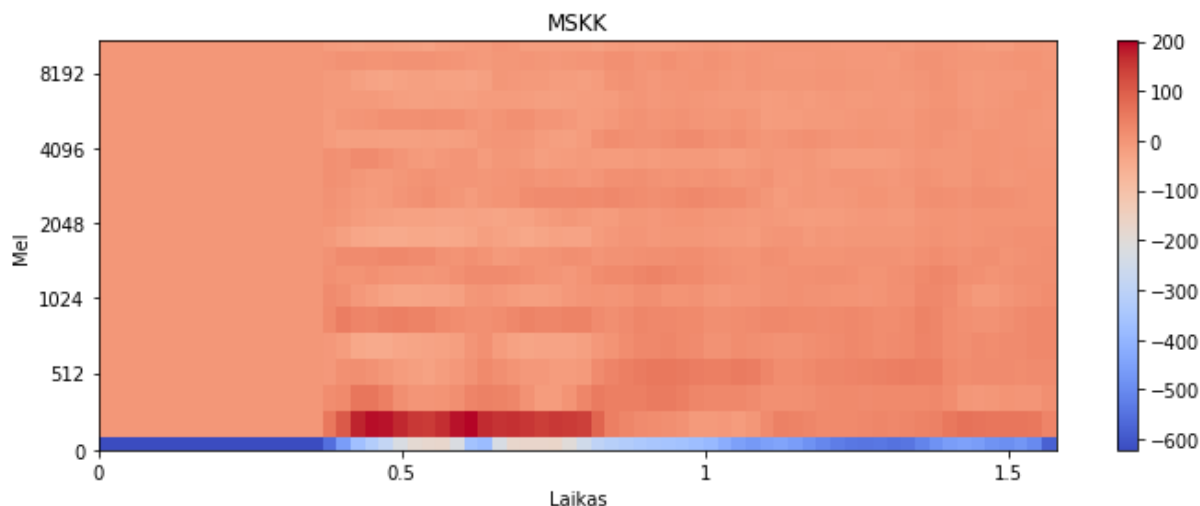
Iš pradžių iš visų pateiktų failų, esančių nurodytame aplanke, išskiriami tie, kurie turi gale *.wav* ir priskiriami prie *fileListas* kintamojo. Tokiu būdu į programą eis tik garso formato failai. Toliau su *for* funkcija praeinama per kiekvieną iš failų ir pridedama prie *zmoniu_vardai* kintamojo garso failo pavadinimas, kuris pavadintas tam tikro žmogaus vardu. *mfcc_pozymiai* kintamasis yra *MSKK* požymių rinkinys, kuris gaunamas iškviečiant *isskyrti_mfcc()* funkciją.

Tada atliekamas kuriamo duomenų rinkinio patikrinimas, ar jame jau egzistuoja tokio žmogaus vardas. Jei duomenų rinkinyje jau yra duomenų su tokiu vardu, jis prideda naujus *MSKK* požymius prie jų, o jei nėra – paprastai sukuria duomenų rinkinį su tokiu žmogaus vardu. Rezultatas – funkcija grąžins sugrupuotą pagal vardus duomenų rinkinį, kuris bus pateikiamas į *PMM* apmokyti.



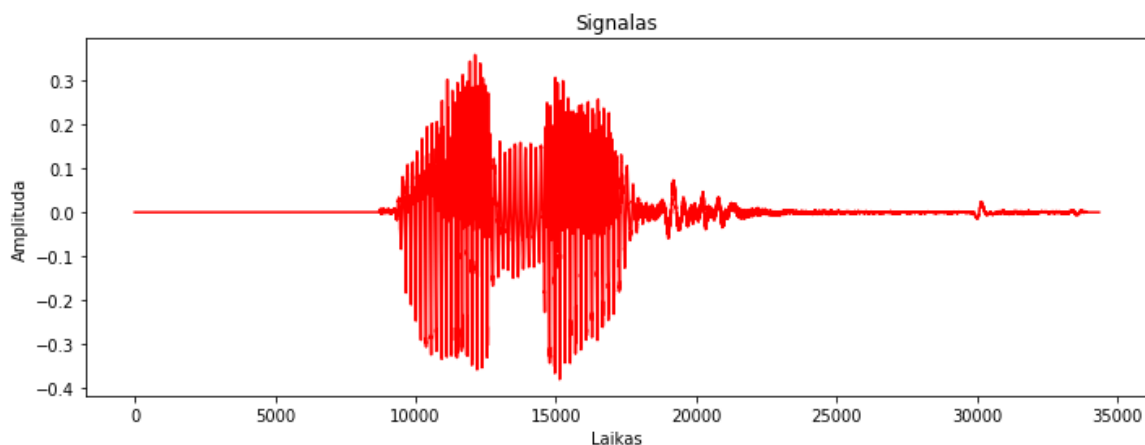
3.2 schema. *PMM* programos *isskyrti_mfcc()* funkcijos diagrama

Kaip buvo minėta, *buildDataSet()* funkcija turi dar vieną iškviečiamą *isskyrti_mfcc()* funkciją, kuri grąžina *MSKK* požymius (3.2 schema). Kaip pavaizduota viršuje, iš pradžių su *librosa.load* nuskaitomas garso failas ir grąžinamas dviejų elementų rinkinys *X*, *sample_rate*. *X* yra garso signalo laiko eilutės masyvas, o *sample_rate* yra atrankos dažnis. Naudojant šiuos du kintamuosius, sužinomi *MSKK* požymiai, paprastai perduodant juos *librosa.feature.mfcc()* funkcijai. Tolesnis veiksmas – į *showGrafikus()* klasę siunčiamas signalo laikymo vietos pavadinimas. Tada kuriamos dvi diagramos, norint patikrinti, ar tikrai gaunami reikalingi duomenys. Pirmiausiai su *plt.figure* nustatomas brėžinio dydis, o tada su *librosa.display.specshow* kuriama melų spektrograma. Kitos komandos papildo spektrogramą spalvų juosta, grafiko ir ašių pavadinimais. *plt.tight_layout()* modulis koreguoja brėžinį taip, kad jis gražiai atitiktų paveikslą. Su *plt.show()* parodoma sugeneruota melų spektrograma (3.6 pav.).



3.6 pav. Melų spektrogramos pavyzdys

Tokiu pat būdu kuriamas ir signalo grafikas, parodantis jo amplitudės priklausomybę nuo laiko. Vienintelis skirtumas yra tas, kad vietoj *librosa.display.specshow* naudojama *plt.plot()* funkcija. Šis modulis sukuria linijinį grafiką (3.7 pav.).



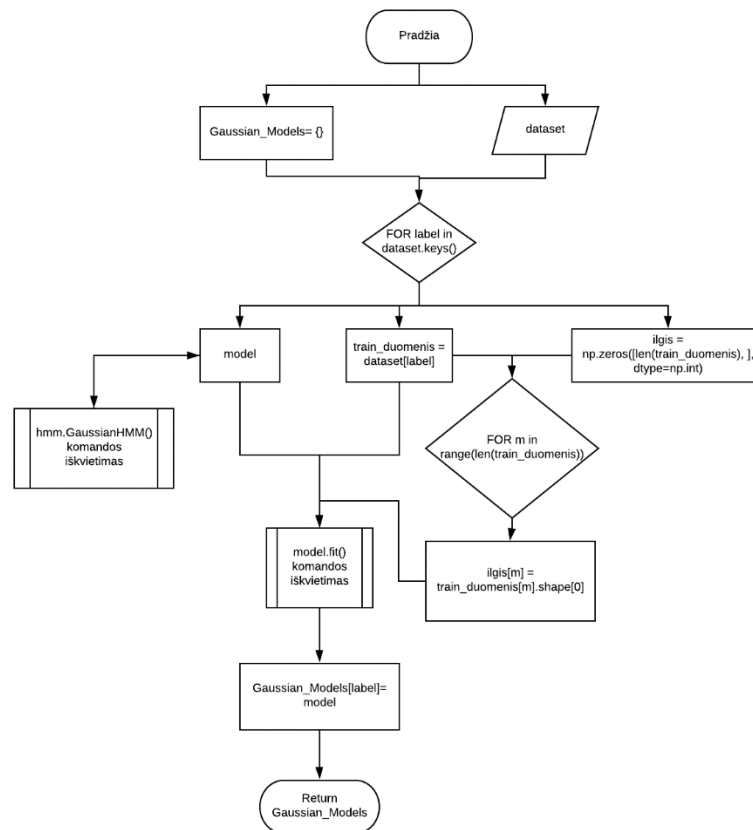
3.7 pav. Garso signalo amplitudės ir laiko grafiko pavyzdys

Pavaizdavus grafikus *isskyrti_mfcc()* funkcija, kaip buvo minėta, grąžina *MSKK* požymius, kurie vėliau bus įdedami į *PMM* apmokymo modelį (3.8 pav.).

```
92 pmmModels = train_GaussianPMM(trainDataSet)
```

3.8 pav. *PMM* programos *train_GaussianPMM()* funkcijos iškviatimo kodas

Kai *MSKK* požymiai gauti ir duomenų rinkinys sugeneruotas, jie siunčiami į *PMM* apmokymo funkciją, kuri grąžina į *pmmModels* kintamąjį patį *PMM* modelių duomenų rinkinį (3.3 schema).



3.3 schema. *PMM* programos *train_GaussianPMM()* funkcijos diagrama

Pirmiausiai su *for* ciklu pagal kiekvieno žmogaus vardo duomenų rinkinį pradedamas modelio apmokymas. Su *hmm.GaussianHMM()* funkcija nustatomi pagrindiniai modelio parametrai, tokie kaip iteracijų skaičius, paslėptųjų narių skaičius, dekoderio algoritmas, kovariacijos parametrų tipas. Sukurtoje sistemoje buvo nustatyti tokie pagrindiniai nekeičiami parametrai:

1. Dekoderio algoritmas yra *Viterbio*, nes jis gali surasti labiausiai tikėtiną paslėptų būsenų seką, kuri lemia stebimų įvykių seką.
2. Kovariacijos matrica, kuri naudojama kiekvienai būsenai, kaupiant viršutinę trikampio formą. Kitaip tariant, kiekvienos matricos eilutės pradžia yra įstrižainės elementas.

Visi kiti parametrai bus keičiami programos testavimo metu. Kitas veiksmas – sukuriamas atskiros sekos ilgio masyvas, sujungiant *train_duomenis* masyvą į vieną. Tada su komanda *fit()* inicijuojamas mokymo pradžios veiksmas, įvedant sujungtų stebėjimo sekų ir sekų ilgių matricas. Taip apmokamas *PMM* modelis ir jo priskyrimas naujai duomenų sekai.

Paskutiniame etape paimamas signalas, skirtas atpažinti, ir siunčiamas į *MSKK* požymių rinkinio sudarymo funkciją *buildDataSet()*. Požymių lyginimo etape paimamas kiekvienas *PMM* modelis pagal žmogaus vardą ir jeigu jo maksimali požymių tikimybė sutampa su atpažinti skirto signalo požymiais, tikslumo skaičiui pridedamas vienetas. Apmokant *PMM* duomenis, sužinoma, kam priklauso identifikuotas signalas, jeigu tokio žmogaus balsas buvo įdėtas į *PMM* apmokytą modelį.

Apmokymo laikas (s) : 1.5903542041778564
Pateiktas testavimo signalo pavadinimas: romas , Atpazintas zmogaus balsas yra: romas
Atpazinimo koeficientas: 100.00 %
Apmokymo ir atpazinimas laikas kartu (s) : 1.5963516235351562
Programos veikimo laikas (s) : 7.858771324157715

3.9 pav. *PMM* programos išvesties rezultato pavyzdys

Rezultatas – išvedami pateikto ir atpažinto žmogaus vardas. Be to, papildomai gaunamas tikslumas, su kurio buvo atpažintas vardas. Norint sužinoti programos veikimo laiką kiekviename etape, po kiekvieno žingsnio atimamas laikas iš dabartinio laiko ir to žingsnio pradžios (3.9 pav.). Valdymo pulte buvo išvedami apmokymo, atpažinimo ir visos programos veikimo laikai.

