



VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOSINFORMATIKOS STUDIJŲ PROGRAMA

Magistro baigiamasis darbas

**Nesubalansuotų vaizdinių duomenų problemos sprendimas
taikant giliuosius neuroninius tinklus**

Addressing imbalanced image data using deep neural networks

Dautartas Valvonis

Darbo vadovas : Prof., Dr. Aistis Raudys

Recenzentas : Dr. Andrius Vytautas Misiukas Misiūnas

**Vilnius
2025**

Santrauka

Šiame darbe nagrinėjama nesubalansuotų vaizdinių duomenų problema taikant giliuosius neuroninius tinklus. Išsprendus šią problemą būtų gaunami geresni rezultatai klasifikavimo uždaviniuose, kurie yra taikomi įvairiose srityse – medicinoje, kompiuterinėje regoje ir t.t. Aptariami trys pagrindiniai duomenų lygmens nesubalansuotų vaizdinių duomenų problemos sprendimo būdai – atsitiktinis pavyzdžių perteklinis įtraukimas, augmentacijos bei DeepSmote. Sukurti dar keturi sprendimo būdai kombinuojant minėtuosius tris. Atliekami eksperimentai pasitelkiant skirtingus sprendimo būdus ir jų kombinacijas, duomenų rinkinius bei išbalansavimo santykius siekiant įvertinti sprendimo būdų efektyvumą.

Raktiniai žodžiai: Gilieji neuroniniai tinklai, dirbtiniai neuroniniai tinklai, duomenų lygmens metodai, pavyzdžių perteklinis įtraukimas, imties retinimas, hibridinis metodas, algoritmų lygmens metodai, vertinimo metrikos, SMOTE, DeepSmote, MNIST, FMNIST, SVHN.

Summary

This work addresses the problem of unbalanced image data using deep neural networks. Solving this problem would lead to better results in classification tasks, which are applied in various fields such as medicine, computer vision, etc. Three main solutions to the problem of unbalanced image data at the data level are discussed - random oversampling, augmentations, and DeepSmote. Four more solutions are developed by combining the mentioned three. Experiments are carried out using different solutions and combinations of solutions, datasets, and imbalance ratios to evaluate the performance of the solutions.

Keywords: Deep learning, deep neural networks, artificial neural networks, data-level approaches, over-sampling, under-sampling, hybrid method, algorithmic-level approaches, evaluation metrics, SMOTE, DeepSmote, MNIST, FMNIST, SVHN.

Iliustracijų sąrašas

1	Dirbtinis ir gilusis neuroniniai tinklai	12
2	Pavyzdžių perteklinis įtraukimas	15
3	Vaizdo augmentacijos pavyzdys	17
4	GAN architektūra	18
5	DeepSMOTE architektūra [DKC23]	20
6	Imties retinimas	21
7	Hibridinis būdas	23
8	Precizijos, atkūrimo ir IoU metrikų vizualizacija	27
9	ROC kreivės grafikas	29
10	Precizijos - atkūrimo (PR) kreivės grafikas	30
11	Klasifikavimo pavyzdys	32
12	Objektų aptikimo pavyzdys	32
13	Vaizdų segmentacijos pavyzdys	33
14	MNIST duomenų rinkinys	35
15	FMNIST duomenų rinkinys	36
16	SVHN duomenų rinkinys	37
17	Koreguotų duomenų rinkinių kūrimas	38
18	Modelis nr. 1 (M1)	38
19	Modelis nr. 2 (M2)	39
20	Modelis nr. 3 (M3)	40
21	Modelis nr. 4 (M4)	40
22	DeepSmote enkoderis/dekoderis	42
23	SVHN duomenų rinkinio pavyzdžiai sukurti naudojant DeepSmote metodą	43
24	MNIST duomenų rinkinio pavyzdžiai sukurti naudojant augmentaciją	43
25	MNIST -99% atkūrimas	62
26	MNIST -99% Modelis 1 visų klasių F1 rezultatai	63
27	MNIST -99% Modelis 2 visų klasių F1 rezultatai	64
28	MNIST -99% Modelis 3 visų klasių F1 rezultatai	65
29	MNIST -99% rezultatų vidurkiai	66
30	MNIST -99,9% atkūrimas	69
31	MNIST -99,9% Modelis 1 visų klasių F1 rezultatai	70
32	MNIST -99,9% Modelis 2 visų klasių F1 rezultatai	71
33	MNIST -99,9% Modelis 3 visų klasių F1 rezultatai	72
34	MNIST -99,9% rezultatų vidurkiai	73
35	FMNIST -99% atkūrimas	76
36	FMNIST -99% Modelis 1 visų klasių F1 rezultatai	77
37	FMNIST -99% Modelis 2 visų klasių F1 rezultatai	78
38	FMNIST -99% Modelis 3 visų klasių F1 rezultatai	79
39	FMNIST -99% rezultatų vidurkiai	80
40	FMNIST -99,9% atkūrimas	83
41	FMNIST -99,9% Modelis 1 visų klasių F1 rezultatai	84
42	FMNIST -99,9% Modelis 2 visų klasių F1 rezultatai	85
43	FMNIST -99,9% Modelis 3 visų klasių F1 rezultatai	86
44	FMNIST -99,9% rezultatų vidurkiai	87
45	SVHN -99% atkūrimas	90
46	SVHN -99% Modelis 2 visų klasių F1 rezultatai	91
47	SVHN -99% Modelis 3 visų klasių F1 rezultatai	92
48	SVHN -99% Modelis 4 visų klasių F1 rezultatai	93

