VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS INSTITUTAS
KOMPIUTERINIO IR DUOMENŲ MODELIAVIMO KATEDRA

Baigiamasis magistro darbas

Ramano spektrų klasifikacija naudojant konvoliucinius neuroninius tinklus

Atliko:
Audrius Mečionis parašas

Vadovas:
dr. Valdas Rapševičius

Vilnius
2019

Turinys

Santrauka

Šiame darbe apžvelgiami Ramano sklaidos spektroskopija gautų duomenų analizės metodai. Eksperimentinės darbo dalies tikslas - sukurti duomenų klasifikavimo modelį, gebantį patikimai atpažinti įvairių rūšių mėsos Ramano sklaidos spektrų klases.

Darbo metu buvo atsižvelgta į egzistuojančius šios srities tyrimus bei panašios problematikos uždavinių sprendimo būdus. Eksperimentinės dalies metu įvertinti keli skirtingi klasifikavimo metodai, palygintas jų tikslumas. Duomenų klasifikatoriai sukurti naudojant atraminių vektorių mašinų algoritmą, dirbtinius neuroninius tinklus bei konvoliucinius neuroninius tinklus.

Aukštas tikslumas buvo pasiektas su visais tirtais klasifikavimo metodais. Geriausią rezultatą pavyko gauti naudojant dirbtinių neuroninių tinklų metodą, kuris siekė 96.85 % tikslumą.

Summary

Raman spectra classification using convolutional neural networks

Raman scattering is the inelastic scattering of monochromatic light in a material, which changes the radiation frequency due to the interaction of light quanta and substance molecules. Raman spectroscopy studies are used to analyze various materials, studying the properties of the material and the structure of the molecules. This method is fast, does not require any preparation of the test substance and is perfect for recognizing various chemical derivatives in the material [25].

It is becoming increasingly popular to use various machine learning techniques for studying Raman spectroscopy data, especially the use of classification methods when attempting to determine the type of test substance. Typical data classification algorithms, except for neural networks that use Raman's spectroscopy data, require nontrivial data preparation prior to model training. Data preparation often consists of steps such as dimensionality reduction and baseline correction [9].

The goal of this research work is to adapt the necessary data preparation methods and to create a classification model for Raman spectroscopy data of various types of meat, capable of reliably recognizing the classes of data. The tasks of the work are to get acquainted with the existing methods of Raman spectrum analysis, to investigate the mostly used machine learning methods and to select the most appropriate algorithm that could be used for future classification of this type of data.

During the research work the Raman spectrum data type identification experiments were carried out using a support vector machine algorithm, artificial neural networks, and convolutional neural networks. High accuracy was achieved with all the methods studied. The best results were obtained using the artificial neural network method. By this method, the classification model has reached a precision of 96.85%. It was also found that the neural network based models works well without applying baseline correction.

Įvadas

Ramano sklaida - tai šviesos išsklaidymas medžiagoje, kurio metu pakinta spinduliuotės dažnis. Ramano sklaidos spektrų tyrimais naudojamasi įvairių medžiagų analizėje, tiriant medžiagos savybes bei molekulių sandarą. Šis metodas yra greitas, nereikalauja išankstinio tiriamos medžiagos paruošimo ir puikiai tinka atpažinti įvairius cheminius darinius medžiagoje [25].

Ramano sklaidos spektroskopijos duomenis tirti taikant kompiuterio mokymosi metodus tampa vis populiariau, ypač naudojant įvairius klasifikavimo metodus ir bandant nustatyti tiriamosios medžiagos tipą. Tipiniams duomenų klasifikavimo algoritmams, išskyrus neuroninius tinklus, naudojantiems Ramano sklaidos duomenis reikalingas netrivialus duomenų apdorojimas prieš apmokant modelį. Duomenų paruošimas dažnai susideda iš tokių žingsnių, kaip dimensijų mažinimas ir bazinės linijos korekcija (angl. *baseline correction*) [9].

Šio darbo tikslas - pritaikyti reikalingus duomenų apdorojimo metodus ir sukurti klasifikavimo modelį įvairių rūšių mėsos Ramano sklaidos spektroskopijos duomenims, gebantį patikimai atpažinti tiriamų duomenų klases. Nustatyti darbo uždaviniai - susipažinti su esamais Ramano spektrų duomenų analizės metodais, ištirti dažniausiai taikomus kompiuterio mokymosi algoritmus ir išrinkti tinkamiausią metodą, kuris galėtų būti naudojamas ateityje klasifikuojant tokio tipo duomenis.

Mokslo tiriamojo darbo metu buvo pritaikyti Ramano spektrų duomenų analizės metodai ir sukurtas atraminių vektorių mašinų klasifikavimo modelis, patikrintas jo tikslumas. Tirtam duomenų rinkiniui klasifikuoti buvo pasiektas 81.45% tikslumas. Šio darbo teoriniai skyriai susiję su Ramano spektroskopija, kompiuterio mokymuisi, atraminių vektorių mašinų metodu buvo paimti iš mokslo tiriamojo darbo projekto. Eksperimentinė darbo dalis buvo perdaryta, nes gautas papildomas kiekis duomenų. Baigiamojo magistro darbo metu buvo nuspręsta papildomai sukurti duomenų klasifikatorius panaudojant dirbtinių neuronių tinklų bei konvoliucinių neuroninių tinklų metodus ir siekta nustatyti galimus šių modelių pranašumus ir trūkumus.

Baigiamojo darbo metu buvo atlikti duomenų tipo atpažinimo eksperimentai naudojant pasirinktus klasifikavimo algoritmus. Aukštas tikslumas buvo pasiektas su visais tirtais metodais. Geriausi rezultatai buvo gauti naudojant dirbtinių neuroninių tinklų metodą. Šiuo metodu kurtas klasifikatorius pasiekė 96.85% tikslumą. Taip pat nustatyta, kad neuroniniais tinklais grįsti modeliai gerai veikia neatlikus bazinės linijos korekcijos.

Tolimesniuose skyriuose aprašyti dažniausiai naudojami kompiuterio mokymosi metodai Ramano spektroskopijos duomenų analizėje. Taip pat apžvelgti pagrindiniai duomenų paruošimo būdai. Eksperimentinėje dalyje aprašomi duomenų analizės eksperimentai kuriant duomenų klasifikatorius, modelių architektūros parinkimas bei optimaliausių mokymosi parametrų paieška.

1. Teorinis įvadas

1.1. Ramano spektroskopija

Ramano sklaida - tai monochromatinės šviesos išsklaidymas medžiagoje, kurio metu pakinta spinduliuotės dažnis dėl šviesos kvantų ir medžiagos molekulių sąveikos. Šviesos kvantai susiduria su įvairių virpesių energijos būsenų molekulėmis ir atiduoda arba netenka dalies savo energijos. Nors molekulė ir gauna energijos iš krintančiosios šviesos, bet ji nesužadinama iki tam tikro energijos lygmens, o tik sklindant bangai įgyja tam tikrą perteklinę energiją. Prasklidus bangai, molekulė grįžta į buvusį energijos lygmenį [25].

Ramano spektras - tai Ramano sklaidos intensyvumo priklausomybė nuo dažnių skirtumo, gauto lyginant su žadinančiąja spinduliuote, išreikšto bangos skaičiumi (cm^{-1}). Ramano sklaidos spektro ir jį papildančių ultravioletinio bei infraraudonojo spektrų tyrimais naudojamasi kokybinėje ir kiekybinėje sudėtingų mišinių analizėje, tiriant molekulių sandarą ir jų struktūrą. Skirtingos cheminės sandaros molekulės turi tik joms būdingus Ramano sklaidos spektrus (molekulinius pirštų atspaudus), todėl ši metodika puikiai tinka atpažinti įvairius cheminius darinius. Taip pat šis metodas yra greitas ir nereikalauja išankstinio tiriamos medžiagos paruošimo. Cheminė sudėtis gali būti nustatyta nekontaktiniu ir nedestrukciniu būdu, pavyzdžiui tiriamai medžiagai esant pakuotėje ar inde.

Vykstant šviesos sklaidai, molekulės atomų virpesių energija padidėja arba sumažėja. Dažnio poslinkis atitinka tam tikrą molekulės virpesinį dažnį. Ramano sklaidos procesas labai mažai efektyvus - tik vienas iš maždaug 10^6 fotonų virsta Ramano fotonu. Ramano sklaidos intensyvumas labai priklauso nuo žadinančiosios lazerio spinduliuotės dažnio, todėl naudojami mažesnio bangos ilgio lazeriai.

1.2. Kompiuterio mokymasis

Kompiuterio mokymasis (angl. *machine learning*) - tai duomenų mokslo sritis, kuri naudoja statistinius metodus leidžiančius kompiuteriams pagal pateiktus duomenis numatyti galimas tendencijas, prognozuoti sistemų elgesį ar būsimus rezultatus. Dažniausiai išskiriamos trys kompiuterio mokymosi šakos:

- Prižiūrimas mokymasis (angl. *supervised learning*) - tai kompiuterio mokymosi šaka, kurioje algoritmai stengiasi susieti duomenų įvesties ir išvesties kintamuosius bei atrasti dėsnį, kuriuo būtų galima bet kuriai įvesčiai atrasti išvestį. Tai plačiausiai naudojama kompiuterio mokymosi sritis, kurios metodai dar skirstomi į klasifikavimo ir regresijos algoritmų tipus. Regresijos tipo metodai leidžia vieno kintamojo reikšmes prognozuoti pagal kitų kintamųjų reikšmes. Klasifikavimo algoritmai pagal pateiktus tiriamo objekto duomenis bando priskirti jį kokiai nors klasei.

- Neprižiūrimas mokymasis (angl. *unsupervised learning*) - tai kompiuterio mokymosi šaka, kai žinomos tik duomenų įvestys ir nežinomi išvesties kintamieji. Šio metodo tikslas sužinoti kuo daugiau informacijos apie turimus duomenis, ištirti struktūrą bei duomenų pasiskirstymą, tankį. Dažnai šiuo metodu yra ieškomi duomenų klasteriai, išgaunami esminiai atributai, duomenų apibendrinimas.

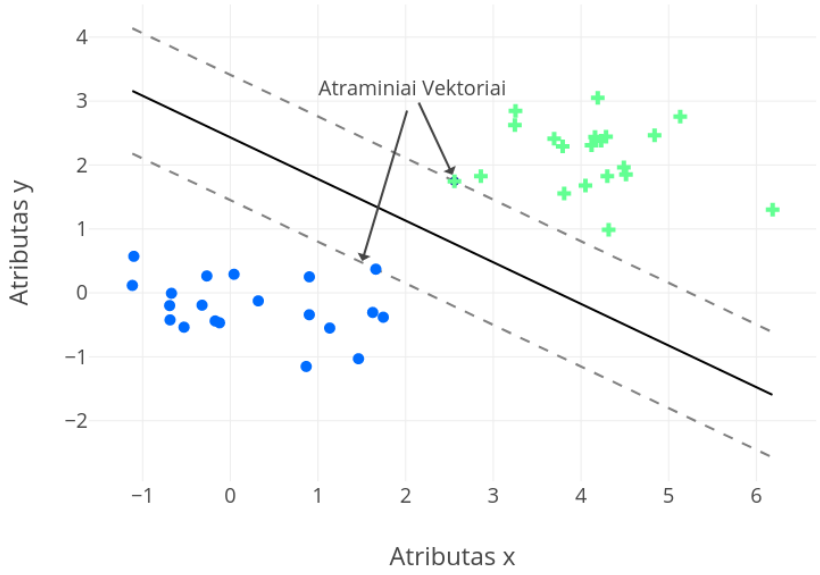
- Mokymasis su pastiprinimu (angl. *reinforcement learning*) - tai kompiuterio mokymosi šaka,

kurioje algoritmai stengiasi išgauti idealų sistemos elgesį pagal iš anksto nustatytus parametrus.

Šiame darbe bus naudojami tik prižiūrimo kompiuterio mokymosi algoritmai, nes tiriamų duomenų tipai bei klasės yra iš anksto žinomi. Toliau esančiuose skyreliuose bus aptarti dažniausiai naudojami kompiuterio mokymosi algoritmai naudojami Ramano spektroskopijos duomenų klasifikavimui atlikti.

1.3. Atraminių vektorių mašinos

Atraminių vektorių mašinos (angl. *support vector machines*, sutr. SVM) yra prižiūrimas kompiuterio mokymosi metodas, plačiai naudojamas klasifikavimo bei regresijos uždaviniams spręsti. Šis klasifikavimo metodas pasitvirtino savo efektyvumu ir pastaruoju metu tampa vis dažniau naudojamas greta neuroninių tinklų. SVM tikslas yra surasti optimalią hiperplokštumą (angl. *hyperplane*), kuri geriausiai atskirtų analizuojamus pažymėtus n-mačius duomenis [6]. Jeigu tiriamų duomenų įrašai gali būti vaizduojami kaip n-dimensinis vektorius, tai atraminių vektorių mašinų algoritmas stengsis rasti (n-1)-dimensinę hiperplokštumą, kuri atskirtų tiriamas klases ir kurios atstumas nuo tiriamų klasių artimiausių taškų yra didžiausias. SVM klasifikatoriaus veikimo principas naudojant dvimačius duomenis pavaizduotas 1 pav. Šis metodas efektyvus klasifikuojant didelio dimensijų skaičiaus duomenis, taip pat tais atvejais, kai požymių kiekis duomenų rinkinyje yra didesnis už pavyzdžių kiekį.



1 pav. Iliustracijoje pavaizduotas SVM algoritmo dvimačių duomenų klasifikacijos veikimo principas. Mėlyni ir žali taškai vaizduoja skirtingas duomenų klases, juoda linija tarp jų - algoritmo sprendimo ribą (angl. *decision boundary*), pagal kurią klasifikatorius suskirsto duomenis į klases.

Pirminis atraminių vektorių mašinų metodas buvo skirtas tiesinėms problemoms spręsti [20]. Vėliau buvo patobulintas ir pritaikytas netiesiniams uždaviniams panaudojant branduolio funkciją (angl. *kernel function*). Šios funkcijos naudojimas nepakeičia pačio metodo veikimo principų, tačiau leidžia suformuoti transformuotą vektorinę erdvę, kurioje būtų įmanomas tiesinis duomenų

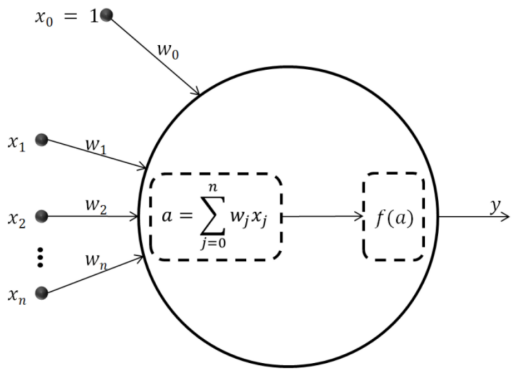
atskyrimas grįžus prie originalios daugiadimensinės parametrų erdvės. Populiariausios branduolio funkcijos, naudojamos praktikoje, yra tiesinė, polinominė, radialinės bazės (sutr. RBF), sigmoidinė. Gali būti naudojamos ir kitos, specialiai probleminei sričiai sukurtos funkcijos.

1.4. Dirbtiniai neuroniniai tinklai

Dirbtinis neuroninis tinklas (angl. *artificial neural network*, sutr. DNT) - tai struktūra, sudaryta iš daugelio nesudėtingų tarpusavyje sujungtų elementų (dirbtinių neuronų). Neuroninių tinklų algoritmai sukurti imituojant gyvų organizmų smegenyse vykstančius procesus, biologinės nervų sistemos veikimą bei savybes mokytis, prisitaikyti ir adaptuotis. Nors DNT pradėti tyrinėti dar 20 a. šeštajame dešimtmetyje, iki pat devintojo dešimtmečio vidurio jie nebuvo plačiai naudojami. Tik išradus greitus ir galingus mokymo mechanizmus DNT galėjo spręsti realius uždavinius [1]. Šiuo metu DNT vis plačiau taikomi duomenų klasifikavimo, klasterizavimo, funkcijų aproksimavimo, prognozavimo, optimizavimo, vizualizavimo uždaviniams spręsti [26].

1.4.1. Dirbtinio neurono modelis

Dirbtiniai neuroniniai tinklai pradėti tyrinėti kaip biologinių neuroninių sistemų modelis. Praėjusio amžiaus ketvirtajame dešimtmetyje dviejų amerikiečių mokslininkų (*W. McCulloch, W. Pitts*) buvo pasiūlytas dirbtinio neurono modelis. Šis modelis, atvaizduotas 2 pav. rėmėsi biologinio neurono sandara.



2 pav. Iliustracijoje pavaizduotas dirbtinio neurono modelis.

Dirbtinio neurono modelį galima suskirstyti į tris pagrindines dalis [11]:

1. **Įėjimai**. Dirbtinis neuronas turi keletą įėjimo reikšmių. Kiekviena įėjimo jungtis $x_1, x_2, ..., x_n$ turi savo perdavimo koeficientą (svorį) $w_1, w_2, ..., w_n$, kurie nurodo kiekvieno įėjimo svarbą. Šalia įėjimų dar yra slenksčio reikšmė w_0 (angl. *bias*), kuri skirta sustiprinti arba pasilpninti gaunamą signalą. Dažniausiai įėjimų ir svorių reikšmės yra realieji skaičiai.

2. **Sužadinimo signalas**. Skaičiuojama įėjimų ir svorių reikšmių sandaugų suma:

$$a = w_1 x_1 + w2 + x2 + ... + w_n x_n + w_0 = \sum_{j=1}^{n} w_j x_j + w_0. \tag{1.1}$$

Galima sutrumpinti (1.1) pridėjus visada pastovią reikšmę turintį nulinį iėjimą, $x_0 = 1$, ir užrašyti formulę taip:

$$a = \sum_{j=0}^{n} w_j x_j. \tag{1.2}$$

3. **Išėjimas**. Neurono išėjimą apibūdina aktyvavimo funkcija:

$$y = f(a) = f(\sum_{j=0}^{n} w_j x_j). \tag{1.3}$$

Neurono išėjimo reikšmė yra skaičiuojama sužadinimo signalui pritaikius aktyvacijos funkciją (1.3). Dirbtinio neurono modelyje gali būti naudojamos įvairios aktyvacijos funkcijos. Tai priklauso nuo konkretaus sprendžiamo uždavinio. Naudojant netiesines funkcijas ir jų kombinacijas neuronų tinklas sugeba apibrėžti sudėtingesnius netiesinius modelius [26]. Dažniausiai naudojamos aktyvavimo funkcijos yra (iliustruotos 3 pav.):

- Sigmoidinė:

$$f(x) = 1/(1 + e^{-x}). \tag{1.4}$$

Ši funkcija sukuria netiesiškumą dirbtiniame neuroniniame tinkle transformuodama išėjimo reikšmę į $(0; 1)$ intervalą. Dažnai naudojama tikimybei modeliuoti dėl patogaus išvesties intervalo. Pagrindinis šios funkcijos minusas - neįmanoma išlaikyti santykinių atstumų tarp plataus spektro įėjimo reikšmių. Pasiekus tam tikrą etapą, tinklo mokymosi greitis sulėtėja, nes net ir žymus įvesties reikšmių pokytis sukelia mažą išėjimo vertės pasikeitimą.