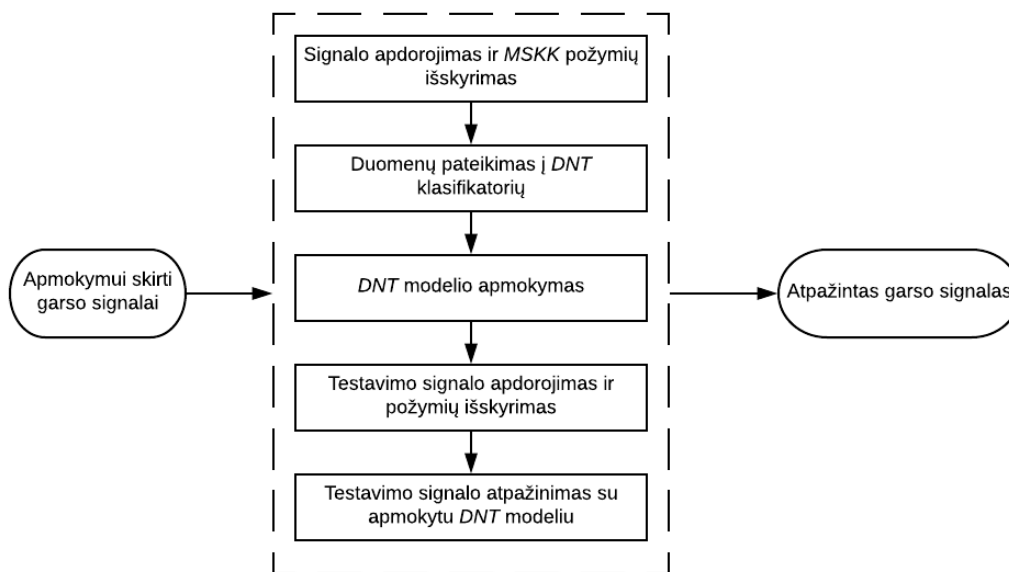
Tokiu metodu buvo sukurta *MSKK* ir *PMM* žmogaus balso atpažinimo sistema, naudojant *Spyder* platformą ir *Python* programavimo kalbą. Programų testavimo etape bus tikrinama duota sistema, keičiant *PMM* modelio parametrus, bei perduodant apmokyti ir atpažinti skirtingus garsus. Be to, rezultatai bus lyginami su kita identifikavimo sistema *MSKK* ir *DNT*, kuri irgi buvo sukurta naudojant *Python* programavimo kalbą.

3.2. *MSKK* ir giliojo mokymosi atpažinimo sistemos kūrimas

Giliojo mokymosi atpažinimo sistema yra dar vienas būdas, kaip galima atpažinti žmogų pagal jo balsą. Kaip ir *PMM* programa, *MSKK* ir *DNT* žmogaus balso identifikavimo programa buvo realizuota, naudojant *Spyder* ir *Python* programavimo kalbą. Šios programos struktūra ir tam tikrų veiksmų eiliškumas yra identiškas anksčiau aprašytai sistemai. Šioje dalyje bus pateikti skirtumai ir papildomi veiksmai, kuriuos reikėjo atlikti, norint realizuoti duotąją programą.

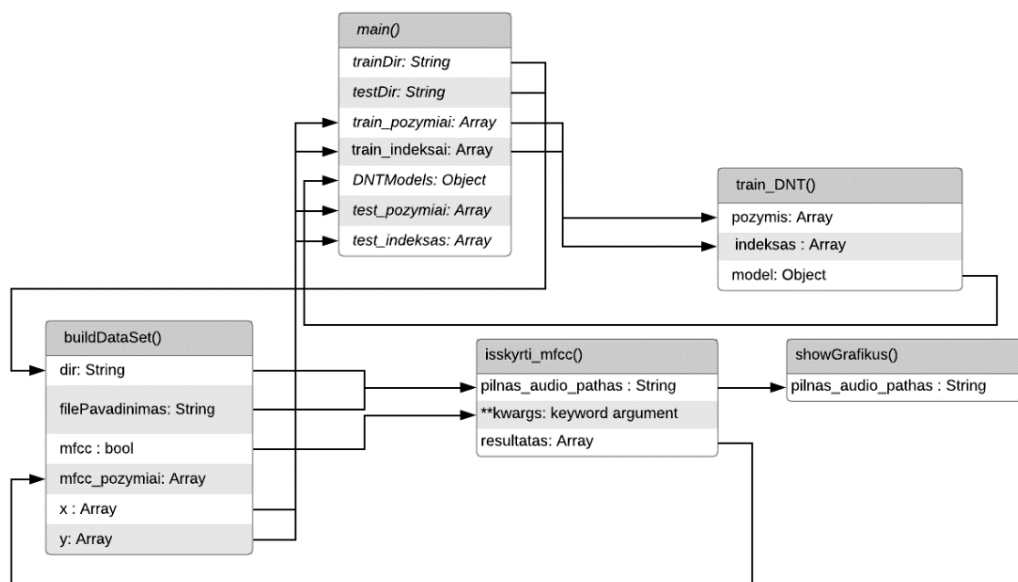
Pagal veikimo principą *DNT* programa beveik niekuo nesiskiria nuo *PMM* atpažinimo programos. Vienintelis skirtumas yra pats pasirinktas klasifikatorius, kuris taip pat paima

MSKK požymius ir pagal juos palygina signalus tarpusavyje (3.10 pav.). Rezultatas – sužinomas balso savininkas ir atpažinimo tikimybė.



3.10 pav. *MSKK* su *DNT* atpažinimo programos veikimo principas

Duomenų perdavimas identifikavimo sistemoje vyksta panašiai kaip ir *PMM* programoje. Tačiau dėl *DNT* klasifikatoriaus specifikos *Python* programavimo kalboje, kai kurie kintamieji buvo pakeisti tam, kad atitiktų reikalavimus. Vietoj duomenų rinkinio iš *buildDataSet()* klasės į *train_DNT()* klasę siunčiami du masyvai *pozymis* ir *indeksas*. Jie įdedami į *fit()* funkciją, kuri apmoko *DNT* modelį ir paruošia jį tolesniam atpažinimui. Identifikavimo etape taip pat naudojami du masyvai, kurių kintamieji paskui lyginami su *DNT* modelio kintamaisiais. Visų kitų klasių duomenų perdavimas yra panašus kaip ir *PMM* programoje (3.11 pav.).



3.11 pav. MSKK su DNT atpažinimo programos UML diagrama

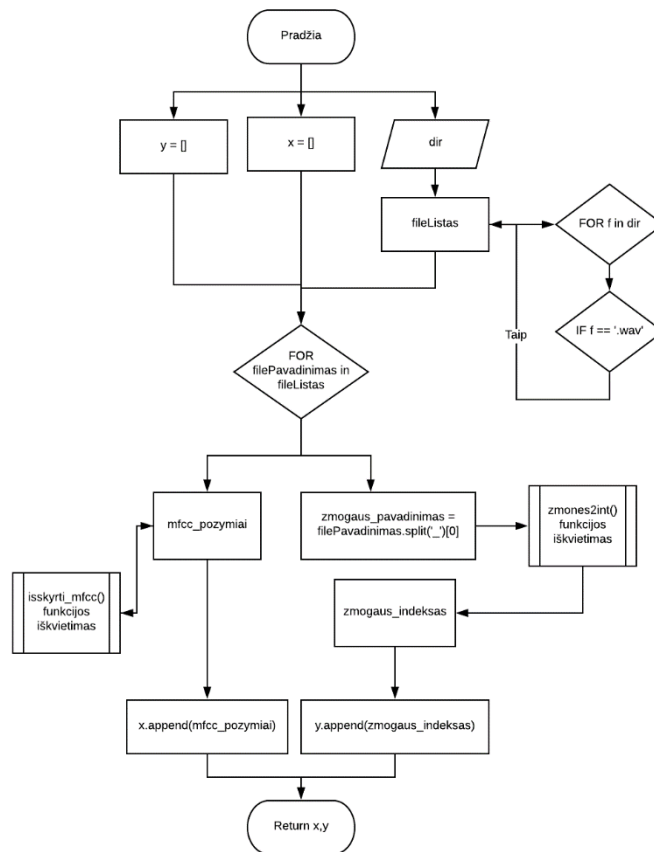
Visas kodo realizavimas bus aprašytas tolesniame darbo apraše. Be to, bus paaiškintos priežastys, kodėl buvo pasirinktas toks programos realizavimas.

```
13 import tensorflow as tf
```

3.12 pav. DNT programos *tensorflow* bibliotekos prijungimo kodas

Iš pradžių vietoj *PMM Python* bibliotekos reikėjo įdiegti *tensorflow* modulį, kuris skirtas giliojo mokymosi modeliui generuoti ir specialioms tinklo nustatymo funkcijoms prijungti (3.12 pav.). Visos kitos bibliotekos yra tokios pačios kaip ir tos, kurios buvo taikomos *PMM* atpažinimo sistemoje (Abadi, 2015; TFLearn, 2018).

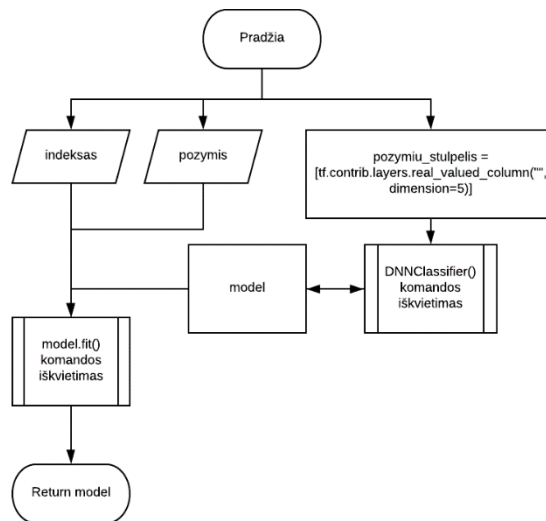
Kaip ir pirmoje programoje į *buildDataSet()* funkciją paduodama apmokyti skirtų failų aplanko vietos pavadinimas (3.4 schema). Tačiau *MSKK* požymių ir apmokymo pavadinimų indeksų masyvai grąžinami atskirai. Tai daroma dėl *tensorflow* duomenų įvedimo specifikos.



3.4 schema. *DNT* programos *buildDataSet()* funkcijos diagrama

buildDataSet() funkcijoje, *DNT* žmogaus balso atpažinimo sistemoje, kaip ir anksčiau išskiriami failų pavadinimai ir gaunami *MSKK* požymiai. Šios funkcijos veikimo skirtumas yra tas, kad sukuriamas ne duomenų rinkinys, o atskiri du masyvai su duomenimis. Be to, kadangi *DNT Python* klasifikatorius nepriima *String* formato kintamųjų, jie pakeičiami į *Int* formato kiekvieno pavadinimo indeksus ir įdedami į bendrą masyvą.

Požymių išskyrimo *isskyrti_mfcc()* funkcijoje grąžinamas *MSKK* požymių aritmetinis vidurkis išilgai 0 ašiai. Tada jis paduodamas į masyvą. Tolesniame apmokymo etape svarbu pateikti vienodą įvedamų duomenų skaičių dėl *tensorflow* specifinio modelio apmokymo.



3.5 schema. *DNT* programos *train_DNT()* funkcijos diagrama

DNT apmokymo funkcijoje paduodami gauti duomenų masyvai (3.5 schema). Iš pradžių paruošiami visi kintamieji *DNT* modeliui kurti, tokie kaip klasių skaičius, modelio naudojami stulpeliai ir epochų skaičius. Tada signalo požymiai ir balso indeksas paduodamas į *fit()* funkciją, nurodant apmokymo žingsnių skaičių. Kai modelis sukurtas, jis grąžinamas atgal į *main()* funkciją.