49	SVHN -99% rezultatų vidurkiai	94
----	---	----

Lentelių sąrašas

1	Klasifikavimo lentelė	25
2	COCO metrikos	28
3	MNIST -99% f1 rezultatų vidurkiai	45
4	MNIST -99,9% f1 rezultatų vidurkiai	46
5	FMNIST -99% f1 rezultatų vidurkiai	46
6	FMNIST -99,9% f1 rezultatų vidurkiai	47
7	SVHN -99% f1 rezultatų vidurkiai	47
8	Vidutinės F1 statistikos reikšmės kiekvienam metodui skirtinguose duomenų rinkiniuose ir disbalanso lygmenyse	48
9	MNIST duomenų rinkinio klasių svorių metodo rezultatų palyginimas su geriausiu eksperimento rezultatu	48
10	FMNIST duomenų rinkinio klasių svorių metodo rezultatų palyginimas su geriausiu eksperimento rezultatu	49
11	SVHN duomenų rinkinio klasių svorių metodo rezultatų palyginimas su geriausiu eksperimento rezultatu	49
12	MNIST rezultatai po 99% duomenų pašalinimo iš klasės su pirmuoju modeliu . . .	60
13	MNIST rezultatai po 99% duomenų pašalinimo iš klasės su antruoju modeliu . . .	60
14	MNIST rezultatai po 99% duomenų pašalinimo iš klasės su trečiuoju modeliu . . .	61
15	MNIST rezultatai po 99,9% duomenų pašalinimo iš klasės su pirmuoju modeliu . .	67
16	MNIST rezultatai po 99,9% duomenų pašalinimo iš klasės su antruoju modeliu . .	67
17	MNIST rezultatai po 99,9% duomenų pašalinimo iš klasės su trečiuoju modeliu . .	68
18	FMNIST rezultatai po 99% duomenų pašalinimo iš klasės su pirmuoju modeliu . .	74
19	MNIST rezultatai po 99% duomenų pašalinimo iš klasės su antruoju modeliu . . .	74
20	FMNIST rezultatai po 99% duomenų pašalinimo iš klasės su trečiuoju modeliu . .	75
21	FMNIST rezultatai po 99,9% duomenų pašalinimo iš klasės su pirmuoju modeliu .	81
22	FMNIST rezultatai po 99,9% duomenų pašalinimo iš klasės su antruoju modeliu . .	81
23	FMNIST rezultatai po 99,9% duomenų pašalinimo iš klasės su trečiuoju modeliu .	82
24	SVHN rezultatai po 99% duomenų pašalinimo iš klasės su antruoju modeliu	88
25	SVHN rezultatai po 99% duomenų pašalinimo iš klasės su trečiuoju modeliu	88
26	SVHN rezultatai po 99% duomenų pašalinimo iš klasės su ketvirtuoju modeliu . .	89

Turinys

Santrauka	2
Summary	3
Iliustracijų sąrašas	4
Lentelių sąrašas	6
Įvadas	9
1. Darbo tikslas, uždaviniai, siekiami rezultatai	10
2. Literatūros apžvalga	11
2.1. Dirbtiniai ir gilieji neuroniniai tinklai	11
2.2. Nesubalansuoti duomenys	12
2.3. Duomenų lygmens metodai	13
2.3.1. Pavyzdžių perteklinis įtraukimas	14
2.3.1.1. Atsitiktinis perteklinis įtraukimas	15
2.3.1.2. Duomenų augmentacija	16
2.3.1.3. Generatyviniai adversariniai tinklai (GAN)	17
2.3.1.4. SMOTE	18
2.3.2. Imties retinimas	20
2.3.2.1. Atsitiktinis šalinimas	21
2.3.2.2. Klasteriais pagrįstas imties retinimas	21
2.3.2.3. NearMiss	22
2.3.2.4. Tomek Links	22
2.3.2.5. Edited Nearest Neighbor (ENN)	22
2.3.3. Hibridinis būdas	23
2.3.4. Požymių atrankos ir požymių ištraukimo metodai	23
2.4. Algoritmų lygmens metodai	24
2.4.1. Klasių svoriai	24
2.4.2. Kaštams jautrus mokymasis	24
2.4.3. Ansamblio metodai	25
2.5. Vertinimo metrikos	25
2.5.1. Slenkstinių verčių metrikos	26
2.5.1.1. Precizijos - atkūrimo metrikos	26
2.5.1.2. Jautrumo - specifiškumo metrikos	27
2.5.1.3. Susikirtimas padalintas iš sąjungos (IoU)	27
2.5.1.4. COCO metrikos	27
2.5.2. Rankingo metodai ir metrikos	28
2.5.2.1. ROC	29
2.5.3. Tikimybinės metrikos	31
2.6. Uždaviniai	31
2.6.1. Klasifikavimo uždavinys	31
2.6.2. Objektų aptikimo uždavinys	32
2.6.3. Segmentacijos uždavinys	32
3. Eksperimentas	34

3.1.	Duomenų rinkiniai	34
3.1.1.	MNIST	34
3.1.2.	FMNIST	35
3.1.3.	SVHN	36
3.2.	Išbalansuotų duomenų rinkinių kūrimas	37
3.3.	Modeliai	38
3.4.	Metrikos	40
3.5.	Pradiniai rezultatai	41
3.6.	Sprendimo būdai	41
3.6.1.	DeepSmote	41
3.6.2.	Augmentacija	43
3.6.3.	Atsitiktins perteklinis įtraukimas	43
3.6.4.	Kombinuoti sprendimo būdai	43
3.7.	Eksperimento rezultatai	44
3.8.	Palyginimas su algoritmo lygmens metodu	48
	Rezultatai ir išvados	50
	Sąvokų apibrėžimai	58
	1 priedas. Eksperimento failai	59
	2 priedas. MNIST -99% rezultatai	60
	3 priedas. MNIST -99,9% rezultatai	67
	4 priedas. FMNIST -99% rezultatai	74
	5 priedas. FMNIST -99,9% rezultatai	81
	6 priedas. SVHN -99% rezultatai	88

Įvadas

Dirbtiniai neuroniniai tinklai yra naudojami įvairiose srityse: duomenų analizėje, kalbos apdorojime, vaizdų atpažinime ar nuspėjant rytojaus orus. Norint sukurti neuroninį tinklą ir jį apmokyti reikia duomenų. Kuo sudėtingesnis ir gilesnis neuroninis tinklas – tuo daugiau duomenų reikia, norint jį gerai apmokyti [RHW19]. Didelis duomenų kiekis modeliui nekenkia, bet dažnu atveju iškelia tokius iššūkius kaip padidėjęs neuroninio tinklo mokymo laikas, duomenų saugojimas bei optimalus mokymo būdas. Kita vertus, esant nepakankamam duomenų kiekiui iškyla persimokymo problema, nes modeliui paduodami tie patys duomenis daugelį kartų ir prarandamas gebėjimas generalizuoti.