- Hiberbolinė tangento:

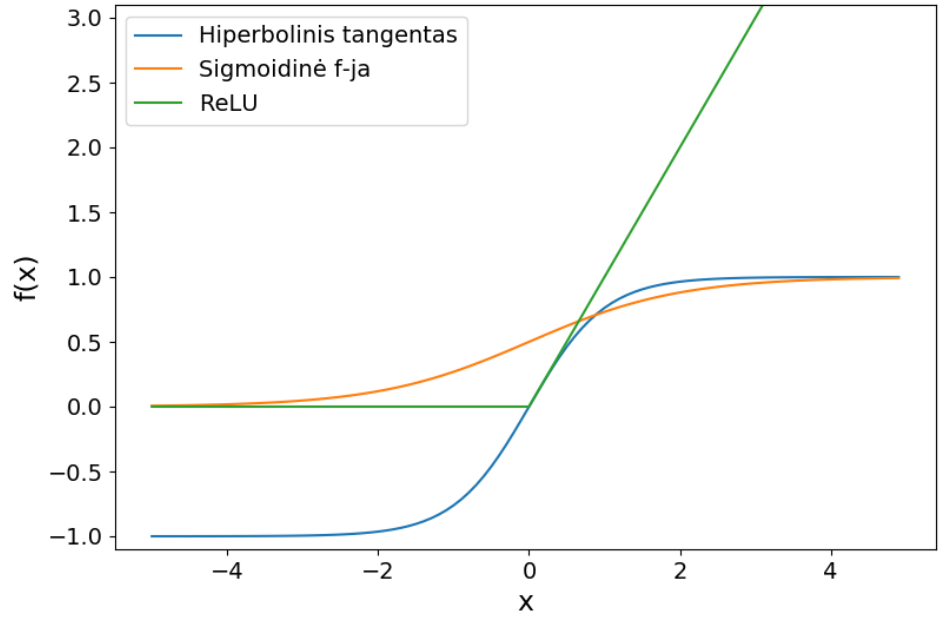
$$f(x) = tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{1.5}$$

Ši funkcija yra panašios formos į sigmoidinę ir netiesiškai transformuoja išėjimo reikšmę į $(-1; 1)$ intervalą. Pagrindinis privalumas, palyginus su sigmoidine aktyvavimo funkcija yra tas, kad įvesties neigiamos reikšmės artės prie -1, tuo tarpu nulinės reikšmės įgis netoli 0 esančią reikšmę. Dažniausiai naudojama klasifikuojant duomenis, kai egzistuoja tik dvi duomenų klasės.

- Išlyginto tiesinio vieneto (angl. *rectified linear unit*, sutr. *ReLU*):

$$f(x) = max(0, x). \tag{1.6}$$

Šiomis dienomis tai plačiausiai naudojama aktyvacijos funkcija. Palyginus su prieš tai aprašytomis funkcijomis, išlyginto tiesinio vieneto funkcija paspartina tinklo mokymosi procesą dėl paprastesnio apskaičiavimo, kadangi jos veikimo principas yra tesiog užnulinti neigiamas reikšmes.



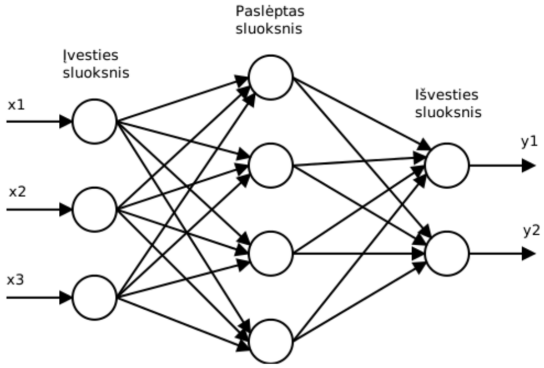
3 pav. Iliustracijoje pavaizduotos dažniausiai naudojamos neuronų aktyvavimo funkcijos.

1.4.2. Neuroninio tinklo sandara

Dirbtinį neuroninį tinklą sudaro tarpusavyje sujungti dirbtiniai neuronai, suskirstyti į sluoksnius. Neuronai, priklausantys vienam sluoksniui, yra įvestis kito sluoksnio neuronams.

DNT sandara (sluoksnių ir neuronų skaičius) priklauso nuo sprendžiamo uždavinio tipo ir sudėtingumo. Kiekvieną neuroninį tinklą sudaro įvesties sluoksnis, perduodantis kintamųjų reikšmes iš išorės, ir išvesties sluoksnis, formuojantis tinklo atsaką. Taip pat naudojami paslėptų neuronų sluoksniai, atliekantys vidinį vaidmenį tinkle. Pagal sluoksnių sujungimo būdą neuroniniai tinklai išskiriami į du tipus [1]:

- Tiesioginio sklidimo (angl. *feedforward*) tinklai, kuriuos sudaro tik vienkryptės jungtys einančios iš įvesties į išvestį per visus paslėptus sluoksnius.
- Grįžtamojo ryšio (angl. *feedback*) arba rekurentiniai tinklai, kuriuose signalas sklinda ir atgalinėmis jungtimis iš vėlesniųjų į ankstesniuosius sluoksnius.



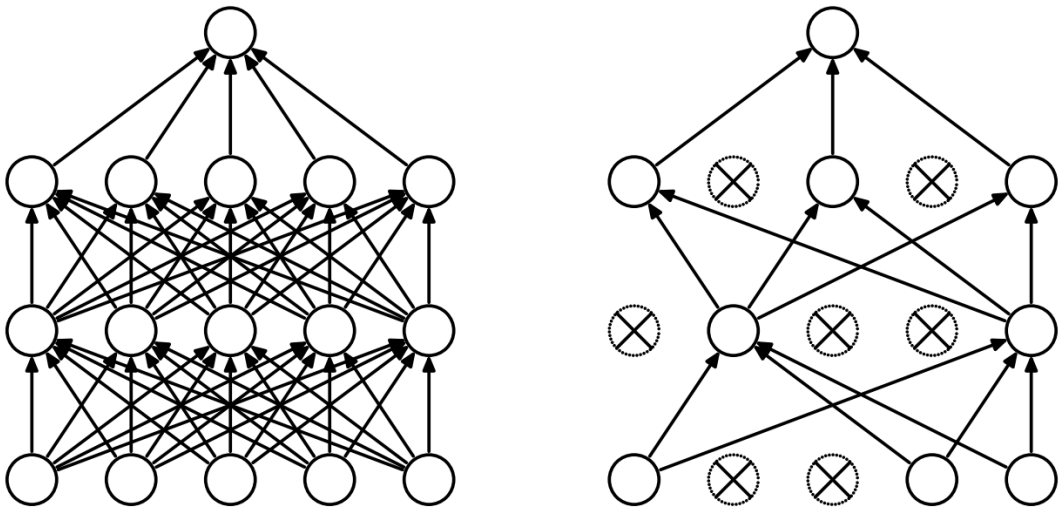
4 pav. Iliustracijose pavaizduota tiesioginio sklidimo dirbtinio neuroninio tinklo architektūra.

Šiuo metu dažniausiai taikomi dirbtiniai neuroniniai tinklai yra tiesioginio sklidimo struktūros. Tokio tipo modelis pavaizduotas 4 pav. Kiekviename paslėptame sluoksnyje atliekama netiesinė ankstesnio sluoksnio įvesties transformacija. Del šios priežasties, kuo daugiau paslėptų sluoksnių turi DNT, tuo sudėtingesnis mokymosi procesas bei didesnis gebėjimas išmokti sudėtingas duomenų struktūras.

Sluoksnių tipų gali būti įvairių. Šiame darbe naudojami sluoksniai dirbtiniams neuroniniams tinklams kurti:

- Pilnai sujungtas sluoksnis (angl. *fully connected layer*) - tai jungiamasis neuroninių tinklų sluoksnis, kuris sujungia visus sluoksnio neuronus su kiekvienu prieš tai esančio sluoksnio neuronu. Todėl šis sluoksnis yra vadinamas pilnai sujungtu sluoksniu. Neurono išėjimas skaičiuojamas sužadinimo signalui pritaikius aktyvacijos funkciją. Dėl galimo didelio jungčių (parametrų) skaičiaus, šis sluoksnis gali išmokti atpažinti sudėtingus duomenų parametrų ryšius, tačiau gali būti sunkiai apmokomas. Sluoksnis pavaizduotas 5 iliustracijos kairėje pusėje.

- Atmetimo sluoksnis (angl. *dropout*) - šis sluoksnis su tam tikra parametru nustatoma tikimybe prilygina nuliui atsitiktinai parinktas įėjimų reikšmes. Tai gali ženkliai padidinti dirbtinių neuroninių tinklų tikslumą bei padeda apsisaugoti nuo vienos iš dažniausiai pasitaikančių problemų - tinklo persimokymo. Atmetimo sluoksnis naudojamas tik tinklo mokymosi metu su apmokymui skirtais duomenimis, naudojant testavimui skirtą duomenų rinkinį sluoksnis yra praleidžiamas. Sluoksnis pavaizduotas 5 iliustracijos dešinėje pusėje.



5 pav. Kairėje iliustracijos pusėje pavaizduotas neuroninis tinklas su pilnai sujungtais sluoksniais. Dešinėje pusėje - tinklas su atmetimo sluoksniais, kurių neuronai yra praleidžiami apmokymo metu. Šaltinis [5].

- *Softmax* sluoksnis - tai išėjimo sluoksnis, kuris transformuoja kiekvieną išėjimo reikšmę intervale nuo 0 iki 1, o visų sluoksnio išėjimo verčių suma yra lygi 1. Prieš tai esantis sluoksnis būtinai turi būti pilnai sujungtas sluoksnis, turintis pilną duomenyse esančių klasių neuronų kiekį. *Softmax* sluoksnis neatlieka duomenų klasifikacijos, o tiesiog transformuoja jau esančias išėjimo vertes į aiškiau suvokiamas reikšmes - tikimybes, kad duomenų įrašas priklauso konkrečiai klasei (jeigu sprendžiamas klasifikacijos tipo uždavinys).

1.4.3. Dirbtinių neuroninių tinklų mokymasis

DNT mokymo procesas apibrėžiamas kaip tinklo struktūros ir jungčių svorių keitimo uždavinys analizuojant didelį kiekį duomenų (kuo didesnį - tuo geriau), kurie jau anksčiau buvo sužymėti teisingomis klasių reikšmėmis bei siekiant, kad tinklas galėtų atlikti jam skirtą užduotį. Duomenų su sužymėtomis teisingų klasių reikšmėmis rinkinys, pateikiamas tinklui apmokymo metu, vadinamas mokymo duomenų rinkiniu.

Neuroninio tinklo mokymas pradedamas su atsitiktiniais neuronų svoriais, kurie yra keičiami kiekviename mokymosi žingsnyje. Tinklas koreguojamas siekiant gauti kiek galima mažesnę paklaidą, t. y. ieškoma tokių svorių, kad skirtumas tarp norimų išėjimo reikšmių ir reikšmių, gautų išmokius neuroninį tinklą, būtų kiek galima mažesnis. Tam yra naudojama paklaidos funkcija (angl. *loss function*). Jeigu naudojamas prižiūrimo mokymosi tipo algoritmas, siekiama, kad gaunamas rezultatas kuo labiau atitiktų norimas išėjimo reikšmes (angl. *desired output*). Tai iš anksto žinomos reikšmės, pavyzdžiui, klasių numeriai, prognozuojamos reikšmės ir pan. Esant skirtumui, tinklo svorių ir neuronų slenksčio reikšmės yra pareguliuojamos ir procesas yra iteratyviai kartojamas bandant kuo įmanoma labiau sumažinti paklaidos funkcijos reikšmę. [26].

Dirbtinis neuroninis tinklas, kuris yra naudojamas klasifikacijos uždaviniams spręsti įvertinamas skaičiuojant tinklo tikslumą (kokią dalį duomenų tinklas suklasifikavo teisingai). Tačiau tai nėra tinkamas būdas vertinant tinklą mokymosi metu. Taip yra todėl, kad neuronų jungčių svoriai mokymosi metu yra keičiami po truputį ir nedidelis reikšmių pokytis gali nepakeisti DNT klasifikavimo tikslumo. Šią problema padeda išspręsti paklaidos funkcija, kurią įtakoja net ir nedidelis tinklo jungčių svorių pokytis.

Mažiausiai paklaidai rasti dažniausiai taikomi gradientiniai optimizavimo metodai. Geriausiai žinomas algoritmas, leidžiantis minimizuoti paklaidos funkciją, vadinamas „klaidos sklidimo atgal“ algoritmu (angl. *backpropagation algorithm*). Šis algoritmas ieško mažiausios neuroninio tinklo paklaidos funkcijos reikšmės atgaline tvarka nuo išvesties link įvesties sluoksnio naudojant gradiento nuolydį (angl. *gradient descent*). Naudojant šį algoritmą svarbu parinkti tinkamą mokymosi greičio parametrą. Per maža reikšmė gali lemti lėtą svorių konvergavimą, tačiau dėl per didelės reikšmės apmokymo procesas gali diverguoti.

Algoritmo veikimą apibūdina du žingsniai:

1. Įvesties neuronų reikšmių „sklidimas pirmyn“ (angl. *forward pass*). Šiame žingsnyje mokymuisi skirti duomenys pereina visus DNT esančius sluoksnius nuo įvesties link išvesties. Atliekama klasifikacija, gaunami išvesties duomenys ir apskaičiuojama paklaida.

2. Gautos paklaidos „sklidimas atgal“ (angl. *backwards pass*) per visą neuroninį tinklą nuo išvesties sluoksnio link įvesties. Šiame žingsnyje naudojami prieš tai gauti išvesties duomenys atnaujinant visų tinkle esančių neuronų jungčių svorius bei slenkstines reikšmes, stengiantis kuo labiau minimizuoti paklaidą.

Klaidos sklidimo atgal algoritmo optimizacija, sudaryta iš šių dviejų žingsnių, yra vadinama iteracija, o visos reikalingos iteracijos, reikalingos apmokyti neuroninį tinklą su pilnu mokymuisi skirtu duomenų rinkiniu, yra vadinama viena epocha.

Apmokius DNT ir norint įvertinti gauto modelio tikslumą naudojamas įvertinimo duomenų rinkinys (angl. *validation dataset*), kuris yra analogiškas tinklo mokymuisi skirtam rinkiniui, susidedančiam iš duomenų su teisingomis klasių reikšmėmis. Įvertinimo metu šio duomenų rinkinio reikšmės modeliu transformuojamos į klasių reikšmes (išvestį) ir palyginamos su tikromis klasių reikšmėmis. Apskaičiavus paklaidos funkciją tikimasi kiek galima mažesnės reikšmės. Modelis

skaitomas paruoštu ir tinkamu naudoti klasifikacijai jeigu paklaidos funkcijos reikšmė yra priimtina ir tinklas pateikė teisingas išvadas įvertinimo metu.

1.4.4. Konvoliuciniai neuroniniai tinklai

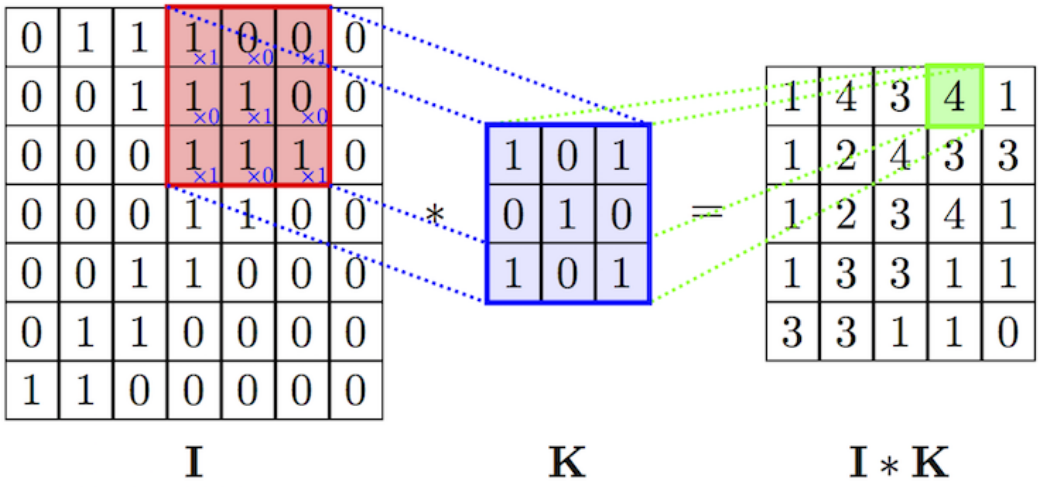
Nors DNT tinka plataus masto uždaviniams spręsti, šis metodas turi trūkumų apdorojant vizualinę informaciją (vaizdus). Vaizdinei medžiagai būdingas aukštas dimensijų skaičius (susidedantis iš $W \times H \times D$ formos tūrio, kurį sudaro W ilgio, H aukščio ir D spalvos kanalų pikseliai) paverčia tinklo apmokymo procesą gan lėtu dėl tinklo architektūros, kurioje kiekvienas neuronas yra sujungtas su visais greta esančio sluoksnio neuronais. Taip pat nėra pritaikomas erdvinis vaizdo įvesties organizavimas, nes kiekviena dviejų greta esančių neuronų sluoksnių pora turi savitą svorį. Tai lemia, kad apmokius tinklą atpažinti tam tikrą objektą vienoje vietoje, tas pats objektas nebus atpažintas kitoje vaizdinės medžiagos vietoje. Šie trūkumai paskatino sukurti konvoliucinius neuroninius tinklus (angl. *convolutional neural network*, sutr. CNN), kurie panaudoja erdvines vaizdinės medžiagos dimensijas (aukštį, plotį ir gylį) ir taip sumažina apmokymui reikalingų parametrų kiekį.