Galiausiai testavimo signalo požymiai ir pavadinimo indeksas pateikiami į *predict()* ir *evaluate()* funkcijas, norint atpažinti balso savininką ir sužinoti identifikavimo tikimybę. Rezultatas – gaunama tokia pat išvestis kaip *PMM* atpažinimo sistemoje (3.13 pav.). Be to, taip pat kaip ir *PMM* programoje, pateikiami apmokymo, atpažinimo ir visos programos veikimo laikai.

Apmokymo laikas (s) : 0.822526216506958

Pateiktas testavimo signalo pavadinimas: romas , Atpazintas žmogaus balsas yra: romas

Atpazinimo koeficientas: 100.00 %

Apmokymo ir atpažinimo laikas kartu (s) : 1.4941847324371338

Programos veikimo laikas (s) : 7.174628496170044

3.13 pav. *DNT* programos išvesties rezultato pavyzdys

Taigi, naudojant *tensorflow* modulį, buvo sukurta giliojo mokymosi žmogaus balso atpažinimo sistema. Pateiktas į identifikavimo programą garso signalas buvo sėkmingai apdorotas ir suskaičiuoti jo *MSKK* požymiai. Paskui, palyginus juos su *DNT* modelyje esančiais požymiais, buvo nustatytas balso savininkas. Sistemos testavimo etape bus keičiamas signalų, paduodamų į *DNT* modulį, skaičius bei pagrindiniai modulio parametrai. Tokiu būdu bus

testuojamos giliojo mokymosi atpažinimo sistemos galimybės ir tada lyginama su *PMM* identifikavimo programa.

3.3. Skyriaus apibendrinimas

Praktinėje magistro darbo dalyje buvo padarytos ir aprašytos dvi žmogaus balso atpažinimo programos. Jos abi buvo realizuotos, naudojant atviro kodo platformą *Spyder* su *Python* programavimo kalba. Jų tikslas – taikant *MSKK* požymius bei vieną iš klasifikavimo metodų nustatyti įrašyto balso savininką. Iš pradžių programoms pateikiami apmokyti skirti garso signalai. Tada iš jų išskiriami *MSKK* požymiai ir jie paduodami vienoje programoje į *PMM* klasifikatorių, o kitoje – į *DNT*. Atliekant paskutinį veiksmą, į programą paduodamas atpažinti skirtas garsas ir išgaunami jo *MSKK* požymiai, kurie lyginami su apmokyto modelio požymiais. Jeigu tarp modelio ir identifikuoti skirto signalo požymių bus surasti maksimaliai vienodų taškų turintys duomenų rinkiniai, jie bus laikomi kaip atpažinimo rezultatas. *PMM* ir *DNT* veikimo principai yra skirtingi ir jų tikslumas taip pat priklauso nuo tam tikrų dalykų. Todėl kitoje darbo dalyje bus tikrinamos kiekvieno iš jų galimybės ir tikslumas. Bus išanalizuotas jų veikimo greitis, su kokiais parametrais kiekvienas iš jų veikia geriausiai ir kiek apmokyti skirtų signalų jiems reikia, norint nustatyti tikslų rezultatą. Galiausiai visi gauti *PMM* ir *DNT* programos rezultatai bus lyginami tarpusavyje ir padaryta išvada, kuris iš jų geriausiai atlieka savo darbą, identifikuojant žmogaus balsą.

4. EKSPERIMENTINIS TYRIMAS

Atlikus literatūros analizę, susijusią su žmogaus balso atpažinimo sistemomis ir dviejų klasifikatorių veikimo principais, bei realizavus programas „Spyder“ aplinkoje su specialia „Python“ programavimo kalba, buvo realizuotos *PMM* ir *DNT* balso identifikavimo sistemos, naudojančios *MSKK* požymius. Abi programos teisingai nustato balso savininką, nustatius *MSKK* požymius ir tinkamai sukonfigūravus klasifikatorių parametrus. Atpažinimo programos naudoja *MSKK*, kurio požymiai laikomi panašiausiais, pagal kuriuos žmogus atpažįsta kito žmogaus balsą. Dėl tokios aukštos kokybės ir lygio jie ir buvo pasirinkti naudoti atpažinimo sistemose.

Pirmoji atpažinimo programa buvo sukurta taikant *PMM* klasifikatorių, kuris laikomas vienu iš geriausių ir dažniausiai naudojamų klasikinių metodų. Jis, naudodamas pateiktus signalo požymius, pradeda mokymosi etapą, po kurio, naudojant testavimo signalą, galima palyginti ir identifikuoti balso savininką. Kita identifikavimo programa buvo sukurta su *DNT* klasifikatoriumi, kurio atpažinimo tikimybė pastaruoju metu patobulėjo tiek, kad jis gali konkuruoti su *PMM* metodu. Jis, kaip ir *PMM*, naudoja pateiktus požymius ir grąžina apmokytą modelį. Tas modelis vėliau gali būti naudojamas atpažinimo procesuose, naudojant testavimo signalą. Testuojant abiejose sistemose bus tikrinama, kaip tinkamai atpažįstami žmonių balsai, keičiant klasifikatorių parametrus ir apmokymo signalų skaičių. Be to, jų palyginimas bus pagrįstas testavimo rezultatais ir laiku, per kurį programos atpažino žmogaus balsą.

4.1. Programų testavimas ir palyginimas

Sistemų veikimas ir funkcionalumas tikrinami pagal neuronų skaičių paslėptame sluoksnyje, pačių žingsnių bei pateiktų apmokyti signalų skaičius. Kai visi testavimo atvejai bus panaudoti abiejose programose, rezultatai bus palyginami tarpusavyje pagal atpažinimo tikimybę ir laiką, per kurį atliekamas sistemos darbas. Iš viso buvo testuojami 5 žmonių balsai ir tikrinami devyni parametrų keitimo atvejai:

- Kai neuronų skaičius ir apmokymo žingsnių skaičius tiesiogiai didinamas.
- Kai apmokymo žingsnių skaičius ir apmokymo signalų skaičius tiesiogiai didinamas.
- Kai apmokymo signalų skaičius ir neuronų skaičius tiesiogiai didinamas.
- Kai neuronų skaičius tiesiogiai didinamas, o apmokymo žingsnių skaičius tiesiogiai mažinamas.

- Kai neuronų skaičius tiesiogiai didinamas, o apmokymo signalų skaičius tiesiogiai mažinamas.
- Kai apmokymo žingsnių skaičius tiesiogiai didinamas, o apmokymo signalų skaičius tiesiogiai mažinamas.
- Kai neuronų skaičius tiesiogiai didinamas, apmokymo žingsnių skaičius yra šimtas, o pateiktų modeliui mokyti signalų skaičius yra penkiasdešimt.
- Kai apmokymo žingsnių skaičius tiesiogiai didinamas, o neuronų skaičius yra dešimt ir signalų skaičius yra penkiasdešimt.
- Kai apmokymo signalų skaičius tiesiogiai didinamas, o neuronų skaičius yra dešimt ir apmokymo žingsnis yra šimtas.

Apibendrinant visus testavimo etapus, bus sudarytos lentelės ir grafikai, siekiant geriau ir lengviau pamatyti gautą rezultatą ir pagal jį suformuluoti išvadas. Pagal tai bus lyginamos *PMM* ir *DNT* atpažinimo sistemos ir pasirinkta geresnė.

6.1 lentelė. Atpažinimo rezultatas ir laikas, didinant neuronų skaičių ir apmokymo žingsnių skaičių

Signalų numeris	Neuronų skaičius	Apmokymo žingsnių skaičius	<i>PMM</i> apmokymo ir atpažinimo laikas (s)	<i>PMM</i> atpažino balsą	<i>DNT</i> apmokymo ir atpažinimo laikas (s)	<i>DNT</i> atpažino balsą
1	1	5	0,249	Taip	1,804	Ne
	5	10	0,762	Taip	1,886	Ne
	10	50	1,066	Taip	1,732	Taip
	50	100	4,006	Taip	1,653	Taip
	100	1000	8,616	Taip	2,898	Taip
2	1	5	0,254	Taip	1,493	Ne
	5	10	0,741	Taip	1,49	Taip
	10	50	1,003	Taip	1,619	Taip
	50	100	4,135	Taip	1,729	Taip
	100	1000	9,498	Taip	2,95	Taip
3	1	5	0,249	Taip	1,482	Ne
	5	10	0,732	Taip	1,481	Ne
	10	50	0,997	Taip	1,58	Taip

	50	100	4,107	Taip	1,788	Taip
	100	1000	8,812	Taip	3,045	Taip
4	1	5	0,229	Taip	1,472	Ne
	5	10	0,745	Taip	1,481	Taip
	10	50	1,081	Taip	1,508	Taip
	50	100	4,151	Taip	1,675	Taip
	100	1000	8,854	Taip	3,021	Taip
5	1	5	0,281	Taip	1,434	Taip
	5	10	0,665	Taip	1,511	Taip
	10	50	0,977	Taip	1,528	Taip
	50	100	3,876	Taip	1,63	Taip
	100	1000	8,511	Taip	4,385	Taip
Laiko vidurkis (s) ir atpažinimo tikimybė (%)			2,983	100,00	1,931	76,00

6.1 lentelėje didinant pagrindinius *DNT* ir *PMM* klasifikatorių parametrus, buvo gautas vidutinis apmokymo ir atpažinimo laikas bei atpažinimo tikimybė. Didinant *PMM* neuronų ir apmokymo žingsnių skaičius, didėjo ir laikas, per kurį apmokomas modelis. Tačiau balso atpažinimas, net su mažomis reikšmėmis, buvo teisingas. Naudojant *DNT* modelį, laikas taip pat didėdavo, bet ne taip žymiai kaip su *PMM*. Klaidingų atpažinimo atvejų buvo daugiau ir pagal lentelės duomenis galima matyti, kad tai būdavo esant mažiems parametrams. Geriausi rezultatai dažniausiai buvo gaunami naudojant nuo paslėptųjų kintamųjų skaičiaus dešimt ir nuo mokymo žingsnių skaičiaus penkiasdešimt.

6.2 lentelė. Atpažinimo rezultatas ir laikas, didinant neuronų skaičių ir apmokymo signalų skaičių

Signalų numeris	Neuronų skaičius	Apmokymo signalų skaičius	<i>PMM</i> apmokymo ir atpažinimo laikas (s)	<i>PMM</i> atpažino balsą	<i>DNT</i> apmokymo ir atpažinimo laikas (s)	<i>DNT</i> atpažino balsą
1	1	5	0,12	Taip	1,813	Taip
	5	10	0,301	Taip	1,54	Taip
	10	50	1,06	Taip	1,868	Taip
	50	100	6,13	Taip	1,526	Taip
	100	400	48,644	Taip	1,893	Taip

2	1	5	0,13	Taip	1,986	Ne
	5	10	0,352	Taip	2,04	Taip
	10	50	1,003	Taip	1,704	Taip
	50	100	6,111	Taip	1,643	Taip
	100	400	48,587	Taip	1,903	Taip
3	1	5	0,119	Ne	1,634	Ne
	5	10	0,289	Taip	1,757	Taip
	10	50	0,989	Taip	1,554	Taip
	50	100	5,986	Taip	1,612	Taip
	100	400	49,178	Taip	1,716	Taip
4	1	5	0,181	Ne	1,603	Ne
	5	10	0,298	Taip	1,765	Taip
	10	50	1,038	Taip	1,750	Taip
	50	100	6,041	Taip	1,683	Taip
	100	400	52,605	Taip	2,468	Taip
5	1	5	0,115	Ne	1,591	Taip
	5	10	0,304	Ne	1,692	Taip
	10	50	0,962	Taip	1,604	Taip
	50	100	6,556	Taip	1,682	Taip
	100	400	52,005	Taip	2,391	Taip
Laiko vidurkis (s) ir atpažinimo tikimybė (%)			11,564	84,00	1,776	88,00

Pagal 6.2 lentelės testavimo atvejį buvo naudojama šimtas apmokymo žingsnių ir tiesiogiai didinamas neuronų skaičius bei pačių paduodamų signalų skaičius. Kaip ir ankstesnėje lentelėje, didinant kintamuosius, didėja ir laikas, per kurį atpažįstamas balsas. Kai signalų skaičius buvo šimtas ir keturi šimtai, programos darbo laikas žymiai pailgėjo, ypač naudojant *PMM* klasifikatorių. Atliekant šį testavimą, *PMM* klasifikatoriaus atpažinimo tikimybė buvo prastesnė ir pagal lentelės duomenis tai atsitikdavo, kai abu parametrai buvo maži. *DNT* klasifikatorius net su mažu parametų skaičiumi sugebėdavo atpažinti žmogaus balsą. Tik kai apmokymų signalų buvo penki ir neuronų skaičius vienas, kartais programa negebėdavo teisingai atpažinti.