Duomenų rinkinys gali būti ne tik didelis ar nepakankamas, bet ir nesubalansuotas. Nesubalansuotų duomenų aibės – problema, kuomet duomenų rinkinyje vienos ar kelių klasių duomenų yra žymiai mažiau nei kitos(ų), kitaip tariant – klasių duomenų aibės yra neproporcingos [YZ21]. Nesubalansuoti duomenys neleidžia tinkamai apmokyti modelių, nes dauguma mokymosi algoritmų gilesiems neuroniniams tinklams veikia geriausiai, kai duomenų kiekis rinkinyje yra panašus kiekvienai klasei.

Klasių disbalansas ypač pastebimas realaus pasaulio duomenų rinkiniuose. Pavyzdys galėtų būti vaizdinis piktybinio naviko identifikavimas medicinoje. Vaizdų su piktybiniu naviku bus žymiai mažiau negu vaizdų be jo. Tokiu atveju, nėra gaunami optimizuoti rezultatai klasei, kuri yra nesubalansuota (vaizdai su naviku), nes modelis niekada negauna pakankamai įžvalgos apie tiriamą klasę.

Tokia situacija negali būti priimtina daugumoje realaus pasaulio sferų (pvz., medicinoje ar įsilaužimo aptikimo srityje), todėl algoritmai, skirti kovoti su klasių disbalanso problema yra intensyviai tyrinėjami jau daugiau nei du dešimtmečius [FGG⁺18].

Nepaisant daugiau nei du dešimtmečius trunkančios pažangos, nesubalansuoti duomenys vis dar laikomi dideliu iššūkiu šiuolaikiniam mašininio mokymosi modeliams [DKC23]. Išsprendus šią problemą būtų gaunami žymiai geresni rezultatai klasifikavimo uždaviniuose, kurie yra taikomi įvairiose srityse – medicinoje, finansinėse paslaugose, kompiuterinėje regoje ir t.t.

Šiame darbe bus aptariami trys pagrindiniai duomenų lygmens nesubalansuotų vaizdinių duomenų problemos sprendimo būdai – atsitiktinis pavyzdžių perteklinis įtraukimas, augmentacijos bei DeepSmote. Sukurti dar keturi sprendimo būdai kombinuojant minėtuosius tris. Atliekami eksperimentai pasitelkiant skirtingus sprendimo būdus ir jų kombinacijas, duomenų rinkinius, modelius bei išbalansavimo santykius siekiant įvertinti sprendimo būdų efektyvumą.

1. Darbo tikslas, uždaviniai, siekiami rezultatai

Darbo tikslas – tirti mašininio mokymo nesubalansuotų vaizdinių duomenų aibių problemą naudojant skirtingus duomenų rinkinius, išbalansavimo santykius, modelius ir sprendimo būdus bei atliekant eksperimentus gauti rezultatus įvairioms eksperimento kombinacijoms.

Darbo uždaviniai:

- Išanalizuoti nesubalansuotų vaizdinių duomenų aibių problemą siekiant apžvelgti esamus problemos sprendimo būdus.
- Pasirinkti tris reprezentatyvius nesubalansuotų vaizdinių duomenų aibių problemos sprendimo būdus bei iš jų formuoti papildomas kombinacijas eksperimentams atlikti.
- Pasirinkti tris vaizdų duomenų rinkinius eksperimentams atlikti.
- Pasirinkti tris skirtingos architektūros mašininio mokymo modelius eksperimentams atlikti.
- Parašyti programinį kodą skirtą eksperimentų vykdymui ir rezultatų apdorojimui.
- Atlikti eksperimentus pasitelkiant skirtingus duomenų rinkinius, išbalansavimo santykius, modelius bei sprendimo būdus siekiant įvertinti sprendimo būdų efektyvumą ir gauti rezultatus.
- Apibendrinti gautus rezultatus ir pateikti išvalgas apie sprendimo būdų veiksmingumą.

Siekiami rezultatai – nustatyti, ar pasirinkti sprendimo būdai koreguojant nesubalansuotas vaizdinių duomenų aibes yra veiksmingi sprendžiant nesubalansuotų duomenų problemą ir pateikti gautus rezultatus bei analizuojant juos aprašyti išvadas.

2. Literatūros apžvalga

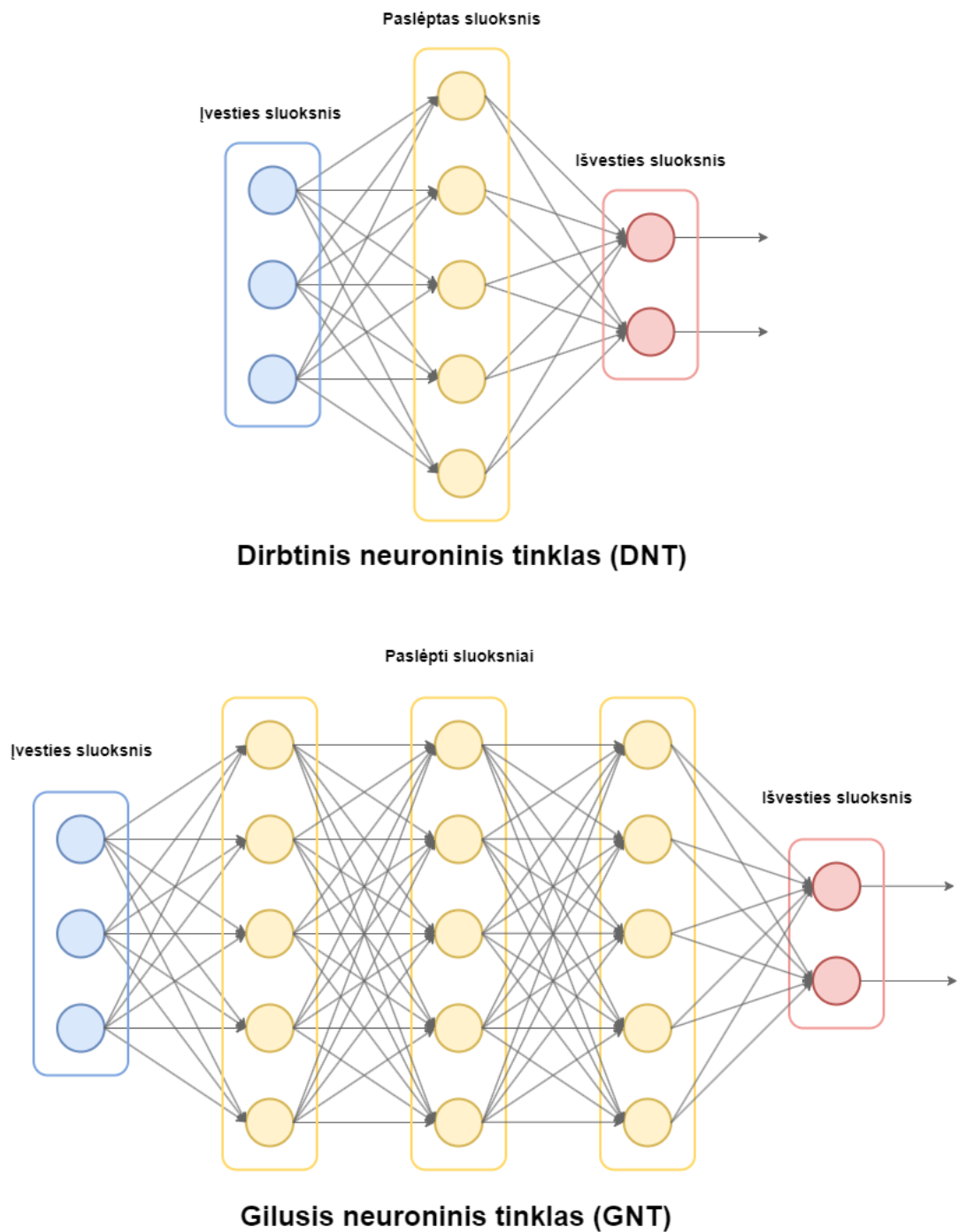
2.1. Dirbtiniai ir gilieji neuroniniai tinklai