CNN veikimo principas buvo sukurtas atkartojant biologinius procesus vykstančius gyvūnų regos žievėje (angl. *visual cortex*), kurioje per akis gaunama vizuali informacija smegenyse apdorojama ir organizuojama hierarchiniu būdu. Pačios paprasčiausios vizualinės formos (pvz. kampai, kontūrai) yra atpažįstamos pirminėje regos žievės dalyje, o sudėtingesnės formos, tokios kaip įvairūs objektai ar jų grupės - toliau esančioje dalyje.

Konvoliucinis neuroninis tinklas - tai tiesioginio sklidimo prižiūrimo mokymosi tinklas, kuris pirmą kartą buvo aprašytas dar 1980 m. [8]. Nuo tada buvo pasiūlyta nemažas kiekis patobulinimų ir parengti veiksmingi tinklo apmokymo metodai. CNN turi žymiai mažiau apmokymui reikalingų parametrų (jungčių svorių tarp neuronų bei neuronų slenksčio reikšmių), todėl modelio apmokymo procesas trunka trumpiau. Šiomis dienomis CNN tipo tinklai yra plačiai naudojami kompiuterinės regos (angl. *computer vision*) ir kalbos apdorojimo (angl. *natural language processing*) srityse.

Konvoliucinių neuroninių tinklų architektūrą, kaip ir DNT, sudaro įvesties, paslėptieji ir išvesties sluoksniai. Paslėptieji sluoksniai CNN būna trijų tipų - konvoliuciniai, sujungimo (angl. *pooling*) ir pilnai sujungti. Kiekvienas sluoksnis priima įvesties duomenis, su jais atlieka tam tikras operacijas ir perduoda išvesties duomenis į toliau esantį sluoksnį. Dažniausiai CNN architektūra yra sudaryta iš dviejų dalių. Pirmoje dalyje paeiliui einantys konvoliuciniai ir sujungimo sluoksniai atlieka savybių paieškos (angl. *feature extraction*) funkciją. Antra dalis yra sudaryta iš pilnai sujungų sluoksnių, kurie netiesiškai transformuoja surastas vaizdinės medžiagos savybes ir veikia kaip klasifikatorius. Šių sluoksnių pagalba CNN transformuoja įvestos vaizdinės medžiagos pikselių reikšmes į galutinę klasės išvestį.

Konvoliucinį sluoksnį sudaro vienas, arba keli, kvadrato formos filtrai, dar vadinami branduoliais (angl. *kernels*). Šie filtrai dažniausiai būna nedidelio dydžio K, susidedantys iš K^2 svorių. Tinklo apmokymo metu filtrai slinkdami per vaizdinę medžiagą atlieka matricų sandaugos operaciją tarp filtro svorių ir filtruojamos vaizdo srities pikselių reikšmių ir gautas reikšmes susumuoja. Konvoliucijos operacija pavaizduota 6 pav.



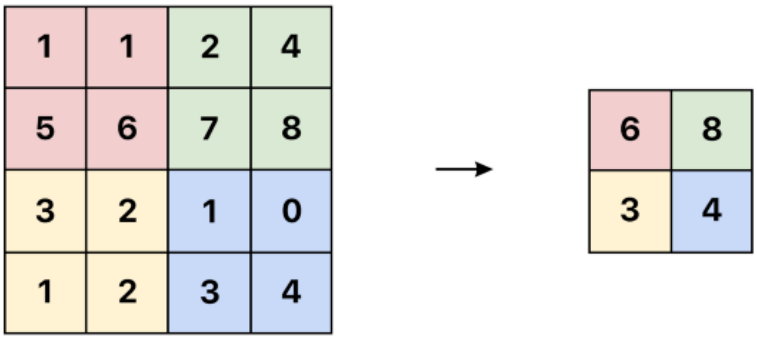
6 pav. Iliustracijoje pavaizduota konvoliucijos operacija, atliekama filtrui slenkant per vaizdo įvesties tūrį I. Skaičiuojama filtro svorių ir įvesties pikselių reikšmių matricų sandaugos suma ($I \times K$). Šaltinis [10].

Konvoliucinis sluoksnis ypatingas tuo, kad atpažįsta įvairias vaizdinės medžiagos ypatybes (angl. *features*). Pirminiai tinklo konvoliuciniai sluoksniai gali atpažinti kontūrus, kraštus, linijas. Toliau esantys sluoksniai atpažįsta sudėtingesnes formas, tokias kaip akių pora (veido atpažinimo uždaviniuose), mašinų ratai, paukščio snapai ir pan.

Šis sluoksnis turi keletą reguliuojamų parametrų:

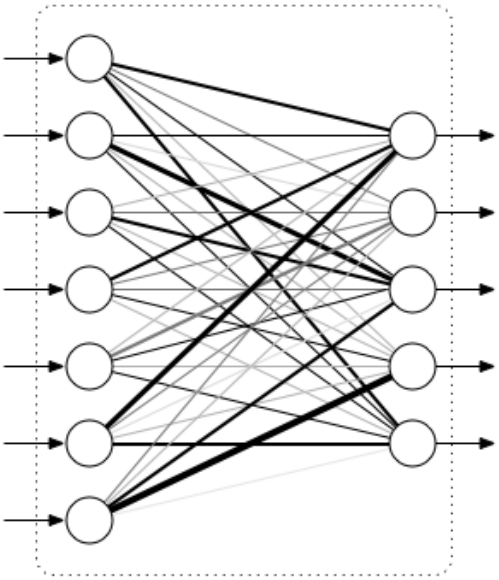
- filtro dydis. Kuo mažesnis filtro dydis, tuo smulkesnes vaizdo detales tinklas sugeba atpažinti. Labiausiai paplitęs filtro dydis yra 3×3.
- filtro slinkimo per įvesties tūrį žingsnis. Pagal nutylėjimą, žingsnio reikšmė būna lygi 1, tai reiškia, kad filtras slinks per kiekvieną vaizdinės medžiagos pikselį. Padidinus žingsnio reikšmę filtras praleis kai kuriuos pikselius ir bus naudojamas su tarpais.
- papildymas nuliais (angl. *zero padding*). Po kiekvienos konvoliucijos operacijos išvestis netenka dalies dimensijų. Šis parametras leidžia padidinti įvesties tūrio kraštus prie kraštų pridedant nulines reikšmes, kad išvesties dydis sutaptų su pirminiu vaizdinės medžiagos dydžiu.

Po konvoliucinio sluoksnio CNN dažniausiai naudojamas sujungimo sluoksnis (angl. *pooling layer*). Šio sluoksnio tikslas yra sumažinti jam pateikto įvesties dydį taip sumažinant visam tinklo apmokymui reikalingų parametrų (neuronų jungčių svorių ir neuronų slenksčio reikšmių) bei operacijų su jais kiekį. Tai ženkliai pagreitina tinklo apmokymo procesą bei leidžia išvengti tinklo prisitaikymo prie apmokymui skirtų duomenų. Panašiai kaip ir konvoliuciniame sluoksnyje naudojamas filtras, kuriuo slenkant per įvesties pikselius atliekamas patenkančių į filtrą elementų kiekio sumažinimas pasitelkiant tam tikrą operaciją. Dažniausiai atliekama maksimumo operacija, kuri tiesiog išrenka didžiausią elemento reikšmę filtruojant įvestį. Taip pat galimos sumavimo ar vidurkinimo operacijos.



7 pav. Iliustracijoje pavaizduota sujungimo sluoksnio veikimo principas. Atliekamas patenkančių į filtrą elementų kiekio sumažinimas naudojant maksimumo operaciją - išrenkant didžiausią elemento reikšmę. Šaltinis [14].

Atlikus duomenų savybių paiešką su konvoliuciniais ir sutelkimo sluoksniais naudojamas paskutinis CNN sluoksnio tipas - pilnai sujungtas sluoksnis (angl. *fully connected layer*), pavaizduotas 8 iliustracijoje. Kaip ir DNT paslėptuose sluoksniuose, kiekvienas pilnai sujungto sluoksnio neuronas yra sujungtas su kiekvienu prieš tai esančio sluoksnio neuronu, o neurono išėjimas skaičiuojamas sužadinimo signalui pritaikius aktyvacijos funkciją. Šis sluoksnis veikia kaip klasifikatorius, kurio pagalba CNN tinklas transformuoja konvoliucijos metu atpažintas įvairias duomenų formas į tiriamų duomenų klasių reikšmes.



8 pav. Iliustracijoje pavaizduotas CNN pilnai sujungtas sluoksnis. Šaltinis [23].

Galutinis CNN tinklo sluoksnis yra išvesties sluoksnis, kuriame siekiama transformuoti iš pilnai sujungto sluoksnio gautas duomenų klasių reikšmes į galutinę tinklo išvestį.

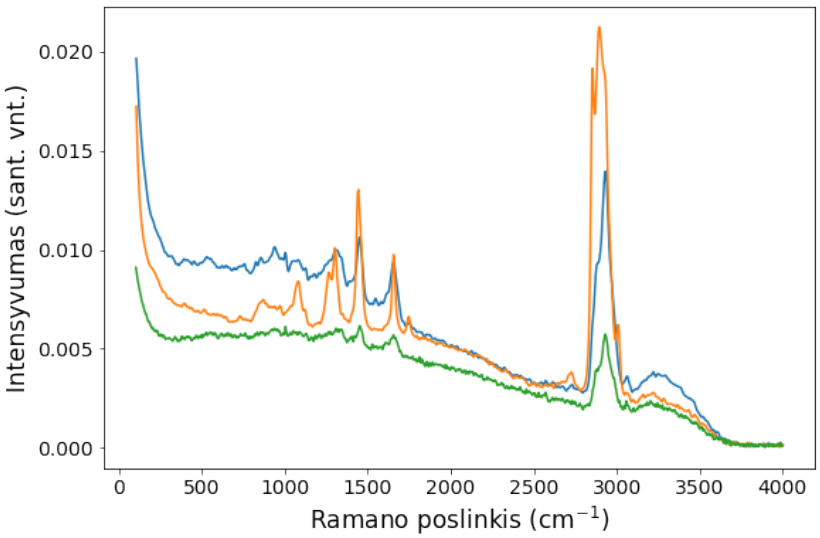
1.5. Ramano spektroskopijos duomenų analizės metodai

Kompiuterio mokymosi metodų taikymas tampa vis populiaresnis tiriant Ramano sklaidos spektroskopija gautus duomenis, ypatingai nustatant cheminę įvairių medžiagų sudėtį, nagrinėjant jų sandarą. Tipiniai duomenų tipo atpažinimo (klasifikacijos) uždaviniai naudojantys Ramano

sklaidos duomenis yra sudaryti iš tokių žingsnių, kaip duomenų paruošimas prieš analizę, dimensijų mažinimas, klasifikavimo modelio kūrimas (duomenų apmokymas) ir galiausiai modelio taikymas. Pirminiame duomenų paruošimo etape dažniausiai atliekama bazinės linijos korekcija, taip pat gali būti pritaikomos kosminės spinduliuotės (triukšmo) pašalinimo ir duomenų glodinimo operacijos. Vėliau, priklausomai nuo pasirinkto klasifikavimo algoritmo, gali būti taikomas duomenų dimensijų mažinimo metodas naudojant pagrindinių komponenčių analizę ar, pastaruoju metu vis populiarėjantį, automatinio šifravimo (angl. *autoencoder*) metodą paremtą neuroniniais tinklais. Paskutiniame žingsnyje su paruoštais duomenimis yra sukuriamas klasifikavimo modelis (apmokomas), kuris vėliau naudojamas nustatant duomenų klasėms. Taikant bet kurį klasifikavimo algoritmą, išskyrus neuroninius tinklus, reikalingas vienoks ar kitoks duomenų paruošimas prieš naudojimą, nes galutinis duomenų klasifikavimo rezultatas yra stipriai paveiktas Ramano sklaidos spektroskopijoje naudojamų optinių metodų paklaidos [9].

1.5.1. Bazinės linijos korekcija

Dažnai dėl įvairių eksperimentinių priežasčių bazinė Ramano sklaidos spektrogramos linija nėra horizontali, todėl pirmiausia spektrą reikia pakoreguoti eliminuojant bazinės linijos iškraipymus, gaunamus dėl optinių metodų paklaidos [25]. Pagrindinė šios paklaidos priežastis yra fluorescencijos efektas, kuris priklausomai nuo tiriamosios medžiagos optinių savybių bei spektrofotometro naudojamos lazerinės spinduliuotės ir skyros, skirtingoms medžiagoms gali skirtis iki kelių kartų. Tai stipriai veikia tiriamo spektro analizę, todėl būtina įvertinti ir jei reikalinga pašalinti šio efekto daromą įtaką spektrui. 9 pav. matomi trijų skirtingų mėsos rūšių (jautienos, vištienos ir kiaulienos) Ramano spektrai su skirtingais bazinės linijos iškraipymais.



9 pav. Iliustracijoje pavaizduoti mėsos sklaidos spektrai su skirtingomis bazinėmis linijomis. Mėlyna spalva pavaizduotas kiaulienos, oranžine - vištienos, žalia - jautienos Ramano sklaidos spektras.

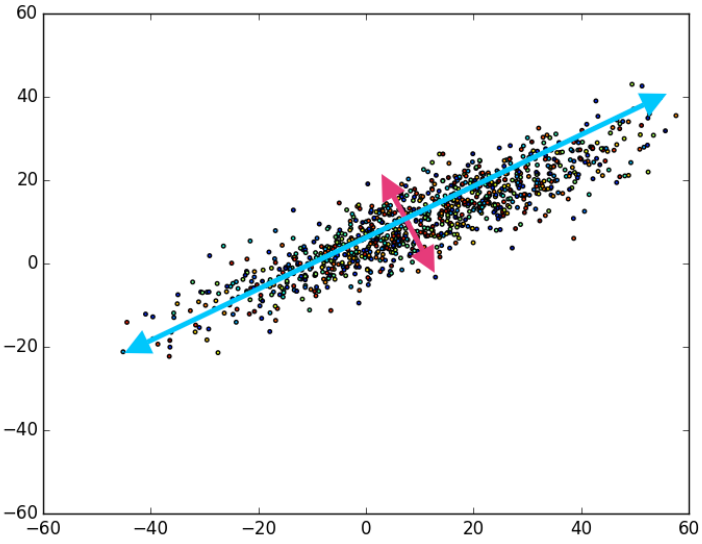
Bazinės linijos iškraipymų eliminavimui taikomi įvairūs algoritmai, kurie aproksimuoja spektrinę liniją teoriniais kontūrais, parenkant parametrus (plotį, aukštį, smailės padėtį), kad gautų geriausią sutapimą su eksperimentine kreive. Korekcijai atlikti yra naudojami tiek sudėtingesni metodai, tokie kaip neuroninių tinklų [22] ar įvairaus laipsnio polynomų modeliavimo [24], tiek paprastesni, tokie kaip bazinių taškų ar mažiausių kvadratų (angl. *least squares estimation*) [2]. Taip pat dažnai taikomi specifiniai metodai, tokie kaip mažiausių kvadratų aukšto laipsnio daugianarės

kreivės priderinimas (angl. *fitting*) [18], „riedančio kamuolio“ (angl. *rolling ball*) metodas, kurio principas yra kreive riedantis kintamo diametro kamuolys (ar apskritimas), kurio centro pozicija yra traktuojama kaip bazinė linija, kurią naudojant reikia atlikti korekciją. Panašus yra „guminės apyrankės” (angl. *rubber band*) metodas, kai yra simuliuojamas apyrankės judėjimas kreive ir skaičiuojamas jos ilgis, kuris transformuojamas į bazinį korekcijos lygmenį [9]. Taip pat naudojamas adaptyvus mažiausių kvadratų (angl. *adaptive iteratively reweighted penalized least squares*, sutr. airPLS) algoritmas, kuris iteratyviai adaptuoja svorius, kurie apibrėžia santykį tarp bazinio lygmens ir signalo [17].

Vienas iš populiariausių algoritmų bazinės spektro linijos ištiesinimui yra asimetrinis mažiausių kvadratų glodinimas (angl. *Asymmetric Least Squares*, sutr. ALS) [9]. Atliekant spektro korekciją šiuo metodu visų pirma siekiama nustatyti pradinį bazinį lygmenį taikant *Whittaker* glodinimo algoritmą, o vėliau yra naudojamas asimetrinis mažiausių kvadratų derinimas bandant iteratyviai sumažinti aukščiau bazinės linijos esančių taškų svorį ir parinkti tokius parametrus, kad būtų gautas geriausias sutapimas su eksperimentine kreive [4].

1.5.2. Pagrindinių komponenčių analizė

Atliekant duomenų analizę pasitaiko atvejų, kai tiriamąjį objektą geriausiai apibūdina ne kažkuri konkreti jo savybė ar jų grupė, bet įvestas abstraktus naujas kintamasis. Vienas iš šių abstrakčių kintamųjų išskyrimo metodas yra pagrindinių komponenčių analizė. Tuo pačiu sprendžiamas ir kitas uždavinys - duomenų matmenų (savybių) kiekio mažinimas. Objektui turinčiam didelį kiekį parametrų, tarp jų dažnai yra vos keletas, kurie nusako didžiąją dalį svarbiausios informacijos apie objektą. Išskiriant pagrindines komponentes, iš duomenų išskiriami informatyviausi kintamieji.



10 pav. Iliustracijoje mėlyna spalva pavaizduota pirmoji pagrindinė komponentė (PC1), raudona spalva - antroji (PC2). Šaltinis [21].

Pagrindinių komponenčių analizė (angl. *principal component analysis*, sutr. PCA) yra vienas iš klasikinių statistikos metodų, plačiai naudojamas duomenų analizėje daugiamačių duomenų matmenų skaičiui mažinti pasitelkus tiesišką duomenų transformaciją. Taip pat šis metodas plačiai taikomas duomenų vizualizavimui, esminių savybių suradimui bei duomenų suspaudimui atsisa-

kant nereikšmingų parametrų [13]. Veikimo principas siekiant sumažinti duomenų dimensijų kiekį yra pagrįstas dalies po tiesinės transformacijos gautų naujų komponenčių, kurių dispersijos yra mažiausios, atsisakymu [19].

Didžiausią dispersiją turinti kryptis vadinama pirmąja pagrindine komponente (sutr. PC1). Ji eina per duomenų centrinį tašką, kuris nutolęs mažiausiu atstumu nuo visų duomenų taškų erdvėje. Iš to paties taško išvedama antroji pagrindinė komponentė (sutr. PC2), statmena pirmajai. Abi šios tiesės vadinamos pagrindinėmis komponentėmis ir pavaizduotos 9 pav.