6.3 lentelė. Atpažinimo rezultatas ir laikas, didinant apmokymo žingsnių skaičių ir apmokymo signalų skaičių

Signalų numeris	Apmokymo žingsnių skaičius	Apmokymo signalų skaičius	<i>PMM</i> apmokymo ir atpažinimo laikas (s)	<i>PMM</i> atpažino balsą	<i>DNT</i> apmokymo ir atpažinimo laikas (s)	<i>DNT</i> atpažino balsą
1	5	5	0,407	Taip	1,594	Ne
	10	10	0,497	Taip	1,431	Taip
	50	50	0,995	Taip	1,49	Taip
	100	100	1,416	Taip	1,603	Taip
	1000	400	3,851	Taip	3,734	Taip
2	5	5	0,41	Taip	1,50	Taip
	10	10	0,442	Taip	1,72	Taip
	50	50	1,017	Taip	1,571	Taip
	100	100	1,577	Taip	1,857	Taip
	1000	400	4,42	Taip	3,893	Taip
3	5	5	0,411	Ne	1,626	Taip
	10	10	0,499	Taip	1,431	Taip
	50	50	1,006	Taip	2,248	Taip
	100	100	2,112	Taip	1,671	Taip
	1000	400	4,309	Taip	3,535	Taip
4	5	5	0,478	Ne	1,431	Ne
	10	10	0,512	Taip	1,517	Taip
	50	50	0,993	Taip	1,722	Taip
	100	100	1,411	Taip	1,649	Taip
	1000	400	3,955	Taip	3,643	Taip
5	5	5	0,386	Ne	1,529	Taip
	10	10	0,489	Ne	1,587	Taip
	50	50	1,003	Taip	1,598	Taip
	100	100	1,605	Taip	1,751	Taip
	1000	400	4,002	Taip	3,556	Taip
Laiko vidurkis (s) ir atpažinimo tikimybė (%)			1,528	84,00	2,035	92,00

6.3 lentelėje neuronų skaičius buvo pasirinktas dešimt ir visi kiti kintamieji buvo didinami. Lentelės duomenys patvirtina, kad su mažais parametrais atpažinimo tikimybė yra prastesnė, ypač naudojant *PMM* klasifikatorių. Tačiau *DNT* sistemos darbo tikslumas buvo aukšto lygio ir tik pora kartų su mažais parametrais ji nesugebėdavo atpažinti žmogaus. Apmokymo ir atpažinimo laikai būdavo ilgesni didinant parametrus. Tai geriau matoma *PMM* klasifikatoriuje, bet jo pradiniai laikai buvo daug kartų mažesni už *DNT*. *DNT* programos laikai buvo maždaug pastovūs ir tik su aukštais parametrais laikas padidėdavo.

6.4 lentelė. Atpažinimo rezultatas ir laikas, didinant neuronų skaičių ir mažinant apmokymo žingsnių skaičių

Signalų numeris	Neuronų skaičius	Apmokymo žingsnių skaičius	<i>PMM</i> apmokymo ir atpažinimo laikas (s)	<i>PMM</i> atpažino balsą	<i>DNT</i> apmokymo ir atpažinimo laikas (s)	<i>DNT</i> atpažino balsą
1	1	1000	0,225	Taip	3,186	Ne
	5	100	0,708	Taip	1,556	Taip
	10	50	1,083	Taip	1,496	Taip
	50	10	3,778	Taip	1,412	Ne
	100	5	8,427	Taip	1,649	Taip
2	1	1000	0,263	Taip	2,603	Ne
	5	100	0,76	Taip	1,574	Ne
	10	50	0,94	Taip	1,479	Taip
	50	10	3,869	Taip	1,427	Ne
	100	5	8,749	Taip	1,433	Ne
3	1	1000	0,224	Taip	1,645	Taip
	5	100	0,728	Taip	1,445	Taip
	10	50	0,974	Taip	1,479	Taip
	50	10	3,885	Taip	1,547	Taip
	100	5	8,613	Taip	1,534	Taip
4	1	1000	0,292	Taip	2,931	Taip
	5	100	0,679	Taip	1,507	Taip
	10	50	0,976	Taip	1,499	Taip
	50	10	3,819	Taip	1,482	Ne
	100	5	8,786	Taip	1,701	Taip
5	1	1000	0,227	Taip	2,598	Ne

	5	100	0,711	Taip	1,705	Taip
	10	50	0,999	Taip	1,403	Taip
	50	10	3,856	Taip	1,391	Taip
	100	5	8,545	Taip	1,483	Taip
Laiko vidurkis (s) ir atpažinimo tikimybė (%)			2,884	100,00	1,726	68,00

Pagal 6.4 lentelę, kai apmokymo signalų skaičius buvo penkiasdešimt, o pagrindiniai sistemos parametrai keisdavosi tarpusavyje, *PMM* darbo laikas didėdavo, jei didėdavo neuronų skaičius, o apmokymo žingsniai mažėdavo. Tačiau tai neturėjo įtakos atpažinimui ir visais atvejais *PMM* sugebėjo be klaidų atpažinti žmogaus balsą. *DNT* klasifikatoriaus darbo laikai visais atvejais buvo maždaug pastovūs, bet atpažinimas buvo labai prastas, kai vienas arba kitas parametras buvo gana mažas, ypač kai neuronų skaičius neviršydavo penkių.

6.5 lentelė. Atpažinimo rezultatas ir laikas, didinant neuronų skaičių ir mažinant apmokymo signalų skaičių

Signalų numeris	Neuronų skaičius	Apmokymo signalų skaičius	<i>PMM</i> apmokymo ir atpažinimo laikas (s)	<i>PMM</i> atpažino balsą	<i>DNT</i> apmokymo ir atpažinimo laikas (s)	<i>DNT</i> atpažino balsą
1	1	400	1,011	Taip	1,66	Ne
	5	100	1,038	Taip	1,597	Taip
	10	50	1,01	Taip	1,587	Taip
	50	10	0,527	Ne	1,612	Taip
	100	5	0,564	Ne	1,686	Taip
2	1	400	1,043	Taip	1,592	Ne
	5	100	1,002	Taip	1,938	Ne
	10	50	0,958	Taip	1,806	Taip
	50	10	0,443	Ne	1,536	Taip
	100	5	0,456	Ne	1,581	Taip
3	1	400	1,096	Taip	1,58	Taip
	5	100	1,093	Taip	1,537	Taip
	10	50	0,954	Taip	1,621	Taip
	50	10	0,448	Ne	2,015	Taip
	100	5	0,434	Ne	2,12	Taip

4	1	400	1,043	Taip	1,685	Taip
	5	100	0,946	Taip	1,596	Taip
	10	50	0,994	Taip	1,601	Taip
	50	10	0,467	Ne	1,582	Taip
	100	5	0,435	Ne	1,588	Ne
5	1	400	1,12	Taip	3,263	Ne
	5	100	1,013	Taip	1,644	Taip
	10	50	0,986	Taip	1,574	Taip
	50	10	0,323	Ne	1,584	Taip
	100	5	0,413	Ne	1,564	Taip
Laiko vidurkis (s) ir atpažinimo tikimybė (%)			0,792	60,00	1,726	80,00

6.5 lentelėje kaip pastovus kintamasis buvo pasirinkta apmokymo žingsniai, o tarpusavyje kintantys parametrai buvo neuronų ir apmokymo signalų skaičius. Kai apmokymo signalų skaičius buvo dešimt arba penki, *PMM* klasifikatoriaus atpažinimas prastėjo ir duodavo klaidingą rezultatą. Laikas šio testavimo metu, kai signalų skaičius buvo mažas, mažėdavo. *DNT* darbo teisingumas – atvirkščiai – nepriklausydavo nuo pateiktų signalų skaičiaus, bet kai neuronų buvo mažai, jis pateikdavo klaidingą rezultatą. Klasifikatoriaus apmokymo ir atpažinimo laikai buvo maždaug pastovūs ir tik kai signalų skaičius buvo keturi šimtai, jis padidėdavo.

6.6 lentelė. Atpažinimo rezultatas ir laikas, didinant apmokymo žingsnių skaičių ir mažinant apmokymo signalų skaičių

Signalų numeris	Apmokymo žingsnių skaičius	Apmokymo signalų skaičius	<i>PMM</i> apmokymo ir atpažinimo laikas (s)	<i>PMM</i> atpažino balsą	<i>DNT</i> apmokymo ir atpažinimo laikas (s)	<i>DNT</i> atpažino balsą
1	5	400	4,09	Taip	1,401	Taip
	10	100	1,375	Taip	1,43	Taip
	50	50	0,94	Taip	1,507	Taip
	100	10	0,468	Taip	1,624	Taip
	1000	5	0,401	Taip	3,457	Ne
2	5	400	3,035	Taip	1,477	Ne
	10	100	1,387	Taip	1,511	Ne

	50	50	1,021	Taip	1,703	Taip
	100	10	0,469	Taip	1,594	Taip
	1000	5	0,376	Taip	2,753	Taip
3	5	400	3,036	Taip	1,441	Ne
	10	100	1,481	Taip	1,51	Ne
	50	50	0,941	Taip	1,52	Taip
	100	10	0,426	Taip	1,85	Taip
	1000	5	0,368	Ne	3,273	Taip
4	5	400	3,997	Taip	1,47	Ne
	10	100	1,558	Taip	1,612	Ne
	50	50	0,977	Taip	1,422	Taip
	100	10	0,507	Taip	1,495	Taip
	1000	5	0,379	Ne	3,330	Ne
5	5	400	4,344	Taip	1,660	Ne
	10	100	1,423	Taip	1,425	Taip
	50	50	0,99	Taip	1,811	Taip
	100	10	0,468	Ne	1,783	Taip
	1000	5	0,368	Ne	2,775	Taip
Laiko vidurkis (s) ir atpažinimo tikimybė (%)			1,393	84,00	1,873	64,00

DNT klasifikatorius pagal 6.6 lentelės duomenis parodė prastus rezultatus, kai apmokymo žingsnių skaičius buvo mažas. O laikas buvo maždaug pastovus ir tik su tūkstančio žingsnių skaičiumi darbo laikas žymiai padidėjo. Pagal šiuos lentelės duomenis, kaip ir ankstesnėje lentelėje, *PMM* sistema parodė klaidingus rezultatus, kai apmokymo signalų skaičius buvo dešimt arba penki. Darbo laikas taip pat mažėdavo, mažinant apmokymo signalų skaičių.

Vertinant keturis testavimo atvejus, kai buvo keičiama po du parametrus, buvo nustatyta, kad abu klasifikatoriai geriausiai dirba tada, kai neuronų skaičius yra maždaug dešimt, apmokymo signalų skaičius – apie penkiasdešimt, o apmokymo žingsnių skaičius – šimtas. Kitų testavimų metu bus tikrinamas tik vieno parametro keitimas, o du kiti liks pastovūs ir bus lygūs tiems parametrams, kuriuos nustačius buvo gauti geriausi rezultatai ankstesnio testavimo metu.