Kai kurie iš pirmųjų šiandien žinomų mokymosi algoritmų buvo skirti biologinio mokymosi kompiuteriniams modeliams, t. y. modeliams, kurie atkartotų kaip mokymasis vyksta, arba galėtų vyksti, smegenyse. Nors neuroniniai tinklai, naudojami mašiniam mokymuisi, kartais buvo naudojami smegenų funkcijoms suprasti, paprastai jie nėra sukurti kaip tikroviški biologinių funkcijų modeliai. Terminas „gilusis mokymasis“ neapsiriboja tik neurologine dabartinės mašininio mokymosi modelių perspektyva. Jis remiasi į bendresnį kelių lygių sudėties mokymosi principą, kuris gali būti taikomas mašininio mokymosi sistemose, nebūtinai įkvėptose neurologijos.

Pagrindinis dirbtinių neuroninių tinklų principas yra toks pat: dirbtiniai neuroniniai tinklai mokosi keisdami ryšius tarp savo neuronų. Mokymasis vyksta, kai tinklas apdoroja tam tikras duomenų įvestis su žinomu ir norimu gauti rezultatu. Informacija kaupiama tinkle ir yra tikimasi, jog sutikus vis daugiau panašių duomenų, tinklas išmoks atpažinti tam tikrus duomenų ypatumus. Tokie tinklai gali atlikti daugybę informacijos apdorojimo užduočių [GBC16a].

Dirbtinio neuroninio tinklo (DNT) prižiūrimajam mokymui reikia duomenų rinkinio, kuriame būtų įvesties duomenys su anotacijomis, kuriuos naudojant tinklas turėtų išmokti tam tikras ypatybes. Dirbtiniai neuroniniai tinklai gali būti išmokyti labai tiksliai klasifikuoti tokius duomenis, koreguojant ryšio tarp jų neuronų stiprumą, ir gali išmokti generalizuoti rezultatą ir kitiems duomenų rinkiniams – su sąlyga, kad nauji duomenys per daug nesiskirs nuo mokymo duomenų. Puikus tokio tipo problemos pavyzdys yra objektų atpažinimas vaizduose, pavyzdžiui, vaizdų, padarytų savarankiškai važiuojančio automobilio, sekoje [Meh21].

Dirbtiniai neuroniniai tinklai ir gilieji neuroniniai tinklai (GNT) yra giminingos sąvokos, tačiau jos pirmiausia skiriasi savo gyliu ir sudėtingumu. Dirbtiniai neuroniniai tinklai yra palyginti negilūs ir naudojami paprastesnėms užduotims spręsti, o gilieji neuroniniai tinklai pasižymi gylumu ir naudojami sudėtingoms šablonų atpažinimo užduotims spręsti, kai svarbūs hierarchiniai požymiai.



1 pav. Dirbtinis ir gilusis neuroniniai tinklai

Gilusis mokymasis šiuo metu yra laikomas perspektyviausia mašininio mokymosi šaka, galinčia pasiekti išskirtinį pažinimo ir atpažinimo potencialą. Tačiau, nepaisant galingų galimybių, giliosios architektūros vis dar yra labai pažeidžiamos dėl nesubalansuotų duomenų pasiskirstymo [BDK⁺20] [BYM⁺20].

2.2. Nesubalansuoti duomenys

Duomenų rinkinys yra nesubalansuotas, jei klasės nėra apytikriai vienodai atstovaujamos [CBH⁺02].

Šiuolaikiniai gilaus mokymosi pasiekimai dar labiau paryškino nesubalansuotų duomenų problemos svarbą, ypač mokantis iš vaizdų [DKC23].

Pirmieji darbai, susiję su nesubalansuotais duomenimis, buvo atlikti tiriant dviejų klasių klasifikavimo problemas. Didžiausių ir mažiausių klasių buvimas yra nuspėjamas su tam tikru disbalanso santykiu. Tokie iškreipti klasių pasiskirstymai yra iššūkis mašininio mokymosi modeliams, nes standartiniai klasifikatoriai vadovaujasi 0-1 nuostolių funkcija, kuri numato vienodą baudą abiejoms klasėms. Todėl bet kokia mokymosi procedūra, kurią lemia tokia funkcija bus šališka didžiausios klasės atžvilgiu. Tuo pačiu metu, mažiausia klasė paprastai yra svarbesnė, todėl negali būti prastai atpažįstama. Dėl to metodai, skirti nesubalansuotų duomenų problemai įveikti, siekia arba sušvelninti klasių iškraipymą arba pakaitomis keisti mokymosi procedūrą [DKC23].