Algoritmo veikimo metu iš duomenų aibės sudaroma kovariacinė matrica, iš kurios apskaičiuojamos tikrinės reikšmės (angl. *eigenvalue*) ir tikriniai vektoriai (angl. *eigenvector*). Tikrinių vektorių skaičius yra lygus duomenų matricą sudarančių vektorių komponenčių skaičiui. Iš surūšiuotų tikrinių vektorių gaunama pagrindinių komponenčių matrica. Pagrindinėms komponentėms nustatyti užtenka rasti d didžiausių šios matricos tikrinių reikšmių ir jas atitinkančių tikrinių vektorių. Duomenų vizualizacijai dažniausiai naudojamas pagrindinių komponenčių skaičius d būna lygus 2 arba 3. Taip pat dažnai pasirenkamas toks skaičius, kad pagrindinių komponenčių procentinis dydis nuo visos dispersijos viršytų tam tikrą pasirinktą lygį, pavyzdžiui, 90% [26].

1.5.3. Susijusių darbų apžvalga

Įvairūs kompiuterių mokymosi metodai, ypač klasifikacijos, Ramano sklaidos spektroskopijos duomenims taikomi jau daugiau nei du dešimtmečius. Vienas dažniausiai taikomų ir geriausius rezultatus pasiekiančių metodų duomenų klasifikacijai yra atraminių vektorių mašinų metodas (sutr. SVM). Šis metodas pasiekia gan aukštą klasifikacijos tikslumą apmokius modelį su nedidelės apimties duomenimis [20]. Dažnai prieš taikant šį algoritmą atliekamas pagrindinių komponenčių analizės metodas ir sumažinamas dimensijų skaičius. Nors praktikoje yra nemažai sėkmingų SVM taikymo pavyzdžių, tačiau šis metodas nėra tinkamas didelės apimties matavimų (turinčių didelį klasių kiekį) atskyrimo uždaviniams spręsti. Kaip galima alternatyva SVM metodui didelės apimties uždaviniams spręsti yra taikomas atsitiktinių medžių (angl. *random forests*, sutr. RF) klasifikacijos algoritmas. Tai yra kompleksinis kompiuterių mokymosi metodas, kuris pagrįstas dideliu sprendimų medžių kiekiu, kurie atkartoja duomenų savybes bei yra pritaikyti efektyviai išvengti persimokymo (angl. *overfitting*) [12]. Nors atsitiktinių medžių metodo taikymas ir populiarėja klasifikuojant Ramano sklaidos spektroskopijos duomenis, rezultatai, palyginus su SVM algoritmu, išlieka prastesni [9].

Taip pat šio tipo duomenims klasifikuoti yra plačiai naudojami įvairios architektūros neuroniniais tinklais pagrįsti algoritmai. Įrodyta, kad Ramano sklaidos spektroskopijos duomenis pakankamai gerai klasifikuoja paprastieji (angl. *feedforward*) neuroniniai tinklai [2]. Tačiau pagrindinis šio metodo trūkumas yra tai, kad jame nėra tinkamos talpos su kuria būtų galima įvertinti subtilias duomenų detales ir spręsti uždavinius sudarytus iš didelio klasių, artimų viena kitai, kiekio [3].

Pastaruoju metu sparčiai populiarėja Ramano sklaidos spektroskopijos būdu gautų duomenų klasifikavimas naudojant konvoliucinius neuroninius tinklus. Šio metodo talpa yra pakankama, kad modelis įvertintų netgi mažiausiais duomenų detales ir subtilybes klasifikuojant duomenis. Konvoliuciniai tinklai sėkmingai pritaikyti klasifikuojant Ramano spektrus iš laisvai platinamos RRUFF mineralų spektrų duomenų bazės [7]. Šio tipo tinklai parodė aukštesnį tikslumą nei tradiciniai klasifikavimo algoritmai, tokie kaip SVM. Be to, priešingai įprastoms praktikoms, konvoliuciniai tinklai geriau veikė apdorojant spektrus be koreguotos bazinės linijos. Tai pasiekti leidžia metode esantis konvoliucinis sluoksnis, kuris įsimena ir geba identifikuoti duomenų savybes nepriklausomai nuo jų absoliučių reikšmių [9].

2. Eksperimentinė dalis

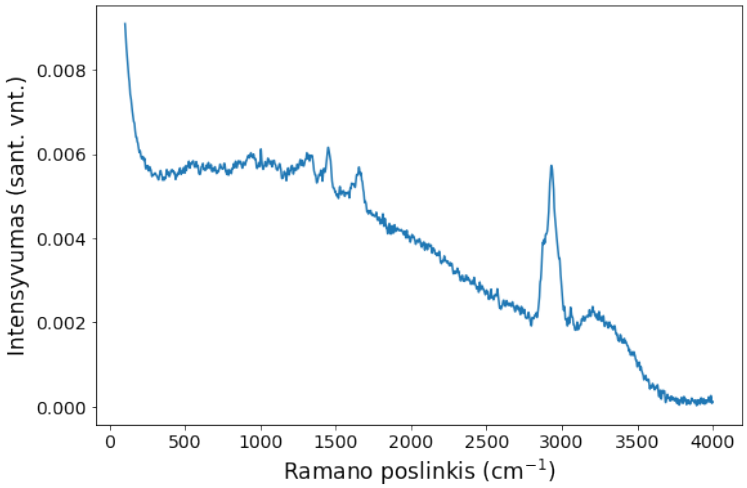
2.1. Tyrimo metodika

Eksperimentinė darbo dalis buvo atliekama naudojant *Python* (3.6 versija) programavimo kalbą. Duomenų analizei bei kompiuterio mokymosi modelių kūrimui buvo naudojami *Scikit-learn*, *Keras*, *TensorFlow*, *NumPy* paketai. Programinis kodas buvo rašomas *Jupyter Notebook* aplinkoje. Ši aplinka leidžia lengvai susieti kodą su analizės aprašymu bei rezultatais. Tyrime kuriami kompiuterių mokymosi modeliai buvo apmokomi ir testuojami naudojant šią įrangą:

- Stacionarus kompiuteris su Intel i5-2500K procesoriumi, 8 GB operatyviosios atminties, Nvidia GeForce GTX 1080 Ti grafiniu procesoriumi, Ubuntu 18.04 operacine sistema.
- Nešiojamasis kompiuteris Lenovo Thinkpad T460s su Intel i5-6200U procesoriumi, 8 GB operatyviosios atminties, Ubuntu 18.04 operacine sistema.

2.2. Tyrimo duomenys ir paruošimas

Eksperimentui atlikti buvo gauti įvairių rūšių mėsos Ramano spektrai: kiaulienos, jautienos, vištienos ir kitų rūšių. Mėsos gaminiai buvo įsigyti vietiniuose prekybos centruose, turgavietėse arba tiesiogiai iš ūkininkų. Tyrimo duomenų rinkiniui gauti buvo naudojami įvairūs tų pačių mėsos tipų (gyvulių rūšių) mėginiai, paimti iš skirtingų gaminių (pvz. kiaulienos šoninė, nugarinė, lašiniai ir t.t.). Didžioji dalis matavimų buvo atlikta su šviežiais mėsos mėginiais, keli mėginiai buvo paimti iš šaldytos arba atšildytos mėsos. Iš kiekvieno mėginio gauta nuo 5 iki 100 Ramano spektrų, tarpusavyje pasižyminčių tam tikromis variacijomis, įtakotomis matavimo paklaidų, triukšmo, matavimo metu minimaliai besikeičiančių medžiagos savybių (temperatūra, drėgmė). Visi mėginių spektrai gauti *Bruker MultiRAM* Ramano spektrometru, sužadinti naudojant 785 nm lazerinę spinduliuotę, 500 mW galią ir 8 cm^{-1} arba 4 cm^{-1} skyrą. Vienas iš tyrime naudojamų mėsos (jautienos) spektrų pavaizduotas 11 pav.



11 pav. Iliustracijoje pavaizduotas jautienos Ramano sklaidos spektras.

Tyrimui atlikti gautos 3771 bylos, kuriose talpinami skirtingi ištirtų medžiagų spektrai (detaliau žr. 1 lentelėje). Kiekvienoje byloje pateikti suvidurkinti 200 to paties mėginio matavimų duomenys. Atliktuose matavimuose Ramano spektro poslinkis svyruoja nuo 4000 cm^{-1} iki 200 cm^{-1}

ir duomenų byloje yra išreikštas per 951 atributų (savybių). Tyrimo duomenų analizei paruošti matavimų duomenys buvo 80/20 santykiu padalinti į apmokymo ir testavimo duomenų rinkinius, išlaikant šias proporcijas kiekvienoje tirtų medžiagų spektrų grupėje. Apmokymo ir testavimo aibių spektrų kiekiai kiekvienai mėsos rūšiai matomi 1 lentelėje.

Mėsos rūšis	Spektrų kiekis	Apmokymo aibė	Testavimo aibė
Antiena	363	289	74
Aviena	260	207	53
Ėriena	440	352	88
Jautiena	168	134	34
Kalakutiena	616	492	124
Kiauliena	881	703	178
Veršiena	615	491	124
Vištiena	428	340	88
Iš viso	3771	3008	763

1 lentelė. Tyrimo duomenų (įvairių rūšių mėsų) Ramano spektrų kiekiai, apmokymo ir testavimo aibės.

2.3. Duomenų klasifikavimas naudojant atraminių vektorių mašinų metodą

Vienas dažniausiai taikomų kompiuterių mokymosi metodų Ramano spektroskopijos duomenų klasifikavimui atlikti yra atraminių vektorių mašinų metodas. Šis metodas plačiau aprašytas 1.3 poskyryje.

2.3.1. Bazinės linijos korekcija

Kaip aptarta 1.5.1 skyrelyje, tiriant Ramano sklaidos spektrus būtina pašalinti bazinės linijos iškraipymus, gaunamus dėl matavimo metu atsirandančios optinių metodų paklaidos. Deja, pilnai automatinis būdas bazinės linijos iškraipymams eliminuoti kol kas nėra sukurtas. Visiems teorinėje dalyje aptartiems algoritmams reikalingas žmogaus įsikišimas nustatant bei reguliuojant algoritmų parametrus siekiant, kad korekcija kuo labiau atitiktų norimą rezultatą.

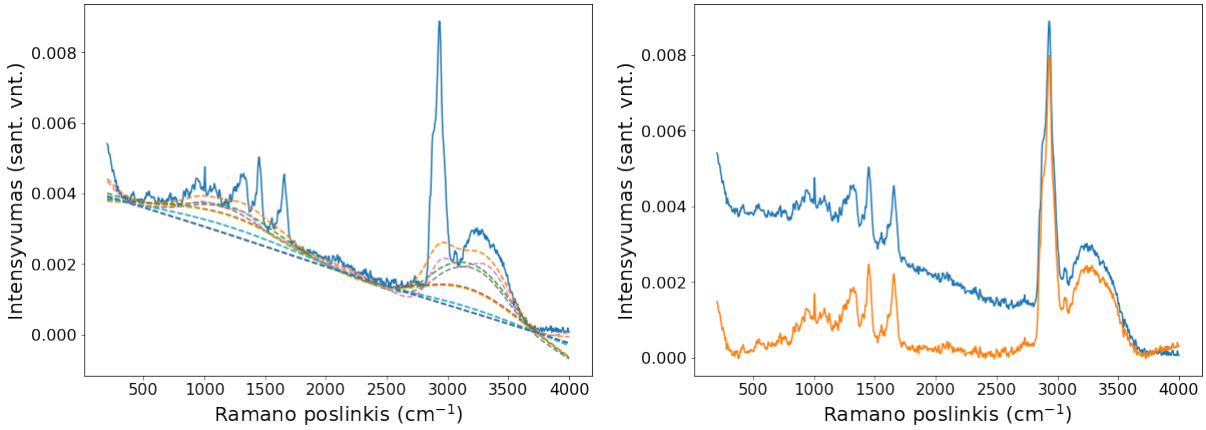
Tiriamų spektrų apdorojimui buvo panaudotas vienas iš populiariausių bazinės spektro linijos ištiesinimui skirtų algoritmų - asimetrinis mažiausių kvadratų metodas (angl. *Asymmetric Least Squares*). Atliekant bandymus šis algoritmas greitai ir efektyviai eliminavo bazinę liniją nepažeisdamas pačio spektro svyravimų. Šis algoritmas turi du parametrus:

1. p - parametras skirtas reguliuoti bazinės linijos aproksimacijos asimetriškumą. Didinant parametro p reikšmę bazinė linija darosi panašesnė į patį spektrą.

2. λ - kontūro linijos glodumo parametras. Didinant šią reikšmę bazinė linija darosi vis labiau panaši į tiesę.

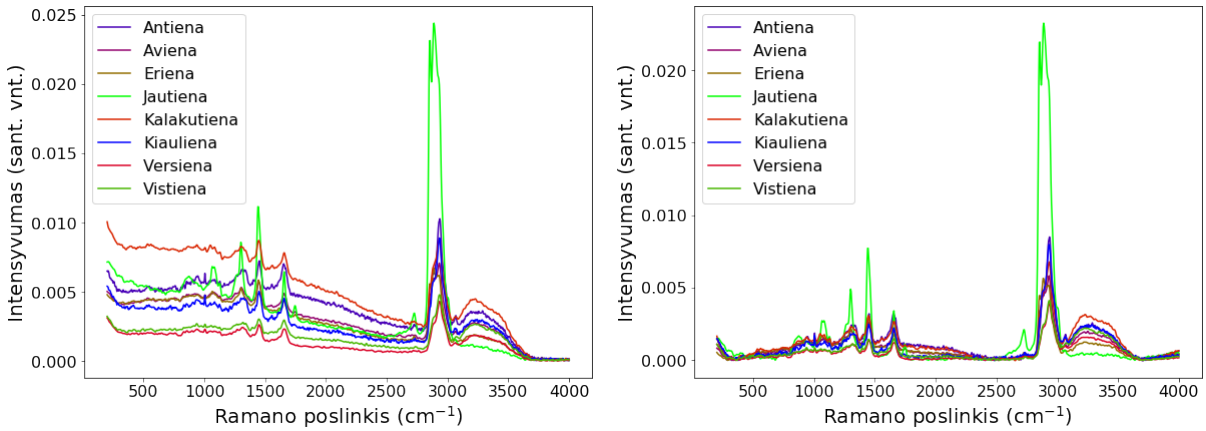
Ištirta, kad teigiamoms spektrų reikšmėms dažniausiai tinka p parametro reikšmės iš $[0.001, 0.1]$ intervalo ir λ reikšmės iš $[10^2, 10^9]$ intervalo [4]. Tačiau parinkti optimalias parametrų reikšmes

yra netrivialu ir skirtingiems spektrams jos gali skirtis, be to bazinės linijos korekcijos rezultatai paprastai vertinami vizualiai, taikant ekspertines dalykinės srities žinias. Eksperimento metu buvo bandomos įvairios algoritmo parametrų reikšmės patenkančios į pateiktus intervalus. 12 pav. pavaizduotas atsitiktinai atrinktas kiaulienos Ramano sklaidos spektras su punktyrinėmis linijomis pažymėtomis asimetrinio mažiausių kvadratų metodu gautomis kontūro linijomis, taikant įvairias parametrų kombinacijas. Vizualiai įvertinus algoritmo parametrų daromą poveikį, buvo pasirinkta p parametro reikšmė lygi 10^{-4}, λ - 10^6. Su šiais parametrais gautas geriausias sutapimas su eksperimentine spektro kreive. Bazinė linija eliminuojama iš tiriamo spektro reikšmių atėmus algoritmu gautos kontūro linijos vertes. Spektro linijos korekcijos su pasirinktais tinkamiausiais parametrais rezultatas matomas 12 pav. dešinėje pusėje.



12 pav. Iliustracijose pavaizduota kiaulienos kumpio Ramano sklaidos spektro korekcija eliminuojant bazinės linijos iškraipymus. Mėlyna spalva iliustracijose pažymėtas originalus kiaulienos spektras. Kairėje pusėje punktyrinėmis linijomis atvaizduotos bazinės kontūrų linijos, gautos panaudojus asimetrinį mažiausių kvadratų metodą taikant įvairias parametrų kombinacijas. Dešinėje pusėje pavaizduotas apdorotas kiaulienos spektras pašalinus bazinės linijos iškraipymus naudojant p - 10^{-4}, λ - 10^6 parametrų reikšmes.

Tyrimo duomenų rinkinį iš viso sudaro 8 skirtingų klasių (mėsos rūšių) spektrai. 13 pav. pavaizduoti iš kiekvienos tiriamos klasės po vieną atsitiktinai atrinkti originalūs spektrai ir šių spektrų korekcijos eliminavus bazinės linijos iškraipymus.



13 pav. Kairėje iliustracijoje pavaizduoti atsitiktinai iš kiekvieno tiriamo mėsos tipo po vieną atrinkti Ramano spektrai. Dešinėje pusėje matomos šių spektrų bazinės linijos korekcijos.

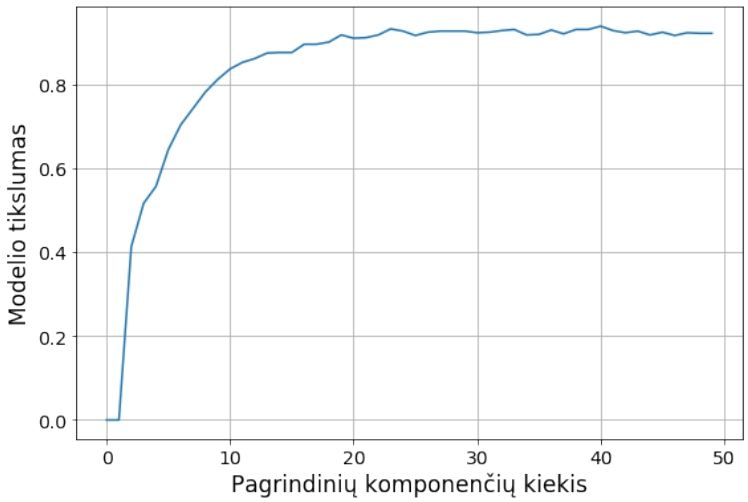
2.3.2. Modelio kūrimas

Kuriant SVM klasifikatorių visų pirma buvo eliminuoti spektrų bazinių linijų iškraipymai panaudojant 2.3.1 skyrelyje aprašytas parametrų reikšmes p - 10^{-4}, λ - 10^{6}. Tada patikrinta koks būtų gautas klasifikatoriaus tikslumas naudojant apmokymui skirtus duomenis ir parinkus standartinius modelio parametrų nustatymus. Įsitikinta, kad gautas modelio tikslumas siekia vos 23.33%. Tikslumui matuoti buvo naudojama testavimui skirtas duomenų rinkinys (20% visų turimų duomenų, kurie nebuvo naudojami apmokant modelį), kurį modelis panaudojo apskaičiuojant teisingai klasifikuotų duomenų santykį su visais testavimui skirtais duomenimis. Pagrindiniai veiksniai lemiantys tokį žemą klasifikacijos tikslumą:

1. Netinkamai parinkti klasifikacijos modelio parametrai.
2. Aukštas duomenų atributų (dimensijų) skaičius, kuris siekia net 951.
3. Nevienodas tyrimo duomenų kiekio pasiskirstymas tarp klasių. Kaip matoma iš 1 lentelės, jautienos Ramano spektrų tėra vos 168, kai tuo tarpu kiaulienos - 881.
4. Didelis duomenų klasių kiekis (tiriamos aštuonios skirtingos mėsos rūšys).