6.7 lentelė. Atpažinimo tikimybė ir laikas, didinant neuronų skaičių

Signalų numeris	Neuronų skaičius	<i>PMM</i> apmokymo ir atpažinimo laikas (s)	<i>PMM</i> atpažino balsą	<i>DNT</i> apmokymo ir atpažinimo laikas (s)	<i>DNT</i> atpažino balsą
1	1	0,247	Taip	1,684	Ne
	5	0,592	Taip	1,591	Taip
	10	0,937	Taip	1,617	Taip
	50	3,986	Taip	1,600	Taip
	100	8,875	Taip	2,429	Taip
2	1	0,23	Taip	1,571	Ne
	5	0,657	Taip	1,582	Taip
	10	0,936	Taip	1,551	Taip
	50	3,849	Taip	1,689	Taip
	100	8,741	Taip	1,742	Taip
3	1	0,243	Taip	1,547	Ne
	5	0,669	Taip	1,744	Taip
	10	0,98	Taip	1,649	Taip
	50	3,767	Taip	1,564	Taip
	100	8,644	Taip	1,58	Taip
4	1	0,234	Taip	1,79	Ne
	5	0,666	Taip	1,548	Taip
	10	1,000	Taip	1,593	Taip
	50	3,799	Taip	1,757	Taip
	100	8,739	Taip	1,599	Taip
5	1	0,24	Taip	1,516	Ne
	5	0,702	Taip	1,825	Taip
	10	0,956	Taip	1,464	Taip
	50	3,788	Taip	1,579	Taip
	100	8,589	Taip	1,577	Taip
Laiko vidurkis (s) ir atpažinimo tikimybė (%)		2,882	100,00	1,655	80,00

6.7 lentelėje buvo keičiamas tik neuronų skaičius, o apmokymo žingsnių skaičius lygus šimtui ir pateikta penkiasdešimt signalų, skirtų apmokyti. Pagal gautus duomenis *PMM* sistemos tikslingumas buvo gana aukštas, bet atpažinimo laikas didėdavo didėjant paslėptiesiems vienetams. O *DNT* pateikė prastesnius rezultatus, kai kintantis skaičius buvo mažas. Apmokymo ir atpažinimo laikai stipriai nesikeisdavo ir buvo lygūs maždaug 1,5 sekundės.

6.8 lentelė. Atpažinimo rezultatas ir laikas, didinant apmokymo žingsnių skaičių

Signalų numeris	Apmokymo žingsnių skaičius	<i>PMM</i> apmokymo ir atpažinimo laikas (s)	<i>PMM</i> atpažino balsą	<i>DNT</i> apmokymo ir atpažinimo laikas (s)	<i>DNT</i> atpažino balsą
1	5	0,948	Taip	1,499	Taip
	10	1,021	Taip	1,416	Taip
	50	1,004	Taip	1,691	Taip
	100	0,993	Taip	1,864	Taip
	1000	0,919	Taip	2,633	Taip
2	5	0,930	Taip	1,723	Taip
	10	1,000	Taip	1,502	Ne
	50	0,972	Taip	1,439	Taip
	100	0,975	Taip	1,48	Taip
	1000	0,932	Taip	2,779	Taip
3	5	0,958	Taip	1,60	Ne
	10	1,001	Taip	1,469	Ne
	50	0,958	Taip	1,441	Taip
	100	0,986	Taip	1,545	Taip
	1000	0,983	Taip	2,7	Taip
4	5	0,957	Taip	1,354	Ne
	10	0,988	Taip	1,467	Taip
	50	1,15	Taip	1,464	Ne
	100	0,934	Taip	1,50	Taip
	1000	1,018	Taip	2,679	Taip
5	5	0,972	Taip	1,412	Taip
	10	0,948	Taip	1,409	Taip

	50	0,96	Taip	1,431	Taip
	100	1,029	Taip	1,599	Taip
	1000	0,957	Taip	2,646	Taip
Laiko vidurkis (s) ir atpažinimo tikimybė (%)		0,979	100,00	1,749	80,00

Priešpaskutinio testavimo metu pagal 6.8 lentelę buvo keičiamas apmokymo žingsnių skaičius. Pagal lentelės duomenis galima daryti išvadą, kad *PMM* klasifikatoriaus atpažinimui žingsnių skaičius neturėjo įtakos ir laikas buvo maždaug vienodas visais atvejais. *DNT* klasifikatoriaus apmokymo ir atpažinimo laikai žymiai padidėdavo, kai apmokymo žingsnis buvo lygus tūkstančiui. Žmogaus balso identifikavimas gerėjo tik tais atvejais, kai apmokymo žingsnių skaičius buvo virš dešimt.

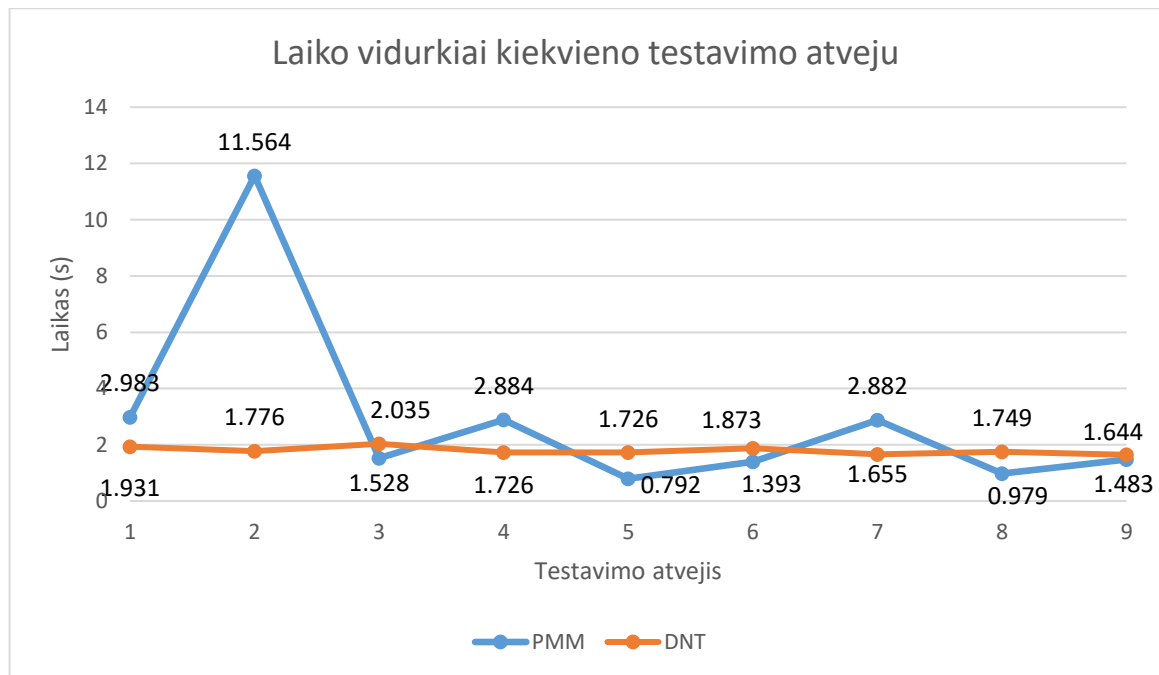
6.9 lentelė. Atpažinimo rezultatas ir laikas, didinant apmokymo signalų skaičių

Signalų numeris	Apmokymo signalų skaičius	PMM apmokymo ir atpažinimo laikas (s)	PMM atpažino balsą	DNT apmokymo ir atpažinimo laikas (s)	DNT atpažino balsą
1	5	0,37	Ne	1,737	Ne
	10	0,618	Taip	1,52	Taip
	50	0,918	Taip	1,80	Taip
	100	1,392	Taip	1,639	Taip
	400	3,979	Taip	1,657	Taip
2	5	0,359	Ne	1,795	Ne
	10	0,655	Taip	1,752	Taip
	50	0,946	Taip	1,671	Taip
	100	1,401	Taip	1,604	Taip
	400	3,892	Taip	1,629	Taip
3	5	0,363	Ne	1,50	Taip
	10	0,748	Taip	1,559	Taip
	50	0,955	Taip	1,55	Taip
	100	1,415	Taip	1,832	Taip
	400	3,992	Taip	1,805	Taip
4	5	0,375	Ne	1,795	Ne

	10	0,705	Taip	1,479	Taip
	50	1,017	Taip	1,531	Taip
	100	1,45	Taip	1,567	Taip
	400	3,932	Taip	1,689	Taip
5	5	0,392	Ne	1,52	Taip
	10	0,739	Taip	1,567	Taip
	50	0,933	Taip	1,667	Taip
	100	1,52	Taip	1,533	Taip
	400	4,033	Taip	1,711	Taip
Laiko vidurkis (s) ir atpažinimo tikimybė (%)		1,483	80,00	1,644	88,00

Paskutinio testavimo metu pagal 6.9 lentelės duomenis buvo tikrinamas pateiktų apmokyti signalų skaičius. Kaip ir ankstesniais testavimo atvejais, *PMM* klasifikatoriaus tikslumas pablogėjo, kai signalų skaičius buvo mažas, o darbo laikai didėjo daugėjant signalų skaičiui. *DNT* klasifikatoriaus atpažinimo rezultatai buvo geresni. Tačiau, kai pateiktų signalų skaičius buvo penki, gauti rezultatai buvo prastesni. Pagal laiką jo darbas buvo maždaug pastovus ir tik truputį padidėdavo daugėjant signalų.

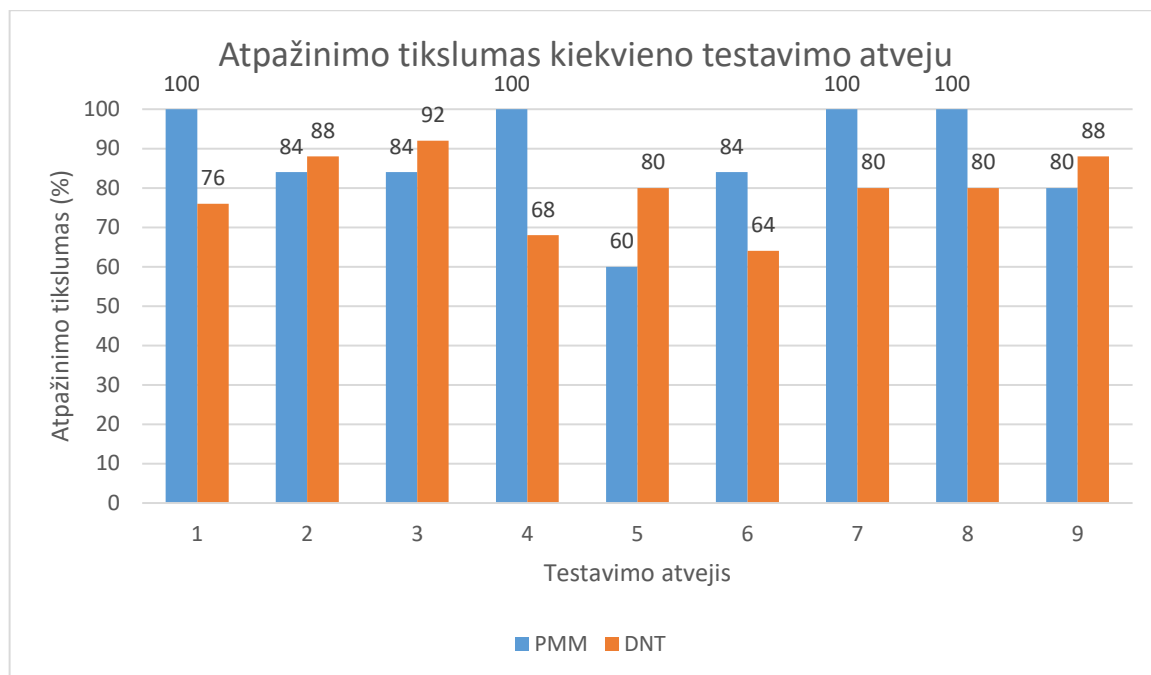
Vadinasi, iš viso buvo atlikta keturi šimtai penkiasdešimt programų testavimų. Pagal visus nagrinėtus testavimo atvejus buvo sudaryti du grafikai, apibendrinantys gautus rezultatus. Jie padės nustatyti ir pateikti galutines išvadas apie klasifikatorių galingumą ir atpažinimo tikslumą. Pirmas grafikas parodo laiko, per kurį buvo atliktas apmokymo ir atpažinimo darbas abiejose sistemose, vidurkius (6.1 grafikas).