Nors duomenų gavybos metodai plačiai naudojami kuriant klasifikavimo modelius, kuriais vadovaujasi priimant komercinius ir valdymo sprendimus, nesubalansuotų duomenų klasifikavimas yra didelis iššūkis šiems tradiciniams klasifikavimo modeliams. Priežastys yra šios [HYS⁺17]:

- Standartiniai klasifikatoriai tinka subalansuotoms mokymo aibėms. Kai susiduriama su nesubalansuotais duomenimis, šie modeliai dažnai suteikia neoptimalius klasifikavimo rezultatus, t.y. gerai padengiama didžiausia klasė, o mažiausia - ne [LFG⁺13].
- Mokymosi procesas, grindžiamas globaliomis metrikomis, tokiomis kaip tikslumas, sukelia šališkumą didžiausios klasės atžvilgiu, o reti pavyzdžiai lieka nežinomi, net jei modelis užtikrina aukštą bendrą preciziją [LMC⁺16].
- Reti mažiausios klasės pavyzdžiai mokymosi modelyje gali būti traktuojami kaip triukšmas ir atvirkščiai - triukšmas gali būti klaidingai identifikuotas kaip mažiausios klasės pavyzdžiai, nes abu šie šablonai yra reti duomenų aibėje [BF15].
- Nors iškreiptus imčių pasiskirstymus ne visada sunku išmokti (pvz., kai klases galima atskirti), mažiausios klasės pavyzdžiai paprastai sutampa su kitais regionais, kuriuose abiejų klasių išankstinės tikimybės yra beveik lygios [DT10].
- Nesubalansuotam mokymuisi kyla iššūkių dėl mažų disjunktyvų [JJ04], nepakankamo tankio ir mažos imties su dideliu požymių dimensiškumu [WC10], todėl mokymosi modeliai dažnai nesugeba aptikti reti pasitaikančių šablonų [BTR16] [LFG⁺13].

Yra sukurta daug technikų, skirtų nesubalansuotų duomenų aibių problemai spręsti. Tokios technikos yra skirstomos į dvi dideles grupes: duomenų lygmens ir algoritmo lygmens [EYR12].

2.3. Duomenų lygmens metodai

Duomenų lygmens grupės yra tos, kuriomis bandoma subalansuoti mokymo duomenų rinkinius [EYR12]. Šis sprendimas turėtų būti laikomas pirminio apdorojimo etapu, kuris nepriklauso nuo klasifikatoriaus. Dėmesys sutelkiamas į duomenų rinkinio subalansavimą prieš pradedant mokymą.

Metodų grupės:

1. Mažinant didžiausios klasės dydį (angl. *undersampling*).
2. Didinant mažiausios klasės dydį (angl. *oversampling*).
3. Abiejų metodų kombinacija (hibridinis būdas).

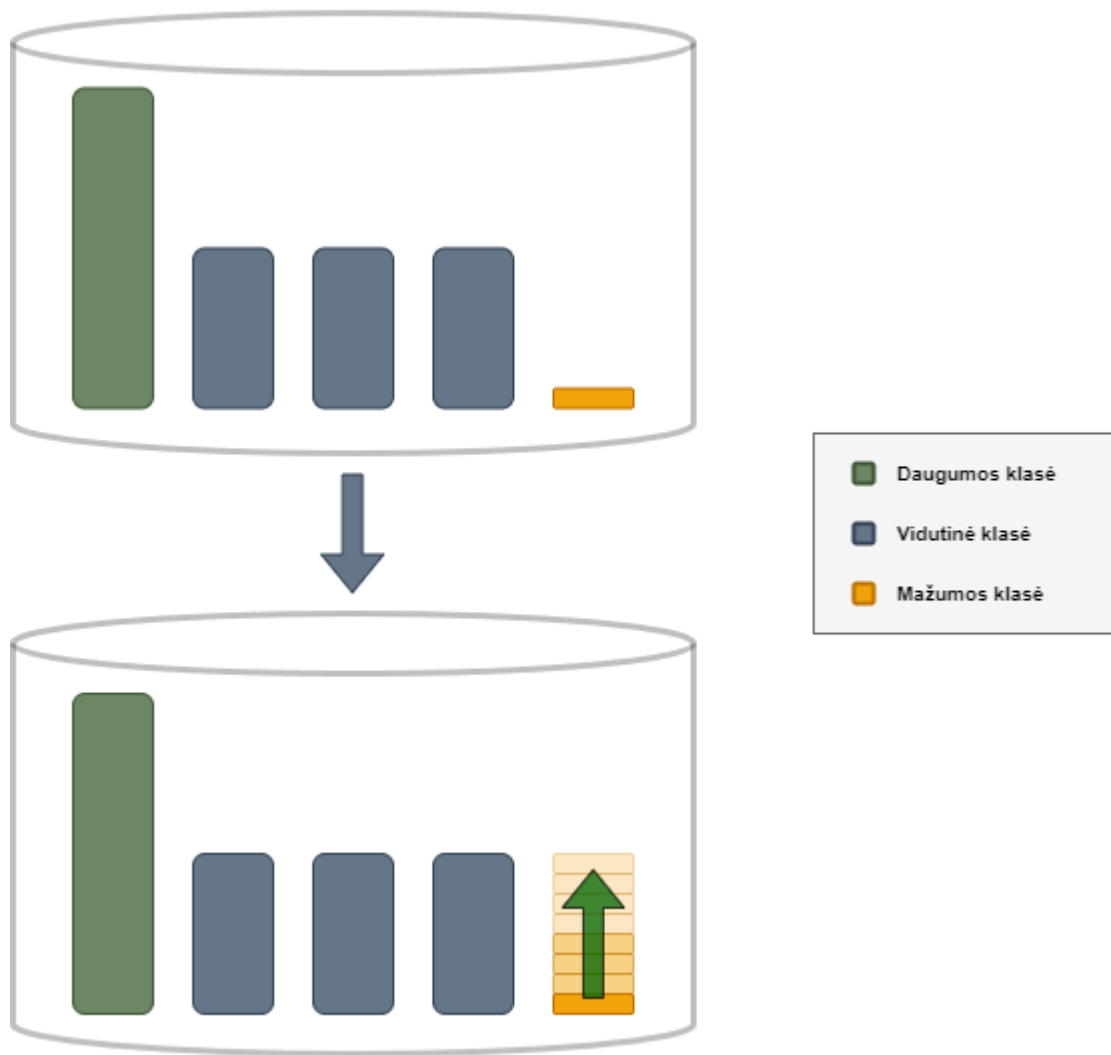
Imties retinimo metodų atveju sukuriamas originalių duomenų poaibis pašalinant tam tikrą kiekį pavyzdžių iš didžiausios klasės. Pavyzdžių perteklinio įtraukimo metodų atveju sukuriamas originalių duomenų viršaišis atkartojant kai kuriuos mažiausios klasės pavyzdžius arba sukuriant naujus pasinaudojant esamais mažiausios klasės duomenimis. Hibridinių metodų atveju kombinuojamas pavyzdžių įtraukimas bei retinimas.