Siekiant optimizuoti klasifikatoriaus veikimą buvo panaudotas *Scikit-learn* bibliotekos *GridSearchCV* modulis, kuris sugeneruoja parametrų aibę ir apskaičiuoja modelio tikslumą naudojant kiekvieną parametrų aibės rinkinį. Taip galima tiksliau nustatyti su kokiomis parametrų reikšmėmis modelis pasiekia aukščiausią klasifikacijos lygį. Šis metodas parodė, kad tyrimo duomenims labiausiai tinka tiesinė (angl. *linear*) branduolio funkcija su parametro C reikšme, lygia 10. Su šiais parametrais modelio tikslumas siekė 35.78%.

Aukšto duomenų dimensijų kiekio problemai spręsti buvo pasitelkta pagrindinių komponenčių analizė (angl. *Principal Component Analysis*, sutr. PCA). Šis metodas padeda sumažinti duomenų matmenų skaičių atliekant tiesinę transformaciją ir atsisakant dalies po transformacijos gautų naujų komponenčių, kurių dispersijos yra mažiausios [26]. Naudojant PCA metodą taip pat buvo vykdoma *Whitening* duomenų paruošimo procedūra, kuri tiesiškai transformuoja duomenis ir užtikrina, kad jie mažiau koreliuotų tarpusavyje ir turėtų tą pačią dispersiją. Pritaikius šią procedūrą greta PCA metodo pastebėtas klasifikatoriaus tikslumo padidėjimas.



14 pav. Iliustracijoje pavaizduota klasifikatoriaus modelio tikslumo, gauto pritaikius PCA metodą, priklausomybė nuo pagrindinių komponenčių kiekio.

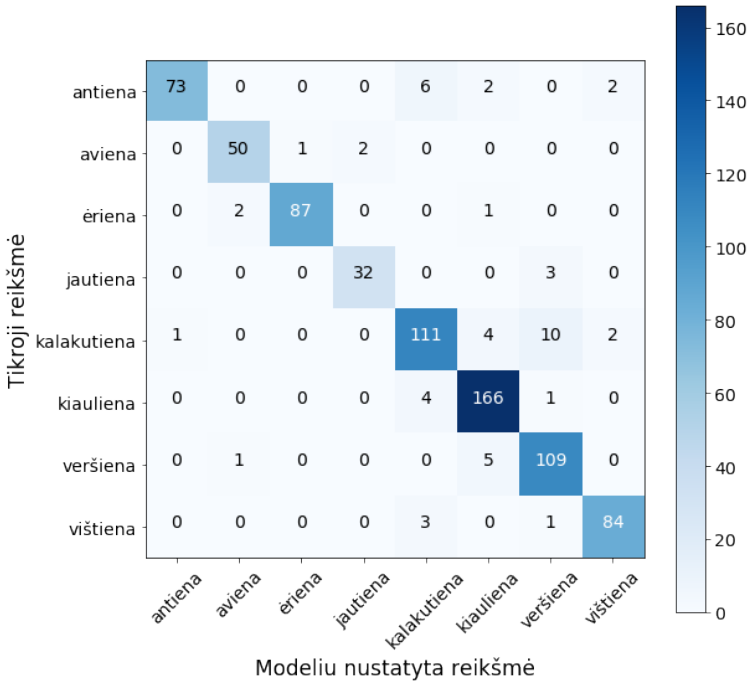
Siekiant išsiaiškinti koks pagrindinių komponenčių (kintamųjų, kurie nusako svarbiausią informaciją apie duomenis) kiekis geriausiai apibūdina duomenis buvo apskaičiuotas klasifikatoriaus tikslumas iteratyviai naudojant pagrindinių komponenčių kiekį nuo 2 iki 50. Šių skaičiavimų rezultatas matomas 14 pav. Iliustracijoje matoma, kad modelio tikslumas auga pagrindinių komponenčių kiekį didinant iki 24, vėliau įsisotina ir išlaiko panašų tikslumą. Didžiausias modelio tikslumas gautas panaudojant 24 pagrindines komponentes ir siekė 92.45%.

Eksperimento metu buvo pastebėta, kad kai kuriais atvejais atraminių vektorių mašinos metodas pasiekia didesnį tikslumą, kai duomenys yra normalizuoti. Todėl apmokymo ir testavimo duomenų rinkiniams buvo pritaikytas klasių vardų normalizavimas pasitelkus *Scikit-learn* bibliotekos *LabelEncoder* modulį. Šis modulis transformuoja neskaitinius duomenų klasių vardus į skaitines reikšmes. Taip pat duomenų rinkiniams buvo pritaikytas standartinis normalizavimas (angl. *standard scaling*) - tai tiesinė spektro reikšmių transformacija, po kurios reikšmių vidurkis tampa lygus 0, o standartinis nuokrypis lygus 1. Tam panaudotas tos pačios bibliotekos *StandardScaler* modulis. Duomenų normalizavimas leido nežymiai padidinti klasifikatoriaus tikslumą iki 93.79%.

Siekiant suvienodinti tyrimo duomenų kiekio pasiskirstymą tarp klasių buvo išbandytas papildomas modelio parametras *Class Weight*, kuris apmokant modelį klasėms priskiria skirtingus svorius, priklausomai nuo jų pasikartojimo apmokymo aibės duomenyse. Pritaikius šį parametrą modelio tikslumas nukrito iki 92.13%, todėl šio parametro naudojimo buvo atsisakyta.

2.3.3. Modelio įvertinimas

Klasifikavimo modelio tikslumas ir korektiškumas yra įvertinamas klasifikuotų duomenų reikšmes lyginant su tikrosiomis (kurios buvo atidėtos testavimo aibėje). Vienas iš pagrindinių būdų klasifikavimo tikslumui įvertinti yra painiavos matrica (angl. *confusion matrix*), kuri parodo teisingų bei klaidingų modelio spėjimų skaičių bei klaidų pasiskirstymą kiekvienai klasei. Šis metodas gali būti naudojamas tiek binarinei, tiek daugiaklasei (angl. *multiclass*) klasifikacijai ištirti. Apskaičiuota modelio painiavos matrica pavaizduota 15 pav.



15 pav. Iliustracijose pavaizduota modelio painiavos matrica.

Iš grafiko matome, kad modelis su panašiu tikslumu identifikavo kiekvieną iš tirtų klasių. Dažniausiai daromos klaidos pasitaikydavo klasifikuojant kalakutieną (modeliui sumaišius su antiena) ir veršieną (sumaišius su kalakutiena).

Kiti dažnai naudojami matai klasifikatoriaus kokybei nustatyti [16]:

- *Accuracy*: šis matas parodo klasifikacijos modelio tikslumą apskaičiuojant teisingai klasifikuotų duomenų santykį su visais testavimui skirtais duomenimis. Šis tikslumo įvertinimo būdas nėra tinkamas naudoti kai duomenys yra nesubalansuoti. Pavyzdžiui, jei vieną iš tiriamų klasių sudaro 90% duomenų, o kitą klasę likusieji 10%, tai modelis pasieks 90% tikslumą visada klasifikuodamas daugumos klasę.

- *Precision*: nusako kokia dalis prognozuotų spėjimų buvo iš tikrųjų teisingi. *Precision* yra klasifikatoriaus gebėjimas nepažymėti teigiamo pavyzdžio, kuris iš tikrųjų yra neigiamas.

- *Recall*: nusako kokia dalis teigiamų spėjimų (angl. *true positive*) buvo prognozuota teisingai padalinus šį skaičių iš bendro elementų, esančių toje klasėje, skaičiaus. *Recall* parodo klasifikatoriaus gebėjimą rasti visus teigiamus klasės atvejus.

- *F1 score*: dar vienas dydis, skirtas tikrinti sukurto modelio klasifikavimo kokybei, apskaičiuotas panaudojant *Precision* ir *Recall*, kuris parodo kokia dalis prognozuotų teigiamų spėjimų buvo teisingi. Parametro reikšmės kinta nuo 0 iki 1, kur 1 yra aukščiausia vertė reiškianti puikų modelio tikslumą, 0 - žemiausia vertė. Kadangi tiriamu atveju skirtingų klasių duomenų skaičius yra nesubalansuotas, šis parametras gana gerai atspindi sukurto modelio tikslumą.

- Taip pat plačiai naudojamos metrikos yra *ROC* ir *AUC* kreivės, tačiau jos naudojamos binariniam klasifikacijos modeliui ir šiuo atveju netinka.

Siekiant apskaičiuoti dažniausiai naudojamas metrikas klasifikatoriaus kokybei nustatyti buvo panaudotas *Scikit-learn* bibliotekos *Classification Report* metodas, kurio rezultatai matomi 2 lentelėje. Šis metodas apskaičiavo tikslumo metrikas kiekvienai klasei atskirai bei modelio vidurkį visoms klasėms. Lentelėje pateikti skaičiavimai sutampa su painiavos matricoje pateiktais rezultatais. Matome, kad modelis gana tiksliai identifikavo kiekvieną iš tirtų klasių. Itin geras rezultatas gautas klasifikuojant ėrieną, o prasčiausias rezultatas pasiektas identifikuojant kalakutienos mėsą.

Mokslo tiriamojo darbo metu atraminių vektorių mašinų metodu tirtam duomenų rinkiniui klasifikuoti buvo pasiektas 81.45% tikslumas. Naudojamą duomenų aibę tuo metu tesudarė 596 Ramano spektrai gauti iš septynių skirtingų mėsos rūšių. Rinkinys taip pat buvo labai netolygiai pasiskirstęs pagal duomenų kiekį kiekvienoje klasėje. Kalakutienos spektrų kiekį sudarė 24, kai tuo tarpu antienos - 160 spektrai. Ruošiant duomenis klasifikacijai taip pat buvo naudojama bazinės linijos korekcija, atlikta su kiek kitais parametrais: p - 10^{-3}, λ - 10^{4}. Duomenų matmenų skaičius buvo mažinamas naudojant pagrindinių komponenčių analizę. Tačiau pasirinktas pagrindinių komponenčių (duomenų dimensijų) skaičius buvo 10. Šiam metodui papildomai pritaikyta *Whitening* procedūra, kuri užtikrina, kad duomenys mažiau koreliuotų tarpusavyje. Siekiant suvienodinti tyrimo duomenų kiekio pasiskirstymą tarp klasių buvo pritaikytas papildomas modelio parametras *Class Weight*, kuris leido pasiekti geresnį modelio tikslumą. Duomenų bei duomenų klasių vardų normalizavimas nebuvo atliktas.

2 lentelė. Apskaičiuotos dažniausiai naudojamos metrikos klasifikatoriaus kokybei nustatyti.

Klasė	Precision	Recall	F1-score	Spėjimų skaičius
Antiena	0.96	0.86	0.90	83
Aviena	0.92	0.94	0.93	52
Ėriena	1.00	0.97	0.98	91
Jautiena	0.91	0.89	0.90	35
Kalakutiena	0.90	0.89	0.89	125
Kiauliena	0.94	0.97	0.95	174
Veršiena	0.88	0.96	0.92	114
Vištiena	0.97	0.96	0.96	89
Vidurkis	0.93	0.93	0.93	763

2.4. Duomenų klasifikavimas naudojant dirbtinius neuroninius tinklus

Dirbtinio neuroninio tinklo veikimas priklauso nuo architektūros bei įvairių reguliuojamų parametrų pasirinkimo. Kaip aprašyta 1.4 skyriuje, naudojantis įvairių tipų neuronų sluoksniais galima sukurti nuo paprastų iki itin sudėtingų tinklų architektūrų naudojant skirtingas aktyvacijos funkcijas bei kitus parametrus. Šiame darbe tiriama kuriamo Ramano spektrų klasifikavimo modelio tikslumo priklausomybė nuo sluoksnių skaičiaus bei mokymosi parametrų.

Naudojant dirbtinius neuroninius tinklus, visi duomenų parametrai buvo papildomai paruošiami. Kadangi naudojamo duomenų rinkinio klasių vardai yra kategoriniai, jų tiesiogiai panaudoti tinklo apmokyme negalima ir yra reikalinga atlikti duomenų transformaciją juos specifiškai paruošiant naudojimui. Panaudojus *LabelEncoder* modulį ir transformavus klasių vardus į skaitines reikšmes atsiranda problema, kad tinklas išmoks jų eilės tvarką ar klasių vardų hierarchiją. Šią problemą padeda išspręsti *Scikit-learn* bibliotekos *OneHotEncoder* modulis, kuris kiekvieną klasės vardą paverčia nauju binariniu požymiu. Tai reiškia, kad prie kiekvieno iš tiriamųjų spektrų pridedamos 8 naujos reikšmės, kurios įgyja 0 arba 1 vertes. Kaip ir ankstesniu atveju, visi duomenys buvo 80/20 santykiu padalinti į apmokymo ir testavimo duomenų rinkinius.

2.4.1. Klasifikatoriaus modelio kūrimas

Dirbtinio neuroninio tinklo modelis buvo kuriamas naudojantis *Keras* bibliotekos *Sequential* moduliu. Šis modulis skirtas kurti tiesioginio sklidimo neuroninių tinklų architektūras nuosekliai jungiant įvairių tipų bei dydžių sluoksnius. Klasifikatoriaus kūrimas buvo pradedamas nuo kuo paprastesnės dirbtinio neuroninio tinklo architektūros siekiant išsiaiškinti koks sluoksnių skaičius bei kokie mokymosi parametrai optimaliausiai tinka Ramano spektrams klasifikuoti.

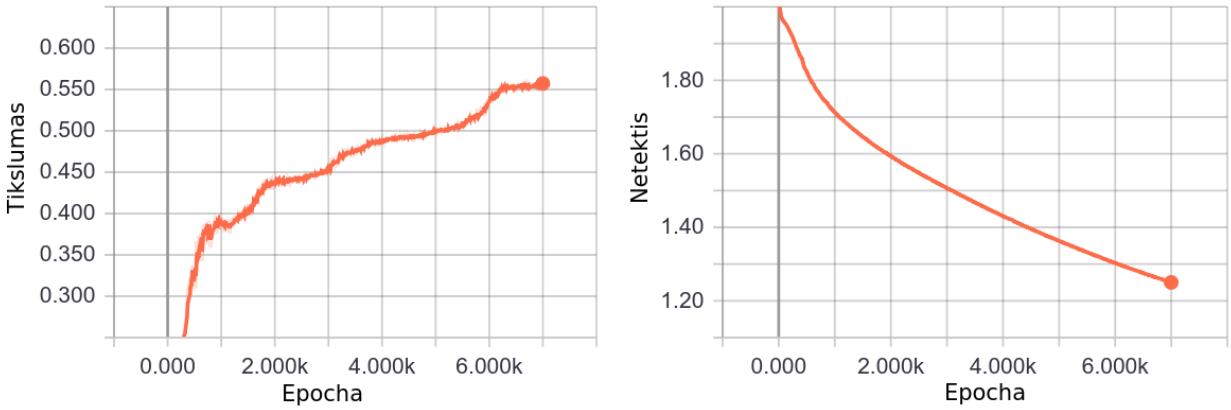
Visų pirma pradėta nuo paprasto bazinio dvisluoksnio modelio naudojantis tuo pačiu duomenų rinkiniu kaip ir tiriant atraminių vektorių mašinų metodą. Šiam rinkiniui nebuvo atlikti jokie duomenų paruošimo darbai, išskyrus klasių vardų transformaciją aprašytą 2.4 skyriuje. Tinklo architektūra pavaizduota 7 lentelėje. Visi tinkle naudojami sluoksniai plačiau aptarti 1.4.2 skyrelyje. Tinklo veikimo pagrindas sudarytas iš vieno pilnai sujungto sluoksnio, pavadinto *dense1*, kuris implementuotas panaudojant *Keras* bibliotekos *Dense* modulį. Šiam sluoksniui parinkta išlyginto tiesinio vieneto (sutr. *ReLU*) aktyvacijos funkcija. *Softmax* sluoksnis duomenų klasifikacijos neat-

lieka, tik transformuoja išėjimo vertes į tikimybes, kad duomenų įrašas priklauso konkrečiai klasei. Iš viso tinklo modelį sudarė 912968 mokomieji parametrai.

Sluoksnis (tipas)	Talpa	Parametrų skaičius
dense1 (Dense)	951	905352
softmax (Dense)	8	7616

3 lentelė. Ramano spektrų klasifikacijai naudoto dirbtinio neuroninio tinklo architektūra.

Šis tinklas buvo mokomas 7000 epochų, naudojant stochastinio gradientinio nusileidimo (angl. *stochastic gradient descent*) algoritmą ir kategorinės kryžminės entropijos (angl. *categorical cross entropy*) klaidos funkciją. *Batch size* parametras, kuris nurodo koks duomenų kiekis bus naudojamas apmokant tinklą vienos iteracijos metu buvo nustatytas 128. Naudojant *TensorBoard* neuroninių tinklų vizualizavimo bei derinimo įrankį buvo gauta 16 iliustracija, kurios kairėje pusėje matyti modelio tikslumo augimas apmokant tinklą kiekvienos epochos metu, dešinėje pusėje matyti klaidos funkcijos praradimo kritimas. Klaidos funkcijos praradimo kreivė atsako į klausimą ar tinklas apmokymo metu sugeba teisingai mokytis. Tiriant šią kreivę tikimasi pamatyti mažėjimo tendenciją. Ši tendencija reiškia, jog tinklas bėgant laikui sureguliuoja jungčių tarp neuronų svorius taip, kad mažėtų klaidų kiekis. Kuo mažesnį klaidų kiekį apmokymo metu sugeba pasiekti tinklas, tuo tikslesnį klasifikacijos rezultatą jis pasiekia.



16 pav. Kairėje iliustracijos pusėje matyti modelio tikslumo augimas apmokant tinklą kiekvienos epochos metu, dešinėje pusėje - klaidos funkcijos praradimo kitimas. Skaičiavimams naudotas neapdorotas apmokymui skirtas duomenų rinkinys.