6.1 grafikas. Atpažinimo sistemų laiko vidurkiai kiekvienu testavimo atveju

Pagal viršuje pateiktą grafiką galima daryti išvadas, kad *PMM* klasifikatoriaus darbo laikas yra stipriai kintantis, ypač keičiant pateiktų apmokymo signalų ir neuronų skaičius. Geriausiai tai įrodo antras, ketvirtas ir septintas testavimo atvejai. Žymiausi laikai, per kuriuos buvo atpažintas balsas, buvo penktame testavimo atvejyje, kai buvo tikrinamas apmokymo žingsnių skaičius. Pagal tai galima daryti išvadą, kad *PMM* klasifikatoriuje didelę įtaką daro tai, kiek signalų ir kiek neuronų buvo pateikta modelio apmokymo procesui. *DNT* klasifikatoriuje pagal grafiką laikas keičiasi nežymiai. Jis išlieka beveik pastovus visais testavimo atvejais. Tačiau tik kai apmokymo žingsnių skaičius buvo pakeistas pirmame, trečiame ir šeštame testavimo atvejuose, apmokymo ir atpažinimo laikai padidėdavo. Žymiausi laikai buvo septintame ir devintame testavimuose, kai buvo keičiami neuronų ir signalų skaičiai. Pagal tai galima nustatyti, kad *DNT* atpažinimo programoje apmokyti pateiktų signalų skaičius nedaro didelės įtakos ir jo darbo laiką labiausiai lemia tai, kiek apmokymo žingsnių jis turi padaryti.

6.2 grafike pateikti kiekvieno testavimo atvejo atpažinimo tikslumo duomenys. Pagal jį bus galima nustatyti, kaip teisingai veikia programos ir kiek jos sugeba atpažinti žmogų pagal jo balsą, keičiant pagrindinius klasifikatorių parametrus ir pateiktų apmokyti signalų skaičių (6.2 grafikas).



6.2 grafikas. Programų atpažinimo tikslumas kiekvienu testavimo atveju

6.2 grafike matyti, kad *PMM* tikslumas buvo šimtas procentų pirmame, ketvirtame, septintame ir aštuntame testavimo atvejuose, kai apmokymo signalų skaičius buvo pastovūs kintamieji. Mažiausias tikslumas, kai *PMM* sistema gebėdavo atpažinti žmogų pagal jo balsą, buvo devintame ir ypač penktame testavimo atvejuose, kai buvo keičiamas apmokymo signalų skaičius. Pagal tai galima daryti išvadas, kad *PMM* klasifikatoriaus tikslumas priklauso nuo to, kiek apmokymo signalų buvo pateikta. *DNT* atpažinimo sistemos teisingų rezultatų gavimo atvejai buvo prastesni negu *PMM*. Daugiausiai atvejų, kai programa gebėdavo atpažinti žmogų, buvo antrame, devintame ir labiausiai trečiame testavimo atvejuose. Tada buvo keičiami pagrindiniai klasifikatoriaus parametrai. Mažiausiai programa gebėdavo teisingai atpažinti žmogaus balsą ketvirtame ir šeštame testavimuose, kai buvo keičiamas apmokymo žingsnių skaičius su kitais parametrais. Vadinasi, *DNT* programai daugiausiai įtakos turi ne tai, kiek buvo pateikta signalų atpažinti, o kokie pagrindiniai kintamieji buvo nustatyti klasifikatoriuje.

6.10 lentelė. Bendras gautų rezultatų apibendrinimas kiekvienam klasifikatoriui

Klasifikatoriai	Bendras apmokymo ir atpažinimo laiko vidurkis (s)	Bendras atpažinimo tikslumas (%)
<i>PMM</i>	2,943	88
<i>DNT</i>	1,791	79,56

Bendras testavimo rezultatas – *PMM* ir *DNT* klasifikatorių veikimas ir atpažinimo tikslumas skiriasi. *DNT* klasifikatorių veikimui yra svarbiau pateiktų apmokyti signalų skaičius,

o atpažinimo tikslumui – apmokymo žingsnių ir neuronų skaičius. 6.10 lentelėje pateikti bendri duomenys apie klasifikatorių veikimą. Pagal bendrą apmokymo ir atpažinimo laiką *DNT* klasifikatorius greičiau atlieka savo darbą ir laikas visais atpažinimo atvejais žymiai nesikeisdavo. Lentelėje matosi, kad *PMM* atlikdavo savo darbą vidutiniškai 1,152 sekundes lėčiau už *DNT*. Tačiau bendras dirbtinio neuroninio tinklo klasifikatoriaus atpažinimo tikslumas yra prastesnis už paslėptąjį *Markovo* modelį. Jų skirtumo procentinė vertė yra lygi 8,44 procento. Buvo net keturi testavimo atvejai, kai *PMM* klasifikatorius šimtu procentų atpažindavo žmogų pagal jo balso *MSKK* požymius. *DNT* sistemos tikslumas neviršydavo 92 procentų ir vidutiniškai jo teisingo rezultato gavimas yra apie 80 procentų. Vadinasi, *PMM* parodė didesnę tikimybę gauti teisingą *DNT* klasifikatoriaus atsakymą.

4.2. Skyriaus apibendrinimas

Abi atpažinimo sistemos buvo testuotos ir palygintos viena su kita, naudojant penkis žmogaus balso variantus. Buvo atlikta apie keturi šimtai penkiasdešimt testavimų, siekiant nustatyti, kuris iš klasifikatorių geriausiai atlieka savo darbą. Testavimo metu, be atpažinimo tikslumo, nemažai dėmesio buvo skirta ir laikui, per kurį atliekamas modelio apmokymas ir pats balso atpažinimas. Pagal gautus rezultatus galima daryti išvadas, kad atpažinimo tikslumas ir darbo laiko trukmė kiekvienoje sistemoje priklauso nuo skirtingų kintamųjų.

PMM klasifikatoriuje didžiausią įtaką daro pateiktų atpažinti signalų skaičius. Kuo jų yra mažiau, tuo žmogaus balso atpažinimo tikslumas yra prastesnis. *DNT* modelis – atvirkščiai – jo tikslumas nepriklausydavo nuo apmokymo signalų skaičių, o labiau nuo nustatytų pagrindinių klasifikatorių parametrų. Testavimas ir programų palyginimas įrodo, kad abi sistemos sugeba atpažinti žmogų pagal jo balsą, kai visi parametrai yra ne per maži ir ne per dideli.

Nepaisant to, atpažinimo ir apmokymo laikas *DNT* modelyje yra mažesnis ir maždaug pastovus. *PMM* sistemoje, pateikiant daug apmokymo signalų, gali žymiai pailgėti programos darbas. Atpažinimo tikslumo skirtumas nėra labai didelis, lyginant abu klasifikatorius. *DNT* teisingo rezultato gavimas priklausydavo daugiau nuo pačių klasifikatoriaus parametrų. Jeigu jie buvo per maži, dažniausiai programa pateikdavo klaidingą atpažintą žmogaus vardą. *PMM* parodydavo geresnius rezultatus, o klaidingą atsakymą dažniausiai išvesdavo tuo atveju, jei apmokymų signalų skaičius buvo per mažas. Vadinasi, pagal visus pateiktus testavimo duomenis ir klasifikatorių palyginimus galima teigti, kad patikimesnė ir galingesnė yra *PMM*

žmogaus balso atpažinimo sistema. Jos žmogaus balso atpažinimo tikslumas, naudojant *MSKK* požymius, buvo aukštesnis negu *DNT*.

IŠVADOS

Magistro darbe analizuojant literatūrą buvo išskirti ir išanalizuoti balso atpažinimo metodai, algoritmai ir su tuo susijusios problemos. Įvertinus ir pritaikius surinktą teorinę informaciją, buvo sukurtos dvi žmogaus balso atpažinimo sistemos, kurios vėliau buvo testuojamos ir lyginamos viena su kita.

1. Atlikus **literatūros analizę, buvo nustatyti dažnai naudojami garso atpažinimo algoritmai, kylančios problemos ir jų sprendimo būdai**. Remiantis gautomis žiniomis, galima teigti, kad yra nemažai atpažinimo sistemų ir metodų, tačiau jie visi turi tam tikrų trūkumų, turinčių įtakos žmogaus balso identifikavimo tikslumui. Be to, šių metodų įvairovė pasunkina apsisprendimą, kuris iš jų yra geresnis ir patikimesnis. Tikslios ir gebančios identifikuoti žmogų pagal jo balsą programos poreikis yra aukštas. Esantys metodai yra beveik tikslūs ir patikimi, pavyzdžiui, paslėptieji *Markovo* modeliai, kurie dažniausiai ir naudojami skirtingose identifikavimo sistemose. Tačiau paskutinįjį dešimtmetį atsirado nauja ir galinga atpažinimo sistema – dirbtiniai neuronų tinklai, kurio tikslumas gali būti ne prastesnis už *PMM*. Dėl tokių klasifikatorių įvairovės naudotojams sunku nustatyti, kuris iš jų yra geresnis, jų nepalyginus. Nagrinėjant pateiktą literatūrą taip pat buvo nustatyta, kad geriausiai atpažinimui naudoti *MSKK* požymius, kurie geba pašalinti daug nereikalingų komponentų originaliame spektre, patobulinant gero rezultato gavimo tikslumą.

Pasiūlymas: siekiant sužinoti, kuri iš populiariausių pasirinktų atpažinimo sistemų yra patikimesnė, buvo nuspręsta jas išanalizuoti plačiau, testuoti ir palyginti gautus rezultatus.

2. Atpažinimo programoms sukurti buvo **plačiau išanalizuota pasirinktų identifikavimo sistemų kūrimo eiga ir veikimo principas**. Pagal pateiktą informaciją atpažinimo sistema atlieka nemažai veiksmų – nuo balso signalo filtravimo ir požymių išskyrimo iki galutinio atpažinto žmogaus vardo pateikimo. *PMM* klasifikatorius turi net dvi skirtingas struktūras – ergodinį ir grandininį metodus. Populiariausias iš jų yra grandininis metodas, kurio būsenos eina viena paskui kitą. Be to, yra trys parametrų įvertinimo algoritmai, naudojami *PMM*: tiesioginis, „forward-backward“ ir *Viterbio*. Giliojo mokymosi klasifikatorius taip pat turi daugybę skirtingų struktūrų, kurių kiekviena skirta tam tikrai užduočiai atlikti. Be to, jo mokymo procesas gali būti tiesioginis, atgalinis ar net hibridinis. Dėl daugybės skirtingų

klasifikatorių realizavimo būdų yra svarbu nustatyti, kurie šių algoritmų labiausiai tinkantys pateiktai užduočiai.

Pasiūlymas: visų algoritmų tikslas yra vienodas, tačiau skaičiavimo metodai, paskirtis ir galingumas yra skirtingi. Norint sukurti žmogaus balso atpažinimo sistemą, *PMM* klasifikatoriuje geriausia naudoti grandininę struktūrą, o parametrus įvertinti – *Viterbio* algoritmą, nes jis yra taikomas nuolatinei kalbai atpažinti. *DNT* sistemai realizuoti buvo pasirinktas mokymosi be mokytojo metodas. Balso atpažinimo proceso metu nėra žinoma, kokia turi būti išvestis, ir reikėtų padaryti taip, kad modelis pats paskirstytų požymius pagal kategorijas. Tada jis geriau gebės identifikuoti žmogų pagal balsą. Naudojant visus pasirinktus klasifikatorių realizavimo algoritmus galima kurti žmogaus balso atpažinimo sistemas.