Tiek imties retinimas, tiek perteklinis įtraukimas gali būti atliekami atsitiktiniu būdu, kas yra nesudėtinga, bet lemia potencialiai nestabilią elgseną (pvz. pašalinami svarbūs arba išryškinami triukšmingi duomenys). Todėl buvo pasiūlyti sprendimai, kuriais bandoma protingai atrinkti atvejus pirminiam apdorojimui. Nors nėra daug sprendimų, kurie yra pasiūlyti imties retinimui, daug daugiau dėmesio buvo skirta pavyzdžių pertekliniam įtraukimui dėl SMOTE sėkmės, dėl kurios buvo pasiūlyta gausybė variantų. Tačiau naujausi darbai rodo, kad SMOTE grindžiami metodai negali tinkamai susidoroti su daugiamodaliais duomenimis ir atvejais, kai yra didelis tarpklasinis persidengimas arba triukšmas. Dėl šios priežasties buvo sėkmingai sukurti visiškai nauji metodai, kurie nesiremia k-artimiausių kaimynų metodu [KKW19], [KW17].

Pakartotinės atrankos metodai dar vadinami „naivosios atrankos“ (angl. *naive resampling*) metodais, nes jie nieko nenumato apie duomenis ir nenaudoja jokios euristikos. Dėl to juos paprasta įgyvendinti ir greitai vykdyti, o tai pageidautina labai dideliems ir sudėtingiems duomenų rinkiniams. Svarbu tai, kad klasių pasiskirstymo pakeitimas taikomas tik mokymo duomenų rinkiniui. Taip siekiama paveikti modelių tinkamumą. Pakartotinė atranka netaikoma duomenų rinkiniams, kurie naudojami modelio veikimui įvertinti. Paprastai pakartotinės atrankos metodai gali būti veiksmingi, bet tai priklauso nuo duomenų rinkinio ir taikomų modelių specifikos [Bro21].

2.3.1. Pavyzdžių perteklinis įtraukimas

Mokymo aibės pavyzdžių perteklinis įtraukimas, kad duomenys skirtingose klasėse būtų tolygesni yra veiksmingas būdas išspręsti duomenų disbalanso problemą duomenų apdorojimo lygyje. Kita vertus, atsitiktinis pavyzdžių perteklinis įtraukimas lemia persimokymą, o tai mažina modelio klasifikavimo efektyvumą sutinkant naujus duomenis [LJL⁺20]. Toliau pateikta keletas pavyzdžių perteklinio įtraukimo metodų.



2 pav. Pavyzdžių perteklinis įtraukimas

2.3.1.1. Atsitiktinis perteklinis įtraukimas Atsitiktinis perteklinis įtraukimas priklauso pakartotinės atrankos metodams. Paprasčiausia strategija - atsitiktinai parinkti pavyzdžius transformuotam duomenų rinkiniui. Atsitiktinis pavyzdžių perteklinis įtraukimas apima atsitiktinai atrinktus mažiausios klasės pavyzdžius, juos pakeičiant, ir įtraukiant juos į mokymo duomenų rinkinį.

Pavyzdžiai iš mokymo duomenų aibės atrenkami atsitiktinai su pakeitimu. Tai reiškia, kad mažiausios klasės pavyzdžiai gali būti atrenkami ir įtraukiami į naują „labiau subalansuotą“ mokymo duomenų rinkinį kelis kartus. Jie atrenkami iš pradinio mokymo duomenų rinkinio, įtraukiami į naują mokymo duomenų rinkinį ir tada grąžinami arba „pakeičiami“ į pradinį duomenų rinkinį, kad juos būtų galima atrinkti dar kartą.

Šis metodas gali būti veiksmingas tiems mašininio mokymosi algoritams, kuriems įtakos turi iškreiptas pasiskirstymas ir kai daugybė pasikartojančių tam tikros klasės pavyzdžių gali turėti įtakos modelio tinkamumui. Tai gali būti algoritmai, kurie iteratyviai mokosi koeficientus, pavyzdžiui, dirbtinių neuronų tinklai, kuriuose naudojamas stochastinis gradientinis nusileidimas. Tai taip pat gali turėti įtakos modeliams, kurie ieško gerų duomenų skirstinių, pavyzdžiui, atraminių vektorių mašinoms ir sprendimų medžiams. Gali būti naudinga sureguliuoti tikslinės klasės pasi-

skirstymą. Tačiau, kadangi taikant šį metodą iš naujo sukuriama kai kurie esami pradinės mažiausios klasės pavyzdžiai, pagrindinis šio metodo trūkumas yra tas, kad jis gali sukelti persimokymą [Gan12]. Dėl šios priežasties mokymo duomenų rinkinio rezultatai gali pagerėti, tačiau validavimo arba testavimo duomenų rinkinio rezultatai atvirkščiai - pablogės [Bro21]. Taip pat, šiuo metodu yra padidinama mokymosi algoritmo skaičiavimo našta dėl išaugusio mokymo duomenų aibės dydžio.

Šis metodas vienas pavyzdys iš perteklinio įtraukimo technikos pagrindų. Daugelis kitų įprastų pavyzdžių perteklinio įtraukimo algoritmų, naudojamų realiose programose, yra sukurti remiantis šiuo metodu [JHC⁺15].

Nėra atlikta pakankamai darbų apie pakartotinę atranką naudojant būtent vaizdų duomenis.

2.3.1.2. Duomenų augmentacija Duomenų augmentacijos - tai naujų pavyzdžių perteklinis įtraukimas siekiant papildyti esamą duomenų rinkinį, transformuojant esamus pavyzdžius. Tai leidžia padidinti klasifikavimo tikslumą ir stabilumą. Klasifikatoriams sąveikaujant su įvairiomis mokymo imčių reprezentacijomis, modelis tampa mažiau šališkas, labiau nekintantis ir atsparus tokioms transformacijoms, kai bandoma generalizuoti modelį naujiems duomenų rinkiniams.