Kaip matome iš iliustracijos tinklas pasiekė vos 55% tikslumą naudojant apmokymui skirtus duomenis. Panaudojus *Evaluate* modulį buvo apskaičiuotas modelio tikslumas su testavimui skirtais duomenimis. Rezultatas siekė 54.91% tikslumą ir 1.253 praradimą. Modelio apmokymas truko 9 min. 16 sec. naudojant grafinį procesorių aprašytą 2.1 skyriuje.

2.4.2. Klasifikatoriaus tikslumo priklausomybė nuo mokymosi parametrų

Siekiant sukurti dirbtiniais neuroniniais tinklais paremtą klasifikatorių, kuris kuo tiksliau klasifikuoja tiriamus Ramano spektrus buvo tikrinama tikslumo priklausomybė nuo tinklo mokymosi parametrų. Geriausiems parametrams rasti buvo naudojamas *Scikit-learn* bibliotekos *GridSearchCV* modulis, kuris randa geriausius modelio parametrus iš pateiktos parametrų aibės apskai-

čiuodamas kiekvieno modelio tikslumą ir juos palygindamas. Ieškomus parametrus sudaro optimizavimo metodas, mokymosi greitis bei momentas. Dažniausiai daugiaklasei klasifikacijai naudojami optimizavimo metodai yra stochastinis gradientinis nusileidimas (angl. *stochastic gradient descent*, sutr. *SGD*) su *Nesterov* momentu ir *Adam* metodas [15]. Šiuose otimizavimo metoduose naudojami skirtingi parametrai. Todėl optimaliausiem modelio parametram rasti naudojantis SGD metodu buvo ieškoma mokymosi greičio, momento bei *Nesterov* reikšmės, o *Adam* metodu - tik mokymosi greičio.

Ieškant geriausią modelio tikslumą duodančių parametrų *GridSearchCV* modulis sukuria daug atskirų klasifikatorių ir lygina jų tikslumus. Tai reikalauja daug skaičiuojamosios galios ir užtrunka nemažai laiko. Dėl šios priežasties iki 50 buvo sumažintas neuroninių tinklų apmokymui naudojamų epochų skaičius. Su tokiu epochų skaičiumi modeliai nepasiekia galimo aukščiausio tikslumo, bet jis tinka išrinkti geriausiai parametrų kombinacijai.

Tiriant *SGD* metodą parametrų aibė pateikta *GridSearchCV* moduliui sugeneruota naudojant dažniausiai pasitaikančias tiriamų laukų reikšmes. Mokymosi greičio vertės buvo parenkamos nuo 0.0001 iki 0.1 iš viso tiriant 4 reikšmes. Momento vertės buvo keičiamos nuo 0 iki 0.9 iteratyviai didinant po 0.2 iš viso tiriant 6 reikšmes. *Nesterov* momentas galimai įgyja tik dvi reikšmes: *True* arba *False*. Iš viso tiriant SGD metodą patikrintos 48 parametrų kombinacijos. 4 lentelėje pavaizduotos mokymosi greičio, momento bei *Nesterov* momento reikšmių kombinacijos, kurioms esant buvo pasiektas didžiausias modelio tikslumas.

4 lentelė. Klasifikatoriaus tikslumo priklausomybė nuo mokymosi parametrų tiriant stochastinio gradientinio nusileidimo optimizavimo metodą.

Mokymosi greitis	Momentas	*Nesterov* momentas	Modelio tikslumas
0.1	0.4	True	0.403923
0.1	0.6	True	0.434508
0.1	0.8	False	0.412234
0.1	0.8	True	0.426529
0.1	0.9	False	0.442819
0.1	0.9	True	0.467753

Iš lentelės matome, kad didžiausias modelio tikslumas (46.78%) yra pasiekiamas su mokymosi greičio reikšme 0.1, momento reikšme lygia 0.9 bei teigiama *Nesterov* momento reikšme.

Tiriant *Adam* metodą mokymosi greičio vertės buvo parenkamos nuo 0.0001 iki 0.3 iš viso tiriant 6 reikšmes. 5 lentelėje pavaizduotos visos tiriamų parametrų reikšmės ir gautos modelio tikslumo vertės.

5 lentelė. Klasifikatoriaus tikslumo priklausomybė nuo mokymosi greičio parametro naudojant *Adam* optimizavimo metodą.

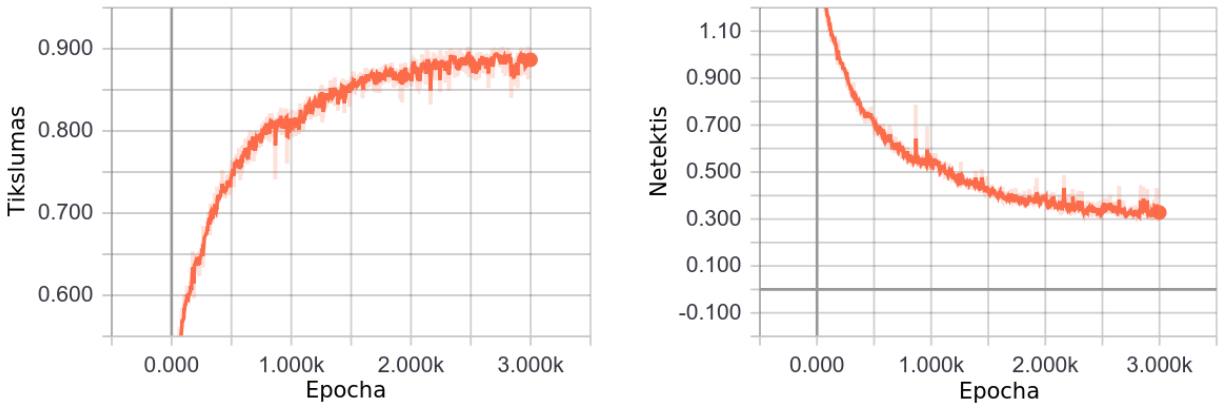
Mokymosi greitis	Modelio tikslumas
0.0001	0.233710
0.001	0.234043
0.01	0.333112
0.1	0.290226
0.2	0.235705
0.3	0.211410

Iš lentelės matome, kad didžiausias modelio tikslumas (33.31%) pasiektas mokymosi greičio reikšmę nustačius 0.01. Tačiau tai ženkliai nusileidžia geriausiam pasiektam tikslumui naudojant SGD metodą, todėl *Adam* metodas toliau naudojamas nebuvo. Tolimesniam modelio kūrimui buvo pasirinktos stochastinio gradientinio nusileidimo optimizavimo metodas su mokymosi greičio reikšme 0.1, momento reikšme lygia 0.9 bei teigiama *Nesterov* momento reikšme.

6 lentelė. Klasifikatoriaus tikslumo priklausomybė nuo *Batch size* parametro nurodančio koks duomenų kiekis bus naudojamas apmokant modelį vienos iteracijos metu.

Batch size parametro vertė	Modelio tikslumas
8	0.523604
16	0.487699
32	0.465426
64	0.538564
128	0.466396
256	0.441489
512	0.372340

Siekiant kuo tiksliau apmokyti kuriamą modelį taip pat tirta *Batch size* parametro daroma įtaka modelio tikslumui. Šis parametras nurodo koks duomenų kiekis bus naudojamas apmokant tinklą vienos duomenų apmokymo iteracijos metu. 6 lentelėje pavaizduotos tirtos šio parametro reikšmės ir gautos modelio tikslumo vertės. Iš lentelėje pavaizduotų rezultatų matyti, kad geriausias modelio tikslumas (53.86%) naudojant 50 epochų yra pasiekiamas su *Batch size* parametro reikšme 64.



17 pav. Iliustracijoje pavaizduota modelio tikslumo bei klaidos funkcijos praradimo kitimas apmokius tinklą su geriausiais rastais mokymosi parametrais.

Radus geriausius mokymosi parametrus Ramano spektro duomenims klasifikuoti nutarta apmokyti tinklą su šiais parametrais padidinus duomenų apmokymui naudojamų epochų skaičių iki 3000. Buvo gautas 89.25% modelio tikslumas ir 0.3908 klaidos funkcijos praradimas. Modelio tikslumo bei klaidos funkcijos praradimo kitimas pavaizduotas 17 iliustracijoje. Buvo naudojami šie parametrai:

- Stochastinio gradientinio nusileidimo optimizavimo metodas;
- Mokymosi greičio reikšmė - 0.1;
- Momento reikšmė - 0.9;

- Teigiama *Nesterov* momento reikšmė;
- *Batch size* parametro reikšmė - 64.

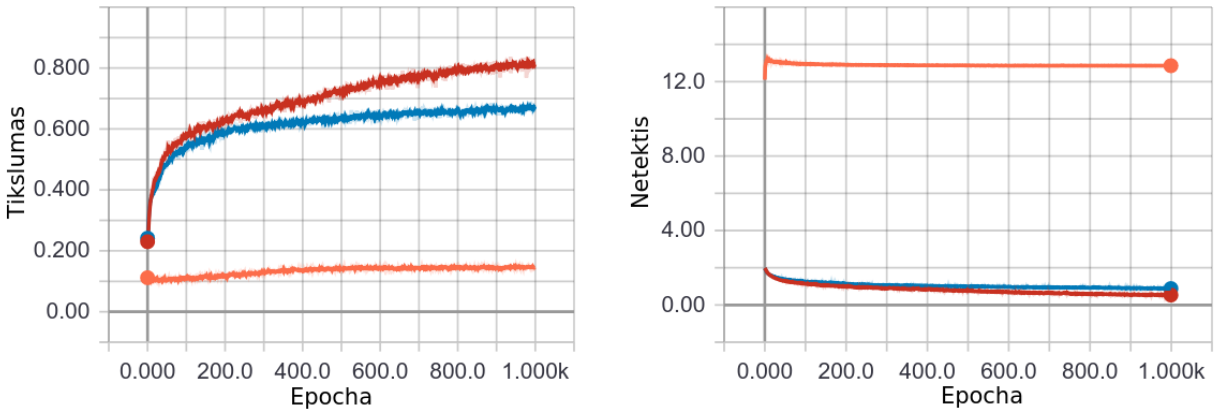
2.4.3. Tinklo architektūros derinimas

Šiame poskyryje tiriamas tinklo architektūros daromas poveikis kuriamo modelio tikslumui bei klaidos funkcijos praradimui. Ieškoma tinkamiausio sluoksnių kiekio bei jų parametrų reikšmių. Kol kas nėra aiškiai apibrėžto būdo kaip parinkti geriausiai modeliui tinkančią tinklo architektūrą. Kiekvienam uždaviniui spręsti kuriami skirtingą sluoksnių kiekį turintys neuroniniai tinklai ir bandoma parinkti geriausią tikslumą duodančius parametrus.

Ieškant tinkamiausios tinklo architektūros Ramano spektrams klasifikuoti pradedama nuo nesudėtingo, vos iš kelių sluoksnių sudaryto, bazinio tinklo modelio, aprašyto 2.4.1 skyrelyje. Visų pirma tiriama kuri iš aktyvacijos funkcijų geriausiai tinka eksperimente naudojamam duomenų rinkiniui klasifikuoti. Tikrinamos trys 1.4.1 skyrelyje aprašytos aktyvacijos funkcijos: sigmoidinė, hiperbolinio tangento ir išlyginto tiesinio vieneto. Kaip ir ieškant geriausių mokymosi parametrų, kiekvienai aktyvacijos funkcijai buvo kuriamas atskiras modelis ir skaičiuojamas šio modelio tikslumas naudojant testavimui skirtus duomenis. Modeliai buvo apmokomi su 2.4.2 skyrelyje gautais geriausiais mokymosi parametrais naudojant 1000 epochų. Rezultatai pateikti 7 lentelėje. Modelių tikslumo bei klaidos funkcijos praradimo kitimas pavaizduotas 18 paveikslėlyje.

7 lentelė. Klasifikatoriaus tikslumo priklausomybė nuo naudojamos aktyvacijos funkcijos.

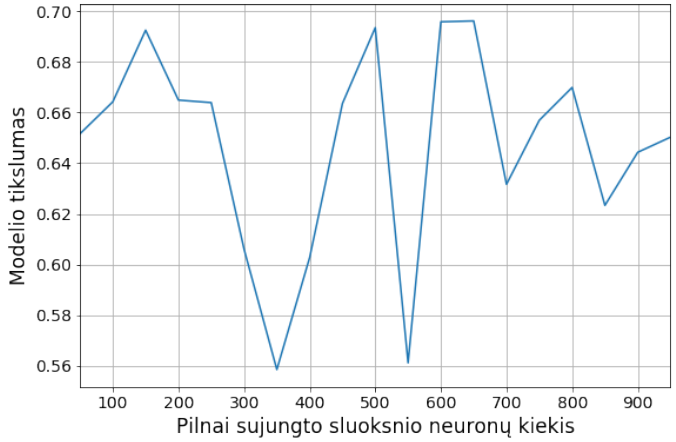
Aktyvacijos funkcija	Modelio tikslumas
Sigmoidinė funkcija	14.66
Hiperbolinio tangento funkcija	66.79
Išlyginto tiesinio vieneto funkcija	81.83



18 pav. Iliustracijoje pavaizduotas modelio tikslumo bei klaidos funkcijos praradimo kitimas apmokius tinklą su skirtingomis aktyvacijos funkcijomis. Raudona spalva pažymėta išlyginto tiesinio vieneto funkcija, mėlyna - hiperbolinio tangento funkcija, oranžine - sigmoidinė funkcija.

Iš pateiktų duomenų matome, kad apmokius tinklą su išlyginto tiesinio vieneto aktyvacijos funkcija buvo pasiektas kur kas geresnis rezultatas negu su kitomis funkcijomis. Dėl šios priežasties toliau bus naudojama ši aktyvacijos funkcija.

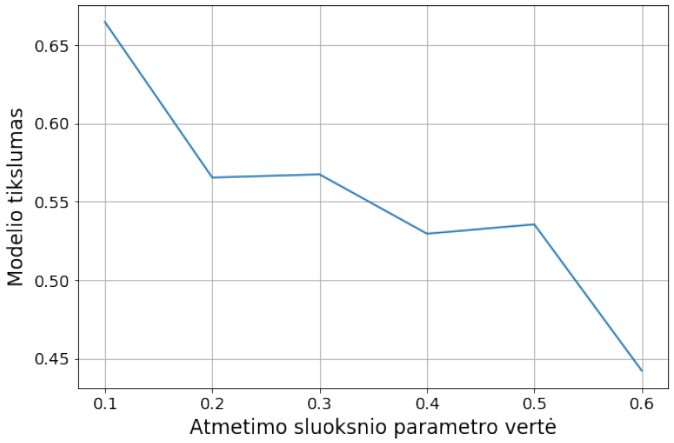
Naudojamas bazinis tinklo modelis kol kas tėra sudarytas iš dviejų sluoksnių: pilnai sujungto bei *Softmax* sluoksnio. Pastarojo išėjimų skaičius lygus tiriamų klasių skaičiui, šiuo atveju - 8. O pilnai sujungto sluoksnio neuronų kiekis pradžioje tyrimo buvo prilygintas duomenų savybių skaičiui - 951. Per didelis pilnai sujungto sluoksnio neuronų kiekis gali lemti tinklo persimokymą bei prisitaikymą prie apmokymui skirto duomenų rinkinio. Iškyla pavojus, jog tinklas pasidarys atminties banku, kuris didžiuliu tikslumu įsimena apmokymui skirtus duomenis, bet veikia prastai su dar nematytais duomenimis. Kadangi galutinis tikslas yra apmokyti tinklą klasifikuoti mėsos tipus pagal jų savybes, o ne atkartoti išmoktus duomenis, todėl yra labai svarbu parinkti tinkamą pilnai sujungtų sluoksnių talpą - neuronų kiekį. Ieškant tinkamiausio neuronų kiekio jo vertės buvo parenkamos nuo 50 iki 950 iteratyviai didinant po 50, iš viso tiriant 19 reikšmių. 19 iliustracijoje pavaizduota gautų modelių tikslumo priklausomybė nuo pilnai sujungto sluoksnio neuronų kiekio apmokius tinklą 200 epochų.



19 pav. Iliustracijoje pavaizduota klasifikatoriaus modelio tikslumo priklausomybė nuo pilnai sujungto sluoksnio neuronų kiekio.

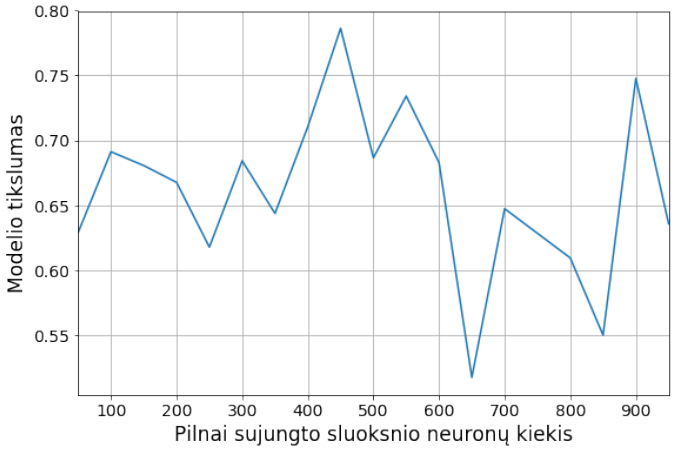
Kaip matome iš pateikto grafiko tinklas pasiekia panašų tikslumą su 150, 500, 600, 650 neuronų talpomis. Kadangi mažesnis neuronų kiekis mažina persimokymo tikimybę, toliau sekantiems tyrimams pilnai sujungto sluoksnio neuronų kiekis buvo pasirinktas 150 .

Siekiant nustatyti tinkamą tinklo sluoksnių kiekį visų pirma buvo siekta atlikti tinklo reguliarizaciją panaudojant atmetimo sluoksnį (angl. *dropout*). Šis sluoksnio tipas atmeta atsitiktinai parinktus neuronus iš tinklo taip sumažindamas persimokymo riziką. Sluoksnis implementuotas panaudojant *Keras* bibliotekos *Dropout* modulį, kuris turi vieną parametrą, nurodantį kokią dalį įvesties sluoksnis turėtų atmesti. Ieškant tinkamiausios šio parametro reikšmės jo vertės buvo parenkamos nuo 0.1 iki 0.6 didinant po 0.1, iš viso tiriant 6 reikšmes. 20 iliustracijoje pavaizduota gautų modelių tikslumo priklausomybė nuo atmetimo sluoksnio parametro reikšmės apmokius tinklą 200 epochų.



20 pav. Iliustracijoje pavaizduota klasifikatoriaus modelio tikslumo priklausomybė nuo atmetimo sluoksnio parametro.