3. Atliekant magistro darbo praktinę dalį, buvo **sukurtos dvi garso atpažinimo programos, naudojant pasirinktus atpažinimo metodus. Programos parašytos naudojant Python programavimo kalbą.** Panaudojus *MSKK* požymius ir *PMM*, *DNT* klasifikatorius buvo realizuotas dviejose programose, sugebančiose atpažinti žmogų pagal jo balsą. Abi programos buvo sukurtos *Spyder* platformoje, kuri leidžia lengviau atlikti įvairias skaičiavimo manipuliacijas. Tačiau *Python* ir *Spyder* turi tam tikrų trūkumų, pasunkinančių programų realizavimą. Tai yra daugybė skirtingų paketų versijų, kurios ne visos tinka tam tikrai programinei įrangai. Be to, pateikta programavimo kalba turi daugybę skirtingų bibliotekų, atliekančių panašias funkcijas. Dėl klasių ir komandų įvairovės *Python* programavime yra svarbu teisingas duomenų pateikimo formatas, kad kiekvienas iš jų atitiktų poreikius.

Pasiūlymas: Supaprastinus *Python* programavimo kalboje duomenų pateikimą į reikalingas klases ir sukūrus bendras bibliotekas, atliekančias tik tam tikras tikslas užduotis, sistemų realizavimas būtų žymiai lengvesnis. Be to, paketų versijų pritaikymas kiekvienai programinei įrangai suteiktų daugiau galimybių skirtingiems naudotojams. Tokiu būdu taikant „*Spyder*“ platformą, galima sukurti *PMM* ir *DNT* žmogaus balso atpažinimo programas daug paprasčiau.

4. Sukūrus programas buvo **atliktas jų eksperimentinis tyrimas.** Testavimo metu programoms buvo pateikti skirtingi pradiniai parametrai. Iš viso buvo atlikta keturi šimtai penkiasdešimt programų testavimų, per kuriuos buvo tikrinama, ar klasifikatorius sugeba atpažinti žmogaus balsą, bei apmokymo su atpažinimu darbo laiką.

Pagal gautus rezultatus buvo nustatyta, kad abu klasifikatoriai gali identifikuoti žmogų pagal jo balsą. Tačiau per daug mažinant jų pagrindinius parametrus, ypač *DNT* klasifikatoriuje, atpažinimo tikslumas tampa prastesnis. Toliau testuojant programas buvo nustatyta, kad *PMM* klasifikatorius žymiai priklauso nuo apmokymo signalų skaičių. Jeigu buvo pateikta per mažai garso signalų apmokyti, atpažinimo tikslumas mažėdavo. Kitų parametrų keitimas turėjo nedidelės įtakos klasifikatoriaus veikimui. *DNT* atpažinimo metodas – atvirkščiai – jo atpažinimo tikslumas nepriklausydavo nuo pateiktų signalų skaičių, bet daugiausiai įtakos jam turėjo pagrindinių parametrų keitimas. Pagal *DNT* programos testavimą buvo nustatyta, kad per maži neuronų ir apmokymo žingsnių skaičiai gali suklaidinti klasifikatorių. Apibendrinant visus testavimo duomenis galima daryti išvadą, kad pagal apmokymo ir atpažinimo laiką *PMM* klasifikatorius yra prastesnis ir nepastovus. Tačiau jo atpažinimo tikslumas buvo aukštesnis už *DNT* atpažinimo metodo.

Pasiūlymas: Toliau tobulinant programas ir sistemas, bus galima sukurti atpažinimo programą, kuri bet koku atveju identifikuotų žmogų. Sukūrus klasifikatorių patobulinimo algoritmą arba net sujungus *PMM* ir *DNT* metodus, žmogaus atpažinimo pagal jo balsą tikslumas gali žymiai pagerėti.

Sukurtos žmogaus balso atpažinimo sistemos leidžia pagal signalo *MSKK* požymius identifikuoti žmogų, naudojant du skirtingus klasifikavimo būdus. *PMM* ir *DNT* atpažinimo metodai laikomi vienais iš geriausių dėl savo tikslumo ir veikimo laiko. Atlikus jų eksperimentinį tyrimą galima teigti, kad šiuo metu geresnis žmogaus balso atpažinimo klasifikatorius yra paslėptasis *Markovo* modelis.

LITERATŪROS SĄRAŠAS

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., ... ir Zheng, X. (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems, p. 1–18. Prieiga per internetą: <<http://download.tensorflow.org/paper/whitepaper2015.pdf> />
2. Aparna, R., ir Chithra, P. (2017). Role of Windowing Techniques in Speech Signal Processing For Enhanced Signal Cryptography, *Advanced Engineering Research and Applications*, p. 446–458. Prieiga per internetą: <https://www.researchgate.net/publication/323127358_Role_of_Windowing_Techniques_in_Speech_Signal_Processing_For_Enhanced_Signal_Cryptography/>
3. Arora, S. J., ir Singh, R. P. (2012). Automatic Speech Recognition: A Review, *International Journal of Computer Applications*, vol. 60– No. 9, p. 34–44. Prieiga per internetą: <https://www.researchgate.net/publication/318749834_Automatic_Speech_Recognition_A_Review/>
4. Athira, G., ir Anjusha, V. K. (2017). Analysis of Feature Extraction Methods for Speech Recognition, *IJISSET - International Journal of Innovative Science, Engineering & Technology*, vol. 4, p. 328–333. Prieiga per internetą: <http://ijiset.com/vol4/v4s4/IJISSET_V4_I04_39.pdf>
5. Bhabad, S. S., ir Kharate, G. K. (2013). An Overview of Technical Progress in Speech Recognition, *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, p. 488–497;
6. Bhattacharjee, U. (2013). A Comparative Study Of LPCC And MFCC Features For The Recognition Of Assamese Phonemes, *International Journal of Engineering Research & Technology (IJERT)*, ISSN: 2278-0181, vol. 2, p. 1–6. Prieiga per internetą: <https://www.researchgate.net/publication/308749743_A_Comparative_Study_Of_LPCC_And_MFCC_Features_For_The_Recognition_Of_Assamese_Phonemes/>
7. Chauhan, H., Samal, S., ir Ghoshal, A. (2015). Voice recognition, *International Journal of Computer Science and Mobile Computing*, vol. 4, p. 296–301. Prieiga per internetą: <<https://ijcsmc.com/docs/papers/April2015/V4I4201563.pdf>>
8. Chauhan, V., Dvivedi, S., Karale, P., ir Potdar, S. M. (2016). Speech to text converter using gaussian mixture model (GMM), *International Research Journal of Engineering and Technology (IRJET)*, vol. 3, p. 160–164. Prieiga per internetą: <<https://www.irjet.net/archives/V3/i5/IRJET-V3I534.pdf>>

9. Chavan, R., ir Sable, G. (2013). An Overview of Speech Recognition Using HMM, *International Journal of Computer Science and Mobile Computing*, vol. 2, p. 233–238. Prieiga per internetą: [<https://www.researchgate.net/publication/335714660_An_Overview_of_Speech_Recognition_Using_HMM/>](https://www.researchgate.net/publication/335714660_An_Overview_of_Speech_Recognition_Using_HMM/)
10. Corvin, J. (2009). The auditory system, p. 1–16.
11. Cuzzocrea, A., Mumolo, E., ir Vercelli, G. (2017). Ergodic Hidden Markov Models for Workload Characterization Problems, p. 1–8. Prieiga per internetą: [<http://ksiresearch.org/seke/dms17paper/dms17paper_20.pdf/>](http://ksiresearch.org/seke/dms17paper/dms17paper_20.pdf/)
12. Dave, N. (2013). Feature Extraction Methods LPC, PLP and MFCC In Speech Recognition, *International Journal for Advance Research in Engineering and Technology*, vol.1, p. 1–5. Prieiga per internetą: [<https://www.researchgate.net/publication/261914482_Feature_extraction_methods_LPC_PLP_and_MFCC_in_speech_recognition/>](https://www.researchgate.net/publication/261914482_Feature_extraction_methods_LPC_PLP_and_MFCC_in_speech_recognition/)
13. De Cheveigne, A. (2005). Pitch perception models. In: Pitch, Springer, p. 169–233.
14. Deller, J. Jr. H., ir Hansen, J. (2005). Methods, Models, and Algorithms for Modern Speech Processing, *The Electrical Engineering Handbook*, p. 861–890;
15. Deng, L., Aksmanovic, M., ir Vu, C. F. J. (1994). Speech Recognition Using Hidden Markov Models with Polynomial Regression Functions as Nonstationary States, *IEEE Transactions On Speech And Audio Processing*, vol. 2, No. 4. Prieiga per internetą: [<https://ieeexplore.ieee.org/document/326610/>](https://ieeexplore.ieee.org/document/326610/)
16. Gales, M., ir Young, S. (2007). The Application of Hidden Markov Models in Speech Recognition, vol. 1, No. 3, p. 195–304. Prieiga per internetą: [<https://mi.eng.cam.ac.uk/~mjfg/mjfg_NOW.pdf/>](https://mi.eng.cam.ac.uk/~mjfg/mjfg_NOW.pdf/)
17. Gulzar, T., Singh, A., ir Sharma, S. (2014). Comparative Analysis of LPCC, MFCC and BFCC for the Recognition of Hindi Words using Artificial Neural Networks, *International Journal of Computer Applications*, vol. 101, p. 22–27. Prieiga per internetą: [<https://www.academia.edu/9844071/Comparative_Analysis_of_LPCC_MFCC_and_BFCC_for_the_Recognition_of_Hindi_Words_using_Artificial_Neural_Networks/>](https://www.academia.edu/9844071/Comparative_Analysis_of_LPCC_MFCC_and_BFCC_for_the_Recognition_of_Hindi_Words_using_Artificial_Neural_Networks/)
18. Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., Sainath, T., ... ir Vanhoucke, V. (2012). Deep Neural Networks for Acoustic Modeling in Speech Recognition: The shared views of four research groups, *IEEE Signal Processing Magazine*, vol. 29, p. 82–97. Prieiga per internetą: [<https://ieeexplore.ieee.org/document/6296526/>](https://ieeexplore.ieee.org/document/6296526/)
19. hmmlearn Tutorial (2010). Prieiga per internetą: [<https://hmmlearn.readthedocs.io/en/latest/tutorial.html/>](https://hmmlearn.readthedocs.io/en/latest/tutorial.html/)

20. Yen, J. (2009). Wavelet for Acoustics, *Technical Report R98942097*, p. 1–19. Prieiga per internetą: <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=CB9763324B4B8F41AA02B14AB7E391A6?doi=10.1.1.695.898&rep=rep1&type=pdf/>>
21. Yu, D., Deng, L., ir Dahl, G. (2010). Roles of Pre-Training and Fine-Tuning in Context-Dependent DBN-HMMs for Real-World Speech Recognition, NIPS Workshop on Deep Learning and Unsupervised Feature Learning, p. 1–8. Prieiga per internetą: <https://pdfs.semanticscholar.org/ecd4/bc32bb2717c96f76dd100fcd1255a07bd656.pdf/>>
22. Jelinek, F. (1976). Continuous Speech Recognition by Statistical Methods, *Proc. IEEE*, vol. 64, p. 532–536. Prieiga per internetą: <https://ieeexplore.ieee.org/document/1454428/>>
23. Juang, B. H., ir Rabiner, L. R. (1991). Hidden Markov Models for Speech Recognition, *Technometrics*, vol. 33, No 3, p. 251–272; Prieiga per internetą: <https://pdfs.semanticscholar.org/df68/2aa90fbbbf665a8b273a57ca87d6cea9ff99.pdf/>>
24. Juang, B. H., ir Rabiner, L. R. (2004). Automatic Speech Recognition – A Brief History of the Technology Development, Georgia Institute of Technology, Atlanta, p. 1–24. Prieiga per internetą: <https://pdfs.semanticscholar.org/1d19/9099a2f4f8749c7e10480b29f5adaecad4a1.pdf?ga=2.100058206.134887037.1587407722-992585985.1543775917/>>
25. Kamarauskas, J. (2009). Asmens atpažinimas pagal balsą, *VGTU leidykla TECHNIKA 1610-M mokslo literatūros knyga*, ISBN 978-9955-28-422-2, p. 23–80. Prieiga per internetą: https://www.mii.lt/files/doc/lt/doktorantura/apgintos_disertacijos/mii_dis_2009_kamarauskas.pdf/>
26. Kare, C.B., ir Navale, V.S. (2015). Speech recognition by Dynamic Time Warping, *IOSR Journal of Electronics and Communication Engineering*, p. 12–16. Prieiga per internetą: <https://pdfs.semanticscholar.org/932c/a12b793440e8b95f7772c5541c8bd78047d7.pdf?ga=2.121927310.44135094.1587658114-992585985.1543775917/>>
27. Kaur, R., ir Pushpalata, S. (2015). Speech Recognition System with Speaker Verification using HMM, LPC & MFCC, *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 4, p. 2333–2339. Prieiga per internetą: <http://ijarcet.org/wp-content/uploads/IJARCET-VOL-4-ISSUE-5-2333-2339.pdf/>>
28. Kekre, H. B., Athavale, A. A., ir Sharma, G. J. (2011). Speech Recognition using Vector Quantization, *International Conference and Workshop on Emerging Trends in Technology (ICWET 2011)*, p. 400–403. Prieiga per internetą: <https://dl.acm.org/doi/10.1145/1980022.1980107/>>