Duomenų augmentacijos pasirodė esančios veiksmingos daugelyje sričių, pavyzdžiui, vaizdų apdorojime ir objektų atpažinime. Naudojant standartinius augmentacijų metodus statistiškai reikšmingai pagerėja tikslumas ir F1 statistika [MMM23]. Netgi įrodyta, kad šis metodas gali padidinti labai gilių neuroninių tinklų tikslumą, palyginti su kitais standartiniais metodais, kurie apima įvairius reguliarizavimo metodus, ypač atsitiktinio praretinimo transformacija (angl. *Dropout*) [LLM20].

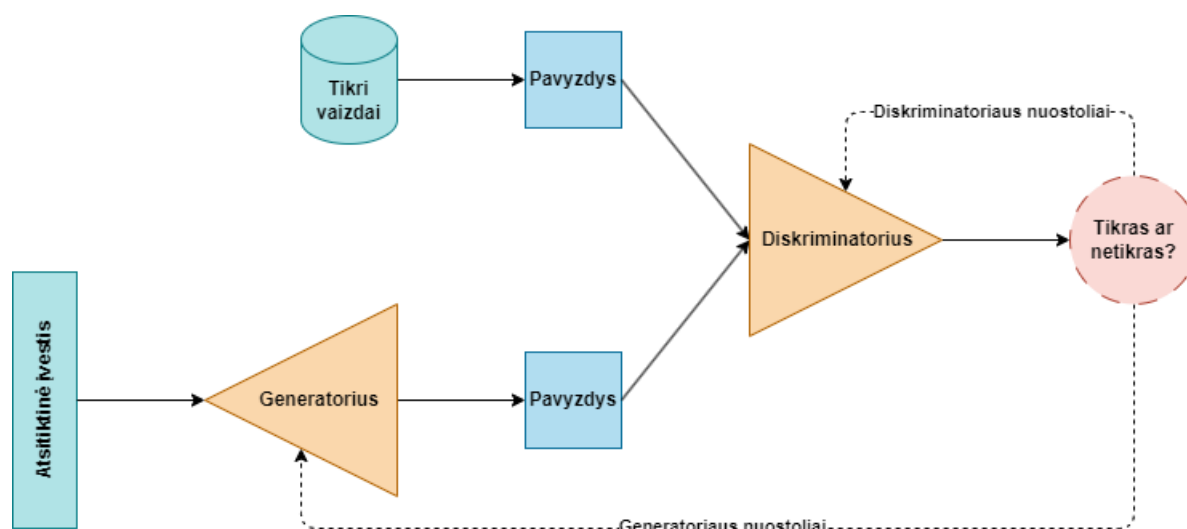
Nauji, augmentuoti duomenys paprastai gaunami dviem būdais. Pirmasis - taikant geometrines transformacijas: perkėlimą (angl. *translation*), sukimą (angl. *rotation*), apkarpymą (angl. *cropping*), apvertimą (angl. *flipping*), mastelio keitimą (angl. *scaling*) ir t. t. Antrasis būdas - esamus mokymo duomenis papildyti triukšmu. Didinant mokymo aibės dydį taip pat lengviau mokytis sudėtingesnius modelius su papildomais parametrais ir (arba) sumažina persimokymą. Tiesa, teisingai žymėti papildytus duomenų rinkinius gali būti sudėtinga [LLM20].



3 pav. Vaizdo augmentacijos pavyzdys

2.3.1.3. Generatyviniai adversariniai tinklai (GAN) Generatyviniai adversariniai tinklai (angl. *Generative Adversarial Networks (GAN)*) [GPM⁺14] yra architektūra, skirta generuoti pavyzdžius, mokant generatoriaus ir diskriminatoriaus modelius adversariniame procese, kai generatorius ir diskriminatorius gali būti bet koks tinklas. Tai generatyvinio modeliavimo metodas naudojant gilaus mokymosi metodus, pavyzdžiui, konvoliucinius neuroninius tinklus.

Generatyviniai adversariniai tinklai yra pagrįsti žaidimų teorijos scenarijumi, kuriame generuojantis tinklas turi varžytis su priešininku. Generatoriaus tinklas tiesiogiai gamina pavyzdžius. Jo priešininkas, diskriminatoriaus tinklas, bando atskirti pavyzdžius, gautus iš mokymo duomenų, nuo pavyzdžių, gautų iš generatoriaus [GBC16b].



4 pav. GAN architektūra

GAN yra įdomi ir sparčiai besikeičianti sritis, ypač sprendžiant vertimo iš vaizdo į vaizdą uždavinius, pavyzdžiui, verčiant vasaros nuotraukas į žiemos arba dienos į nakties nuotraukas, ir generuojant fotorealistines objektų, scenų ir žmonių nuotraukas, kurių nuo tikrų nuotraukų negali atskirti net žmonės [Bro19].

Kaip pavyzdys, GAMO (angl. *Generative Adversarial Minority Oversampling*) yra veiksmingas pavyzdžių perteklinio įtraukimo metodas, skirtas klasių disbalansui tvarkyti gilaus mokymosi sistemose. Be to, tai taip pat yra svarbus žingsnis siekiant mokyti patikimus diskriminacinius modelius naudojant priešišką mokymą (angl. *adversarial learning*) [MDD20].

2.3.1.4. SMOTE Bene plačiausiai naudojamas naujų pavyzdžių sintezės metodas SMOTE (angl. *Synthetic Minority Over-sampling Technique*) yra pavyzdžių perteklinio įtraukimo metodas, pagal kurį mažiausiai klasei kuriami „sintetiniai“ pavyzdžiai, o ne naudojamas perteklinis įtraukimas su pakeitimu. Toks metodas yra įkvėptas sėkmingos ranka rašytų simbolių atpažinimo technikos (nuotraukos yra pasukamos ar ištempiamos taip sukuriant naujus duomenis) [HB97]. Sintetiniai pavyzdžiai generuojami veikdami požymių erdvėje, o ne duomenų erdvėje (angl. *data space*).

SMOTE pirmiausia atsitiktinai pasirenka mažiausios klasės pavyzdį a ir suranda k artimiausių mažiausios klasės kaimynų. Tada sintetinis pavyzdys sukuriama atsitiktinai pasirenkant vieną iš k artimiausių kaimynų b ir sujungiant a ir b , kad požymių erdvėje būtų sudaryta linijos atkarpa. Sintetiniai pavyzdžiai sukuriama kaip dviejų pasirinktų atvejų a ir b išgaubtas derinys (angl. *convex combination*) [HC13].