Kaip matyti iš iliustracijos modelio tikslumas ženkliai krenta didėjant atmetimo sluoksnio parametro vertei. Taip galėjo atsitikti dėl to, kad tinklas atmetė per daug duomenų reikalingų tiksliam apmokymui. Dėl šios priežasties buvo nuspręsta pasirinkti 0.1 atmetimo sluoksnio parametro vertę ir pridėti dar vieną pilnai sujungtą sluoksnį prie tinklo architektūros. Kadangi kiekvienam tinklo sluoksniui neuronų skaičius turi būti pritaikomas iš naujo, o ne nustatomas toks pat, buvo vėl tiriama tinklo tikslumo priklausomybė nuo naujai pridėto sluoksnio neuronų kiekio. Tyrimo parametrai išliko tokie pat kaip ir ieškant optimaliausio pirmo pilnai sujungto sluoksnio neuronų kiekio. Rezultatai, gauti apmokius tinklą 200 epochų, pavaizduoti 21 iliustracijoje.



21 pav. Iliustracijoje pavaizduota klasifikatoriaus modelio tikslumo priklausomybė nuo antro pilnai sujungto sluoksnio neuronų kiekio.

Iš pateiktos iliustracijos matyti, kad tinklas pasiekia aukštą tikslumą su antro pilnai sujungto sluoksnio talpa, lygia 450 neuronų. Įsitikinus, kad pridėtas antras pilnai sujungtas sluoksnis pagerino tinklo tikslumą apie 10 % nutarta patikrinti ar pridėjus dar vieną atmetimo ir pilnai sujungto sluoksnio porą tinklo tikslumas pagerėtų. Šiam eksperimentui atmetimo sluoksnio parametro reikšmė buvo parinkta 0.1, o pilnai sujungto sluoksnio neuronų kiekis buvo nustatomas identišku būdu kaip ir pirmam bei antram šio tipo sluoksniui. Gauti rezultatai parodė, kad geriausią tinklo tikslumą pridėjus atmetimo ir pilnai sujungtą sluoksnį pavyko gauti su pastarojo sluoksnio talpa, lygia 450. Tačiau gautas modelio tikslumas, lyginant su dviem pilnai sujungtus sluoksnius turinčiu

modeliu, nukrito 8 %. Todėl šių dviejų sluoksnių buvo atsisakyta ir laikoma, kad pavyko rasti optimalią tinklo architektūrą.

2.4.4. Dirbtinių neuroninių tinklų klasifikatoriaus apibendrinimas

Kuriamo Ramano spektrų klasifikatoriaus architektūra buvo tiriama pridedant papildomus sluoksnius, nustatant sluoksnių parametrus ir apskaičiuojant gautą modelio tikslumą. 2.4.3 skyrelyje rasta geriausią tikslumą rodanti dirbtinio neuroninio tinklo architektūra, kuri pavaizduota 8 lentelėje. Tinklą sudaro du pilnai sujungti sluoksniai, atmetimo bei *Softmax* sluoksnis. Mokomųjų parametrų skaičius siekia 214358.

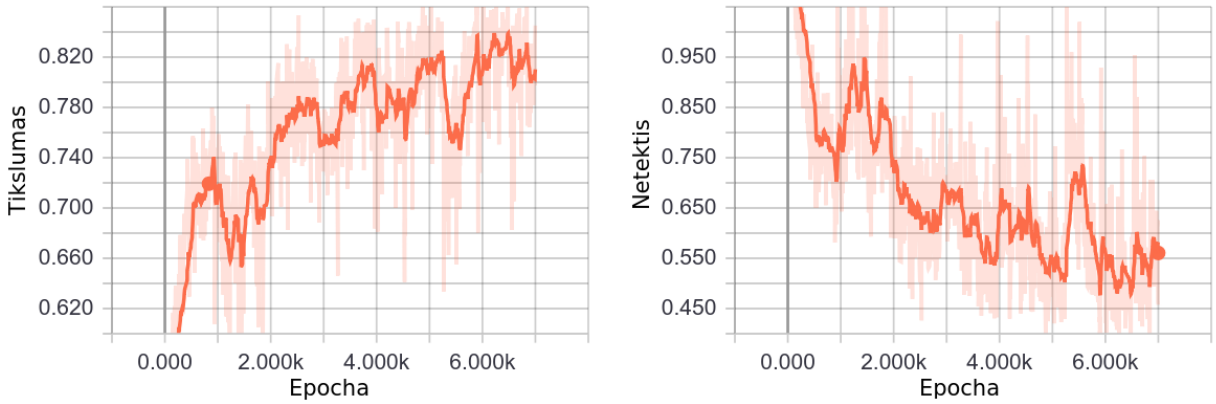
Sluoksnis (tipas)	Talpa	Parametrų skaičius
dense1 (Dense)	150	142800
drop1 (Dropout)	150	0
dense2 (Dense)	450	67950
softmax (Dense)	8	3608

8 lentelė. Ramano spektrų klasifikacijai naudojamo dirbtinio neuroninio tinklo architektūra.

Apmokant klasifikatoriaus modelį buvo naudojama stochastinio gradientinio nusileidimo optimizavimo metodas su teigiama *Nesterov* momento reikšme ir kategorinės kryžminės entropijos klaidos funkcija. Kiti mokymosi parametrai:

- Mokymosi greičio reikšmė - 0.1;
- Momento reikšmė - 0.9;
- *Batch size* parametro reikšmė - 64.

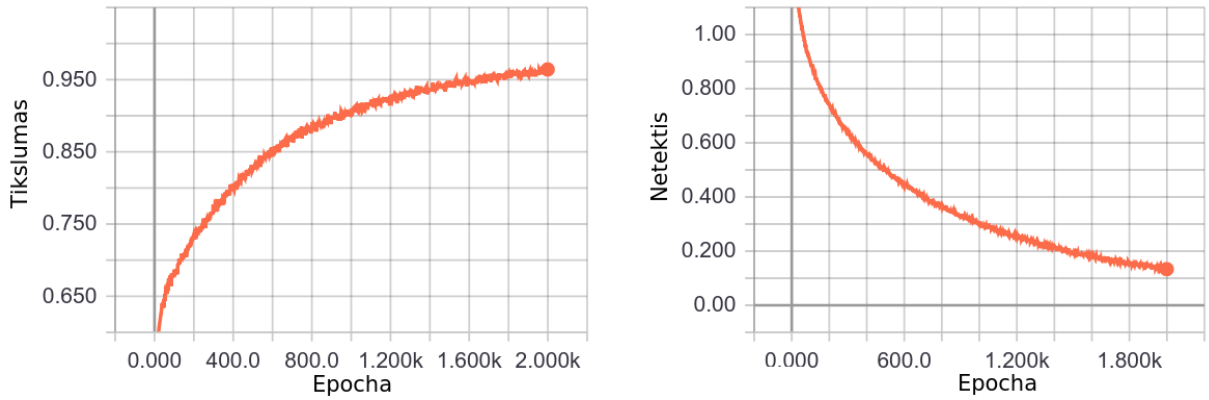
Tiriant modelio savybes visų pirma buvo išmatuota tikslumo bei klaidos funkcijos praradimo kitimai apmokymui naudojant 7000 epochų. Rezultatai pavaizduoti 22 paveikslėlyje. Apskaičiuotas modelio tikslumas su testavimui skirtais duomenimis siekė 81.91% tikslumą ir 0.5325 praradimą. Modelio apmokymas truko 19 min. 39 sec. naudojant grafinį procesorių.



22 pav. Kairėje iliustracijos pusėje matyti modelio tikslumo augimas apmokant tinklą kiekvienos epochos metu, dešinėje pusėje - klaidos funkcijos praradimo kitimas.

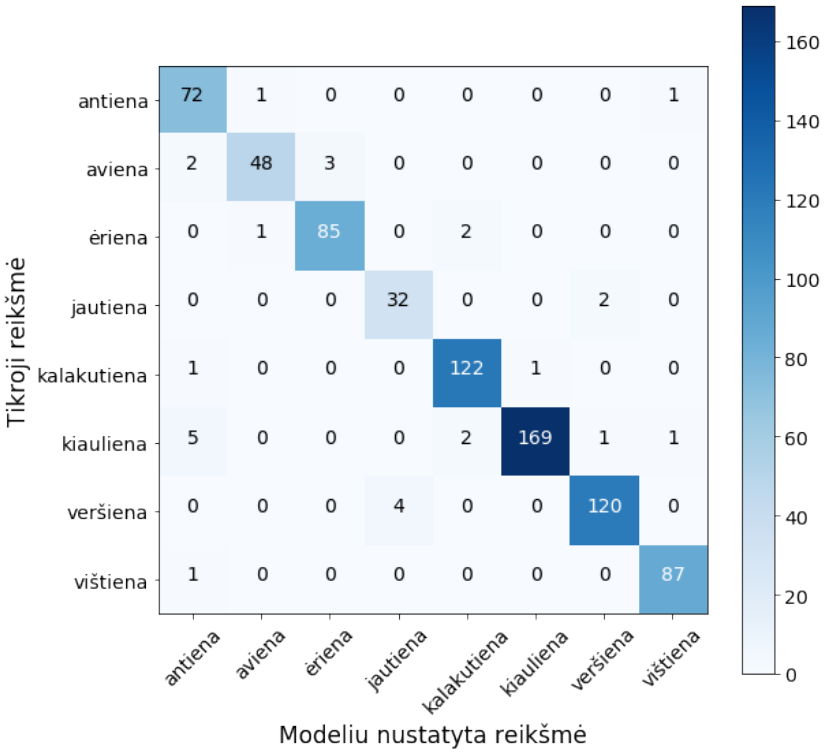
Kadangi modelio tikslumas nebuvo pakankamas, nuspręsta tinklą su tokiais pat parametrais apmokyti paruošus duomenis - atlikus standartinį duomenų normalizavimą panaudojant tokius pat

parametrus kaip tiriant atraminį vektorių mašinų metodą. Modelio tikslumo bei klaidos funkcijos praradimo kitimas pavaizduotas 23 iliustracijoje.



23 pav. Iliustracijoje pavaizduotos modelio tikslumo metrikos apmokius tinklą su duomenų rinkiniu, kuriam atliktas standartinis duomenų normalizavimas.

Šiuo atveju apskaičiuotas modelio tikslumas su testavimui skirtais duomenimis siekė 96.85% tikslumą ir 0.1334 praradimą. Modelio apmokymui buvo naudojama 2000 epochų, o procesas truko 3 min. 49 sec. naudojant grafinį procesorių. Tikslumui įvertinti apskaičiuota modelio painiavos matrica, kuri pavaizduota 24 iliustracijoje. Ji parodo teisingų bei klaidingų modelio spėjimų skaičių bei klaidų pasiskirstymą kiekvienai klasei. Iš pateiktos iliustracijos galime pamatyti, kad modelis pakankamai tiksliai klasifikavo kiekvieną iš tiriamų klasių. Daugiausiai klaidų gauta klasifikuojant antieną - sumaišius su kiauliena, bei jautieną - sumaišius su veršiena.



24 pav. Iliustracijose pavaizduota modelio painiavos matrica.

Taip pat apskaičiuotos dažniausiai naudojamos metrikos klasifikatoriaus kokybei nustatyti, kurios pavaizduotos 9 lentelėje. Šioje lentelėje matomi modelio tikslumo rezultatai kiekvienai klasei

atskirai bei bendras modelio įvertis suvidurkinus visų klasių rezultatus. Lentelėje pateikti skaičiavimai rodo, kad modelis gana tiksliai identifikavo kiekvieną iš tirtų klasių. Prasčiausi *F1-score* rezultatai buvo gauti klasifikuojant jautieną bei avieną. Itin geri rezultatai pasiekti klasifikuojant paukštieną - antieną, kalakutieną bei vištieną.

Klasė	Precision	Recall	F1-score	Spėjimų skaičius
Antiena	0.99	0.97	0.98	74
Aviena	0.96	0.91	0.93	53
Ėriena	0.97	0.97	0.97	88
Jautiena	0.89	0.94	0.91	34
Kalakutiena	0.97	0.98	0.98	124
Kiauliena	0.99	0.95	0.97	178
Veršiena	0.98	0.97	0.97	124
Vištiena	0.98	0.99	0.98	88
Vidurkis	0.97	0.96	0.97	763

9 lentelė. Apskaičiuotos dažniausiai naudojamos metrikos klasifikatoriaus kokybei nustatyti.

2.5. Duomenų klasifikavimas naudojant konvoliucinius neuroninius tinklus

Ramano spektrų klasifikavimas naudojant konvoliucinius neuroninius tinklus sparčiai išpopuliarėjo per pastaruosius keleta metų dėl efektyvaus darbo su iš anksto neparuoštais duomenimis (pvz. be bazinės linijos korekcijos). Tai pasiekti leidžia metode naudojami konvoliuciniai bei sujungimo sluoksniai. Šio tipo tinklas plačiau aprašytas 1.4.4 skyriuje. Tiriant šio metodo pritaikymą klasifikatoriaus kūrimui buvo naudojami tie patys modelio mokymosi parametrai, kurie veikė geriausiai su DNT paremtu modeliu. Taip pat naudojami duomenų rinkinio klasių vardai buvo transformuoti 2.4 skyrelyje aprašytu būdu ir duomenims atliktas standartinis duomenų normalizavimas panaudojant tokius pat parametrus kaip ir ankstesniuose tyrimuose. Šiame tyrime ieškoma optimaliausios Ramano spektrų klasifikavimui tinkančios konvoliuciniais neuroniniais tinklais paremto modelio architektūros.

2.5.1. Tinklo architektūros derinimas

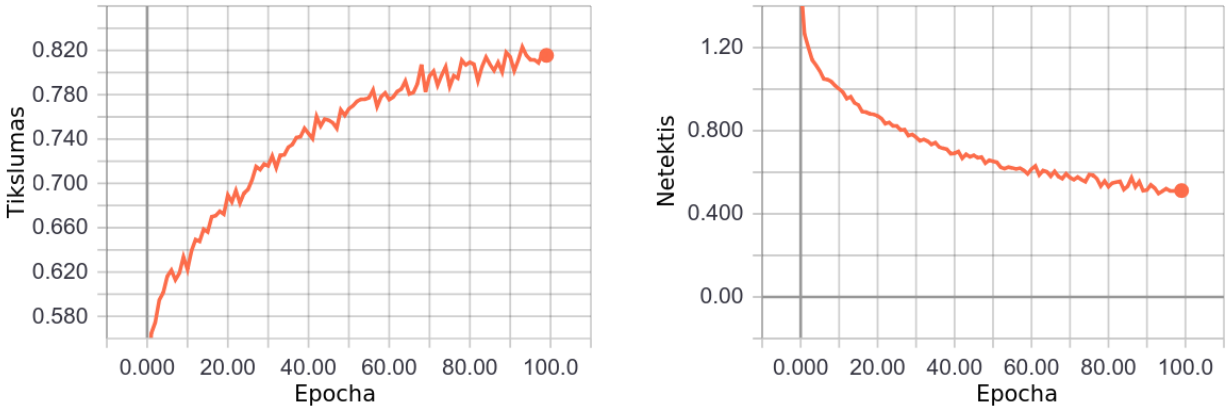
Klasifikatorius paremtas konvoliuciniais neuroniniais tinklais buvo pradėtas kurti naudojant kuo paprastesnę architektūrą, sudarytą vos iš vienos konvoliucinio bei sujungimo sluoksnių poros. Tyrime siekiama rasti tinkamiausią tinklo architektūrą palaipsniui didinant sluoksnių skaičių bei reguliuojant jų parametrus.

Bazinio tinklo architektūra pavaizduota 10 lentelėje. Tinklo veikimo pagrindą sudaro vienas konvoliucinis sluoksnis, pavadintas *conv1*, kuris atlieka duomenų savybių paieškos funkciją. Šis sluoksnis implementuotas panaudojant *Keras* bibliotekos *Conv1D* modulį. Branduolių (filtrų) kiekis lygus 10. Branduolių dydis (svorių kiekis) taip pat lygus 10. Šiam sluoksniui parinkta išlyginto tiesinio vieneto (sutr. *ReLU*) aktyvacijos funkcija. Toliau einantis sujungimo sluoksnis, pavadintas *maxpool1*, sumažina įvesties elementų kiekį (parametrų skaičių) išrenkant didžiausias į nustatyto dydžio filtrą patenkančių elementų reikšmes. Filtro dydis buvo pasirinktas lygus 3, sumažinantis

Sluoksnis (tipas)	Talpa	Parametrų skaičius
conv1 (Conv1D)	942 (10 sluoksnių)	110
maxpool1 (MaxPooling1D)	314 (10 sluoksnių)	0
flat1 (Flatten)	3140	0
dropout1 (Dropout)	3140	0
softmax (Dense)	8	25128

10 lentelė. Ramano spektrų klasifikacijai naudoto konvoliucinio neuroninio tinklo bazinė architektūra.

parametrų kiekį tris kartus. Branduolių sujungimui panaudotas *flat1* sluoksnis. Taip pat atlikta duomenų atmetimo operacija *Dropout*, atmetant pusę atsitiktinai parinktų parametrų. Iš viso tinklo modelį sudarė 25238 mokomieji parametrai.



25 pav. Kairėje iliustracijos pusėje matyti modelio tikslumo augimas apmokant tinklą kiekvienos epochos metu, dešinėje pusėje - klaidos funkcijos praradimo kitimas.

Tinklo modelis buvo mokomas 100 epochų naudojant tokius pačius modelio mokymosi parametrus, kurie veikė geriausiai su DNT paremtu modeliu. Rezultatai pavaizduoti 25 paveikslėlyje. Apskaičiuotas modelio tikslumas su testavimui skirtais duomenimis siekė 84.14% tikslumą ir 0.4631 praradimą. Modelio apmokymas truko viso labo 23 sekundes naudojant grafinį procesorių.

Branduolių kiekis	Branduolio dydis	Modelio tikslumas
16	7	0.923870
32	3	0.921210
32	7	0.920213
64	5	0.938830
64	7	0.945811
96	5	0.925811
128	7	0.923870

11 lentelė. Klasifikatoriaus tikslumo priklausomybė nuo konvoliucinio sluoksnio parametrų.

Siekiant nustatyti tinkamiausią tinklo architektūrą visų pirma buvo ieškomi optimaliausi konvoliucinio sluoksnio parametrai - branduolių kiekis ir dydis. Kuo daugiau branduolių turi tinklas, tuo daugiau duomenų savybių sugeba atpažinti. Ieškant tinkamiausios šio parametro reikšmės jo

vertės buvo parenkamos nuo 8 iki 128, iš viso tiriant 6 reikšmes. Branduolio dydžio reikšmės buvo parenkamos nuo 3 iki 7, iš viso tiriant 3 reikšmes. 11 lentelėje pavaizduota konvoliucinio sluoksnio parametrų kombinacijos, kurioms esant buvo pasiektas didžiausias modelio tikslumas apmokius tinklą 100 epochų. Modelio tikslumas apskaičiuotas naudojant vien tik apmokymui skirtus duomenis. Iš lentelės matome, kad didžiausias modelio tikslumas (94.58%) buvo pasiektas pasirinkus 64 branduolius su dydžio verte, lygia 7.