29. Levinson, S. E., Rabiner, L. R., ir Sondhi, M. M. (1983). "An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition", *The Bell System Technical Journal*, vol. 62, p. 1035–1074. Prieiga per internetą: <<https://ieeexplore.ieee.org/document/6768244/>>
30. LibROSA Docs (2018). Prieiga per internetą: <<https://librosa.github.io/librosa/>>
31. Lippmann, R. P. (1990). Review of Neural Networks for Speech Recognition, Readings in Speech Recognition“, p. 374–392.
32. Loveimi, E., Ahadi, S. M., Drugman, T., ir Loveymi, S. (2013). On the Importance of Pre-emphasis and Window Shape in Phase-based Speech Recognition, p. 160–167. Prieiga per internetą:
<https://www.researchgate.net/publication/299710314_On_the_Importance_of_Pre-emphasis_and_Window_Shape_in_Phase-Based_Speech_Recognition/>
33. NumPy (2018). Prieiga per internetą: <<https://numpy.org/>>
34. Pyplot tutorial (2012). Prieiga per internetą:
<<https://matplotlib.org/tutorials/introductory/pyplot.html/>>
35. Python Documentation contents (2019). Prieiga per internetą: <<https://docs.python.org/3/contents.html> />
36. Poularikas, A. D. (1999). The handbook formulas and tables for signal processing – Chapter 7: Windows, *CRC Press LLC*, ISBN 0-8493-8579-2. Prieiga per internetą:
<<http://www.mongroupsyzdney1.com/3.pdf/>>
37. Ramljak, M., Stella, M., ir Šaric, M. (2013). Front-End Signal Processing for Speech Recognition, Recent Advances in Circuits, Systems, Telecommunications and Control, ISBN: 978-960-474-341-4, p. 102–106. Prieiga per internetą: <<http://www.wseas.us/e-library/conferences/2013/Paris/CCTC/CCTC-15.pdf/>>
38. Sakoe, H., ir Chiba S. (1978). Dynamic Programming Algorithm Quantization for Spoken Word Recognition, *IEEE Trans. Acoustics, Speech and Signal Proc.*, vol. ASSP-26, No. 1, p. 43–49. Prieiga per internetą:
<https://www.irit.fr/~Julien.Pinquier/Docs/TP_MABS/res/dtw-sakoe-chiba78.pdf/>
39. Sakran, A. E., Abdou, S. M., Salah, E. H., ir Rashvan, M. (2017). A Review: Automatic Speech Segmentation, *International Journal of Computer Science and Mobile Computing (IJCSMC)*, vol. 6, p. 308–315. Prieiga per internetą:
<https://www.researchgate.net/publication/317339722_A_Review_Automatic_Speech_Segmentation/>

40. Schickhofer, L. (2017). Sound Generation and Propagation in the Human Upper Airways, *Technical Reports from Royal Institute of Technology*, p. 2–64. Prieiga per internetą: <https://kth.diva-portal.org/smash/get/diva2:1088462/FULLTEXT01.pdf>
41. Shafee, S., ir Anuradha, B. (2015). Speaker Identification and Spoken word Recognition in Noisy Environment, *International Journal of Applied Engineering Research ISSN 09734562*, vol. 10, No. 18, p. 43–49. Prieiga per internetą: https://www.irit.fr/~Julien.Pinquier/Docs/TP_MABS/res/dtw-sakoe-chiba78.pdf
42. Singh, P. P., ir Rani, P. (2014). An Approach to Extract Feature using MFCC, *IOSR Journal of Engineering (IOSRJEN)*, vol. 4, p. 21–25. Prieiga per internetą: [http://iosrjen.org/Papers/vol4_issue8%20\(part-1\)/D04812125.pdf](http://iosrjen.org/Papers/vol4_issue8%20(part-1)/D04812125.pdf)
43. Spyder Docs (2018). Prieiga per internetą: <https://docs.spyder-ide.org/overview.html>
44. Stamp, M. (2018). A Revealing Introduction to Hidden Markov Models, Department of Computer Science, San Jose State University, p. 1–21. Prieiga per internetą: https://www.researchgate.net/publication/288957333_A_revealing_introduction_to_hidden_markov_models
45. Station Apparatus (1969), *Fundamentals of Telephone Communication Systems*, Western Electrical Company, sk. 2.2. Prieiga per internetą: <http://etler.com/docs/WEC/Fundamentals/files/2-StationApparatus.pdf>
46. Stuttle, M. N. (2003). A Gaussian Mixture Model Spectral Representation for Speech Recognition, *Dissertation for the degree of Doctor of Philosophy*, p. 1–148. Prieiga per internetą: http://mi.eng.cam.ac.uk/~mjfg/thesis_mns25.pdf
47. Sun, X., Mianaga, Y., ir Sai, B. (2014). Dynamic Time Warping for Speech Recognition with Training Part to Reduce the Computation, *Journal of Signal Processing*, vol. 18, No. 2, p. 89–96. Prieiga per internetą: https://www.jstage.jst.go.jp/article/jsp/18/2/18_89/_pdf
48. Tebelskis, J. (1995). Speech Recognition using Neural Networks, p. 1–180. Prieiga per internetą: <http://isl.anthropomatik.kit.edu/pdf/Tebelskis1995.pdf>
49. TFLearn (2018). TFLearn: Deep learning library featuring a higher-level API for TensorFlow. Prieiga per internetą: <http://tflearn.org/>
50. Theodoridis, S., ir Koutroumbas, K. (2003). *Pattern Recognition: Second Edition*, Elsevier Academic Press. Prieiga per internetą: <http://www.cs.ukzn.ac.za/~sviriri/Books/Machine-Learning-Pattern-Recognition/book3.pdf>
51. Trivedi, P. A. (2014). Introduction to Various Algorithms of Speech Recognition: Hidden Markov Model, Dynamic Time Warping and Artificial Neural Networks, *International*

Journal of Engineering Development and Research, vol. 2, p. 3590-3596. Prieiga per internetą: <<https://www.ijedr.org/papers/IJEDR1404035.pdf>>

52. Vacher, M., Fleury, A., Portet, F., Serignat, J., ir Noury, N. (2010). Complete Sound and Speech Recognition System for Health Smart Homes: Application to the Recognition of Activities of Daily Living, Domenico Campolo. *New Developments in Biomedical Engineering*, p. 645–673. Prieiga per internetą: <[https://hal.archives-ouvertes.fr/hal-00422576/PDF/Complete_Sound_and_Speech_Recognition_System_for_Health_Smart_Homes- Application to the Recognition of Activities of Daily Living.pdf](https://hal.archives-ouvertes.fr/hal-00422576/PDF/Complete_Sound_and_Speech_Recognition_System_for_Health_Smart_Homes-Application_to_the_Recognition_of_Activities_of_Daily_Living.pdf)>
53. Vaibel, A., ir Lee, K. (1990). *Readings in Speech Recognition*, ISBN 1-55860-124-4. Prieiga per internetą: <https://books.google.lt/books/about/Readings_in_Speech_Recognition.html?id=IJoOLVYD8KQC&redir_esc=y>
54. Vijoyo, T., ir Vijoyo, S. (2011). Speech Recognition Using Linear Predictive Coding and Artificial Neural Network for Controlling Movement of Mobile Robot, *2011 International Conference on Information and Electronics Engineering*, vol. 6, p. 179–183. Prieiga per internetą: <<http://www.ipcsit.com/vol6/36-E091.pdf>>
55. Zhang, S. (2014). *Structured Support Vector Machines for Speech Recognition*, Dissertation for the degree of Doctor of Philosophy, p. 54–61. Prieiga per internetą: <http://mi.eng.cam.ac.uk/~mjfg/thesis_sxz20.pdf>

SUTIKIMAS DĖL ASMENS DUOMENŲ NAUDOJIMO

2020-05-25

(Data)

Šiuo sutikimu aš, Artūras Krivenkis (toliau – Duomenų subjektas) sutinku, kad Vilniaus Gedimino technikos universitetas, juridinio asmens kodas 111950243, adresas Saulėtekio al. 11, LT-10223 Vilnius (toliau – Duomenų valdytojas), tvarkytų mano asmens duomenis kitų studentų mokymosi tikslu. T. y. tvarkytų (*pažymėkite tinkamą*):

- ☐ vardą, pavardę, bakalauro baigiamąjį darbą;
- ☐ bakalauro baigiamąjį darbą, nenurodant vardo, pavardės;
- ☒ vardą, pavardę, magistro baigiamąjį darbą;
- ☐ magistro baigiamąjį darbą nenurodant vardo, pavardės.

Šiuo tikslu tvarkomų asmens duomenų Duomenų valdytojas neperduos jokiems tretiesiems asmenims, studentams su baigiamaisiais darbais bus leidžiama susipažinti vidinėje informacinėje sistemoje. Duomenų subjekto asmens duomenys šiuo tikslu bus naudojami ne ilgiau nei 5 metai.

Šiuo sutikimu Duomenų subjektas patvirtina, kad yra supažindintas su šiomis teisėmis:

- Susipažinti su savo duomenimis ir kaip jie yra tvarkomi (teisė susipažinti);
- Reikalauti ištaisyti arba, atsižvelgiant į asmens duomenų tvarkymo tikslus papildyti asmens neišsamius asmens duomenis (teisė ištaisyti);
- Savo duomenis sunaikinti arba sustabdyti savo duomenų tvarkymo veiksmus (išskyrus saugojimą) (teisė sunaikinti ir teisė „būti pamirštam“);
- Reikalauti, kad asmens duomenų valdytojas apribotų asmens duomenų tvarkymą (teisė apriboti);
- Teise į duomenų perkėlimą (teisė perkelti);
- Nesutikti, kad būtų tvarkomi asmens duomenys, kai šie duomenys tvarkomi ar ketinami tvarkyti kitais tikslais;
- Pateikti skundą Valstybinei duomenų apsaugos inspekcijai;

Duomenų subjektas turi teisę bet kuriuo metu atšaukti savo sutikimą.

Artūras Krivenkis

[Duomenų subjekto vardas, pavardė, parašas]

Duomenų valdytojo rekvizitai:
Vilniaus Gedimino technikos universitetas
Juridinio asmens kodas: 111950243
Adresas: Saulėtekio al. 11, LT-10223 Vilnius
Tel. (8 5) 274 5030
Faks. (8 5) 270 0112
El. paštas: vgtu@vgtu.lt

PVM mokėtojo kodas: LT119502413

Duomenų apsaugos pareigūno tel. (8 5) 251 2191, el. paštas: dap@vgtu.lt