Radus geriausius konvoliucinio sluoksnio parametrus toliau buvo tiriama sujungimo sluoksnio daroma įtaka modelio tikslumui. Šio sluoksnio svarbiausias reguliuojamas parametras - filtro dydis, kuris nurodo kiek kartų bus sumažintas prieš tai esančio sluoksnio parametrų kiekis išrenkant didžiausias į filtrą patenkančias reikšmes. Ieškant tinkamiausios šio parametro reikšmės buvo tikrinamos dažniausiai naudojamos šio parametro vertės. Rezultatai pavaizduoti 12 lentelėje. Iš lentelės matome, kad didžiausias modelio tikslumas (95.54%) buvo pasiektas pasirinkus filtro dydį, lygų 2.

Filtro dydis	Modelio tikslumas
2	0.955452
3	0.941157
5	0.920878
7	0.896941
10	0.855053

12 lentelė. Klasifikatoriaus tikslumo priklausomybė nuo sujungimo sluoksnio filtro dydžio parametro.

Siekiant rasti tinkamiausią tinklo architektūrą buvo tiriama koks konvoliucinių bei sujungimo sluoksnių kiekis veikia optimaliausiai. Tyrime buvo pridedama po vieną konvoliucinio bei sujungimo sluoksnių porą kiekvieną kartą pakartojant apmokymo procesą ir apskaičiuojant modelio tikslumą. Rezultatai pavaizduoti 13 lentelėje, iš kurios matome, kad didžiausias modelio tikslumas (95.15%) buvo pasiektas pasirinkus 3 konvoliucinių bei sujungimo sluoksnių poras. 4 sluoksnių porų modelio tikslumas pradėjo mažėti, todėl sluoksnių didinimas buvo nutrauktas. Modelio tikslumas įvertintas naudojant testavimui skirtą duomenų aibę.

Sluoksnių kiekis	Modelio tikslumas
1	0.90563564
2	0.92136304
3	0.95150720
4	0.94626474

13 lentelė. Klasifikatoriaus tikslumo priklausomybė nuo konvoliucinio bei sujungimo sluoksnių porų kiekio.

Tolesniame tyrime buvo keičiamas pilnai sujungtų sluoksnių kiekis bei tūris siekiant padidinti konvoliuciniuose sluoksniuose atpažintų duomenų ypatybių klasifikacijos tikslumą. Tačiau modifikuojant šiuos sluoksnius nepavyko išgauti geresnio modelio tikslumo. Todėl tolesnių tyrimų su šiais sluoksniais buvo atsisakyta ir laikoma, kad pavyko rasti optimalią konvoliucinio neuroninio tinklo architektūrą.

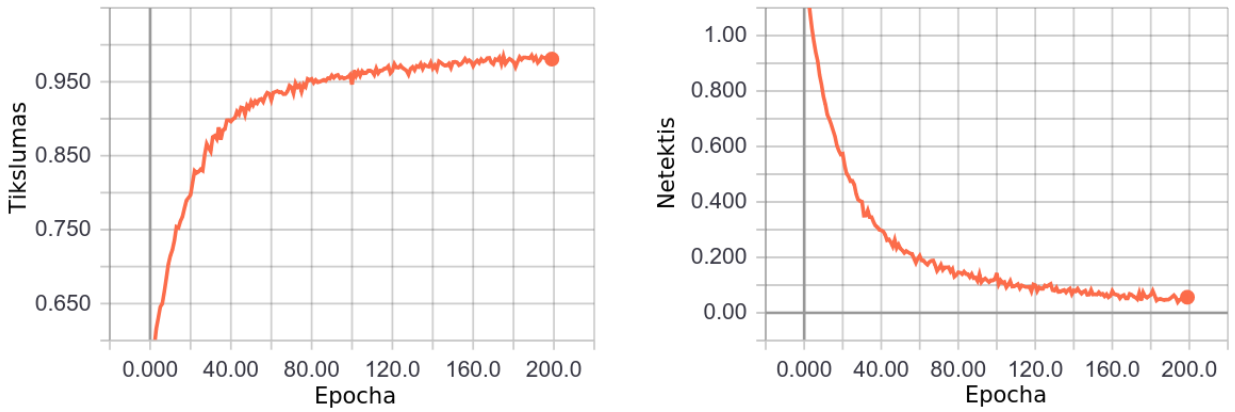
2.5.2. Konvoliucinių neuroninių tinklų klasifikatoriaus apibendrinimas

Klasifikatorius paremto konvoliuciniais neuroniniais tinklais architektūra buvo tirta ieškant optimalaus konvoliucinio bei sujungimo sluoksnių kiekio bei reguliuojant šių sluoksnių parametrus. 2.5.1 skyriuje rasta tiksliausiai veikianti dirbtinio neuroninio tinklo architektūra, kuri pavaizduota 14 lentelėje. Tinklą sudaro trys konvoliucinių bei sujungimo sluoksnių poros, atmetimo bei *Softmax* sluoksnis. Mokomųjų parametrų skaičius siekia 115848.

Sluoksnis (tipas)	Talpa	Parametrų skaičius
conv1 (Conv1D)	945 (64 sluoksniai)	512
maxpool1 (MaxPooling1D)	472 (64 sluoksniai)	0
conv2 (Conv1D)	466 (64 sluoksniai)	28736
maxpool2 (MaxPooling1D)	233 (64 sluoksniai)	0
conv3 (Conv1D)	227 (64 sluoksniai)	28736
maxpool3 (MaxPooling1D)	113 (64 sluoksniai)	0
flat (Flatten)	7232	0
dropout (Dropout)	7232	0
softmax (Dense)	8	57864

14 lentelė. Ramano spektrų klasifikacijai naudojamo konvoliucinio neuroninio tinklo architektūra.

Klasifikatoriaus modelio apmokymui buvo naudojama stochastinio gradientinio nusileidimo optimizavimo metodas naudojant tokius pačius modelio mokymosi parametrus, kurie veikė geriausiai su DNT paremtu modeliu.

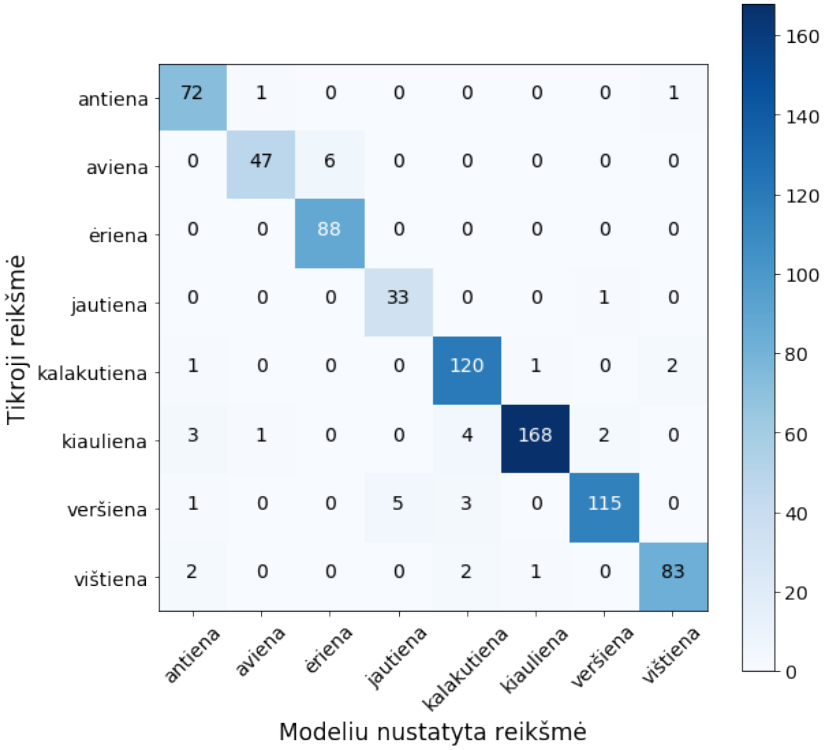


26 pav. Kairėje iliustracijos pusėje matyti modelio tikslumo augimas apmokant tinklą kiekvienos epochos metu, dešinėje pusėje - klaidos funkcijos praradimo kitimas.

Tiriant modelio veikimą buvo išmatuota tikslumo bei klaidos funkcijos praradimo kitimai apmokius tinklą 200 epochų. Rezultatai pavaizduoti 26 paveikslėlyje. Apskaičiuotas modelio tikslumas su testavimui skirtais duomenimis siekė 95.94% tikslumą ir 0.1653 praradimą. Modelio apmokymas truko 1 min. 14 sec. naudojant grafinį procesorių, 38 min. 15 sec. naudojant nešiojamą kompiuterį, aprašytą 2.1 skyriuje.

Modelio tikslumui įvertinti taip pat buvo apskaičiuota modelio painiavos matrica, kuri parodo teisingų bei klaidingų modelio spėjimų skaičių kiekvienai klasei. Painiavos matrica pavaizduota 27 iliustracijoje. Iš kurios galime pamatyti, kad modelio tikslumas bei daromų klaidų braižas yra

gan panašus į DNT modelio. Kaip ir pastarajame, daugiausiai klaidų gauta klasifikuojant antieną - sumaišius su kiauliena, bei jautieną - sumaišius su veršiena.



27 pav. Iliustracijose pavaizduota modelio painiavos matrica.

Taip pat apskaičiuotos klasifikatoriaus kokybės nustatymui dažniausiai naudojamos metrikos. Šių metrikų rezultatai kiekvienai tirtai duomenų klasei pavaizduoti 15 lentelėje. Apačioje taip pat pateiktas bendras modelio įvertis suvidurkinus visų klasių rezultatus. Iš pateiktų skaičiavimų matome, kad modelis aukštu tikslumu klasifikavo kiekvieną iš tirtų klasių. Iš *F1-score* rezultatų matome, kad modelio tikslumas yra panašus į DNT modelio. Prasčiausi šios metrikos įverčiai taip pat gauti klasifikuojant jautieną bei avieną. Geri rezultatai pasiekti klasifikuojant antieną, ėrieną bei kiaulieną.

Klasė	Precision	Recall	F1-score	Spėjimų skaičius
Antiena	0.96	0.97	0.97	74
Aviena	0.96	0.89	0.92	53
Ėriena	0.94	1.00	0.97	88
Jautiena	0.87	0.97	0.92	34
Kalakutiena	0.93	0.97	0.95	124
Kiauliena	0.99	0.94	0.97	178
Veršiena	0.97	0.93	0.95	124
Vištiena	0.97	0.94	0.95	88
Vidurkis	0.96	0.95	0.95	763

15 lentelė. Apskaičiuotos dažniausiai naudojamos metrikos klasifikatoriaus kokybei nustatyti.

Išvados ir rekomendacijos

- Tyrimo metu buvo atlikti mėsos Ramano spektrų klasifikavimo eksperimentai panaudojant dažniausiai spektroskopijos duomenų analizėje naudojamus kompiuterio mokymosi metodus: atraminių vektorių mašinų metodą, dirbtinius neuroninius tinklus bei konvoliucinius neuroninius tinklus. Didžiausias klasifikavimo tikslumas buvo gautas naudojant dirbtinius neuroninius tinklus, kuris siekė 96.85%. Tačiau pakankamai aukštas tikslumas buvo pasiektas su visais tirtais metodais.

- Naudojant SVM ar kitus panašaus tipo klasifikatorius Ramano spektrų analizei yra reikalingas netrivialus duomenų apdorojimas, kuris apsunkina tyrimo eigą. Šiame tyrime duomenims paruošti buvo atliktas duomenų normalizavimas, bazinės spektro linijos korekcija ir pritaikytas principinių komponenčių analizės metodas duomenų dimensijų mažinimui. Nustatyta, kad bazinės spektro linijos ištiesinimui naudotas asimetrinis mažiausių kvadratų glodinimo metodas geriausiai veikia su p - 10^{-4}, λ - 10^{6} parametrais. Aukštą duomenų dimensijų kiekį tiksliausiai pavyko sumažinti pritaikius PCA metodą ir panaudojus 24 pagrindines komponentes. Taip pat pastebėta, kad SVM klasifikatoriaus tikslumą pagerino standartinis duomenų normalizavimas.

- Tiriant neuroniniais tinklais grįstus metodus buvo ieškoma aukščiausią klasifikacijos tikslumą suteikiančių tinklų architektūrų bei nuodugniai tiriama įvairių tinkluose naudojamų parametrų daroma įtaka mokymosi procesui. Pastebėta, kad klasifikuojant spektrus neuroniniais tinklais nėra būtina atlikti sudėtingas duomenų paruošimo operacijas, tokias kaip bazinės linijos korekcija ar duomenų dimensijų mažinimas. Tai ženkliai palengvino modelių kūrimo procesą. Taip pat nustatyta, kad atlikus standartinį duomenų normalizavimą ženkliai išaugo neuroninių tinklų klasifikatorių tikslumas.

Ateities tyrimų planas

Baigiamojo magistrinio darbo metu buvo nustatyta, kad Ramano sklaidos spektrų klasifikavimo uždaviniui spręsti yra tinkami visi trys tirti metodai. Kadangi klasifikavimo modelio kūrimui naudojant SVM metodą reikalingos papildomos duomenų paruošimo procedūros, rekomenduojama naudoti neuroninius tinklus. Tačiau kiekvienas iš eksperimentuose sukurtų modelių galėtų būti tobulinamas. Rekomenduojamos tokios tolimesnės darbų gairės:

- Naudojant SVM klasifikatorių papildomas dėmesys galėtų būti skirtas siekiant optimizuoti bazinės linijos korekciją. Galėtų būti panaudoti kiti bazinės linijos iškraipymų eliminavimui taikomi algoritmai, aprašyti 1.5.1 skyriuje.
- Siekiant padidinti klasifikavimo tikslumą neuroninių tinklų modeliuose galėtų būti išbandyta daugiau tinklo architektūros variantų, įterpiant papildomų sluoksnių ir reguliuojant jų parametrus.
- Siekiant sumažinti nevienodą duomenų kiekio pasiskirstymą tarp klasių turėtų būti stengiamasi gauti papildomų duomenų. Tikėtina, kad tai pagerintų modelių tikslumą.
- Kadangi realius eksperimentinius duomenis gauti yra sunku, galima būtų sugeneruoti dirbtinių sintetinių duomenų. Tai būtų galima atlikti pridedant triukšmo prie realių Ramano spektrų, stumdant spektrų signalus į šonus ar naudojant specialius algoritmus. Papildomas kiekis duomenų turėtų pagerinti kuriamų modelių tikslumą.
- Kadangi skirtingų spektroskopijos metodų duomenys turi daug panašumų, galėtų būti sukurtas generalizuotas modelis, kuris tiktų ne tik Ramano, bet ir panašių metodų duomenims klasifikuoti. Galėtų būti panaudotas mokymosi perdavimo metodas (angl. *transfer learning*) pritaikant modelį skirtingų spektrų klasifikacijai atlikti.
- Tikslesniam neuroninių tinklų architektūros parinkimui būtų galima panaudoti augančių topologijų (angl. *augmented topologies*) metodą. Tai yra genetinis algoritmas, kuris evoliucionuojant keičia neuroninių tinklų parametrus bei ieško optimaliausios architektūros.

Literatūros šaltiniai

[1] A. Gelžinis A. Verikas. Neuroniniai tinklai ir neuroniniai skaičiavimai, 2003.

[2] S. Botti G. Dipoppa A. Puiu S Almaviv. Raman spectra massive classification using artificial neural networks, 2014.

[3] C. Carey T. Boucher S. Mahadevan P. Bartholomew and M. Dya. Machine learning tools for mineral recognition and classification from raman spectroscopy, 2015.

[4] Paul H. C. Eilers Hans F.M. Boelens. Baseline correction with asymmetric least squares smoothing, 2005.

[5] Amar Budhiraja. Dropout in (deep) machine learning. https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5.

[6] V. Vapnik C. Cortes. Support-vector networks. Machine Learning, 20(3):273–297, 1995.

[7] H. Yang N. Stone B. Lafuente R. T. Downs. The power of databases: the rru project. http://rru .info, 2015.

[8] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition una ected by shift in position, 1980.

[9] J. Liu M. Osadchy L. Ashton M. Foster C. J. Solomon S. J. Gibson. Deep convolutional neural networks for raman spectrum recognition: A unified solution. Analyst, 142, 4067-4074, 2017.

[10] Dominique Gilleman. Deep learning essentials: Convolutional network. http://www.deeplearningessentials.science/convolutionalnetwork/.

[11] S. Haykin. Neural networks and learning machines, 2009.

[12] T. K. Ho. The random subspace method for constructing decision forests, 1998.

[13] I. Jolli e. Principal component analysis, 2002.

[14] Andrej Karpathy. Convolutional neural networks. http://cs231n.github.io/convolutional-networks/.

[15] Andrej Karpathy. Neural networks part 3: Learning and evaluation. http://cs231n.github.io/neural-networks-3/.

[16] William Koehrsen. Beyond accuracy: Precision and recall. https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c.

[17] S. Chen M. Zhang and Y. Liang. Baseline correction using adaptive iteratively reweighted penalized leastsquares, 2010.

[18] C. A. Lieberand A. Mahadevan-Jansen. Automated method for subtraction of fluorescence from biological raman spectra, 2003.

[19] P. Taylor. Intelligent data analysis: an introduction, 2003.

[20] V. Vapnik. The nature of statistical learning theory. Springer US, 1995.

[21] Austin Walters. Pca: Principal component analysis. https://austingwalters.com/pca-principal-component-analysis/.

[22] S. Sigurdsson J. Larsen P. A. Philipsen M. Gniadecka H. C. Wulf and L. K. Hansen. Estimating and suppressing background in raman spectra with an artificial neural network, 2003.

[23] Jessica Yung. Explaining tensorflow code for a convolutional neural network. http://www.jessicayung.com/explaining-tensorflow-code-for-a-convolutional-neural-network/.

[24] Prasad Manda Zhengmao Ye, Gregory Auner. Raman spectra calibration, extraction and neural network based training for sample identification, 2003.

[25] V. Šablinskas J. Čeponkus. Modernioji molekulių virpesinė spektrometrija, 2014.

[26] G. Dzemyda O. Kurasova J. Žilinskas. Daugiamačių duomenų vizualizavimo metodai, 2008.