Šią procedūrą galima taikyti norint sukurti tiek sintetinių mažiausios klasės pavyzdžių, kiek reikia. Pirmiausia siūloma naudoti atsitiktinį imties retinimą, kad būtų sumažintas didžiausios klasės pavyzdžių skaičius, o tada naudojant SMOTE generuoti mažiausios klasės pavydžius, kad būtų subalansuotas klasės pasiskirstymas. Šis metodas yra veiksmingas, nes sukuriama nauji sintetiniai mažiausios klasės pavyzdžiai, kurie yra tikėtini, t. y. požymių erdvėje santykinai artimi esamiems mažiausios klasės pavydžiams.

Bendras šio metodo trūkumas yra tas, kad sintetiniai pavyzdžiai kuriami neatsižvelgiant į didžiausią klasę, todėl, jei klasės labai sutampa, pavyzdžiai gali būti dviprasmiški.

Yra atlikta keletas bandymų išplėsti pavyzdžių perteklinio įtraukimo metodus, pavyzdžiui, SMOTE, ir pritaikyti juos gilaus mokymosi modeliams, tačiau rezultatai yra nevienareikšmiai [AH17] [FGH⁺18] [JK19].

Esant poreikiui turėti pavyzdžių perteklinio įtraukimo metodą, kuris būtų specialiai pritaikytas gilaus mokymosi modeliams, galėtų dirbti su neapdorotais vaizdais, išsaugant jų savybes, ir galėtų sukurti aukštos kokybės dirbtinius vaizdus, kurie galėtų sustiprinti rečiausias klases ir subalansuoti mokymo rinkinį buvo sukurtas „DeepSMOTE“ metodas.

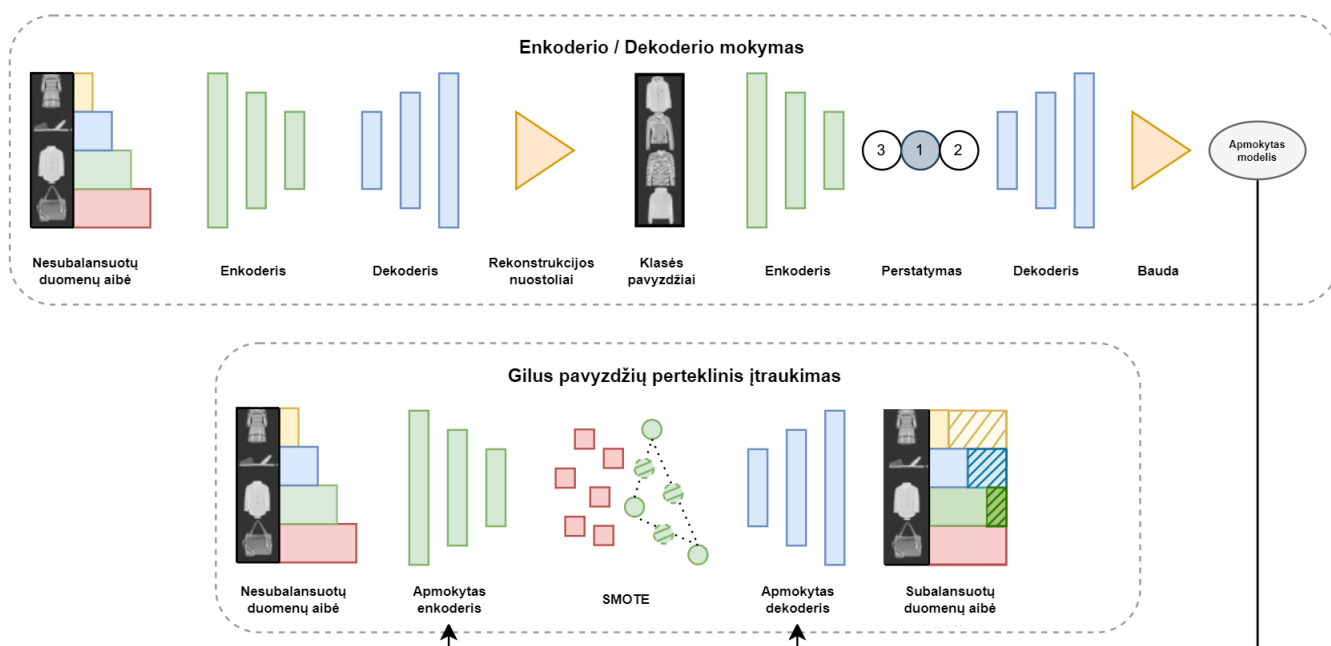
„DeepSMOTE: Fusing Deep Learning and SMOTE for Imbalanced Data“ darbe pažymima, kad norint sėkmingai taikyti pavyzdžių perteklinio įtraukimo metodą gilaus mokymosi modeliams, jis turėtų atitikti tris esminius kriterijus [DKC23]:

- Jis turėtų veikti nuo pradžios iki pabaigos (angl. *end-to-end*), priimdamas neapdorotus įvesties duomenis, pavyzdžiui, vaizdus (t. y. panašiai kaip VAE, WAE ir GAN).
- Jis turėtų išmokti atvaizduoti neapdorotus duomenis ir įterpti duomenis į mažesnės dimensijos požymių erdvę, kuri gali būti naudojama pavyzdžių pertekliniam įtraukimui.
- Jis turėtų lengvai generuoti išvestį (pvz., vaizdus), kurią būtų galima vizualiai patikrinti be didelių manipuliacijų.

Atliekant projektavimo etapus ir eksperimentinį vertinimą parodoma, kad „DeepSMOTE“ atitinka šiuos kriterijus.

„DeepSMOTE“ - naujas pavyzdžių perteklinio įtraukimo algoritmas, skirtas gilaus mokymosi modeliams. Jis yra paprastas, tačiau veiksmingas savo konstrukcija. Jį sudaro trys pagrindiniai komponentai:

- Kodavimo ir dekodavimo sistema
- SMOTE grindžiamas pavyzdžių perteklinis įtraukimas
- Speciali nuostolių funkcija, papildyta baudos nariu



5 pav. DeepSMOTE architektūra [DKC23]

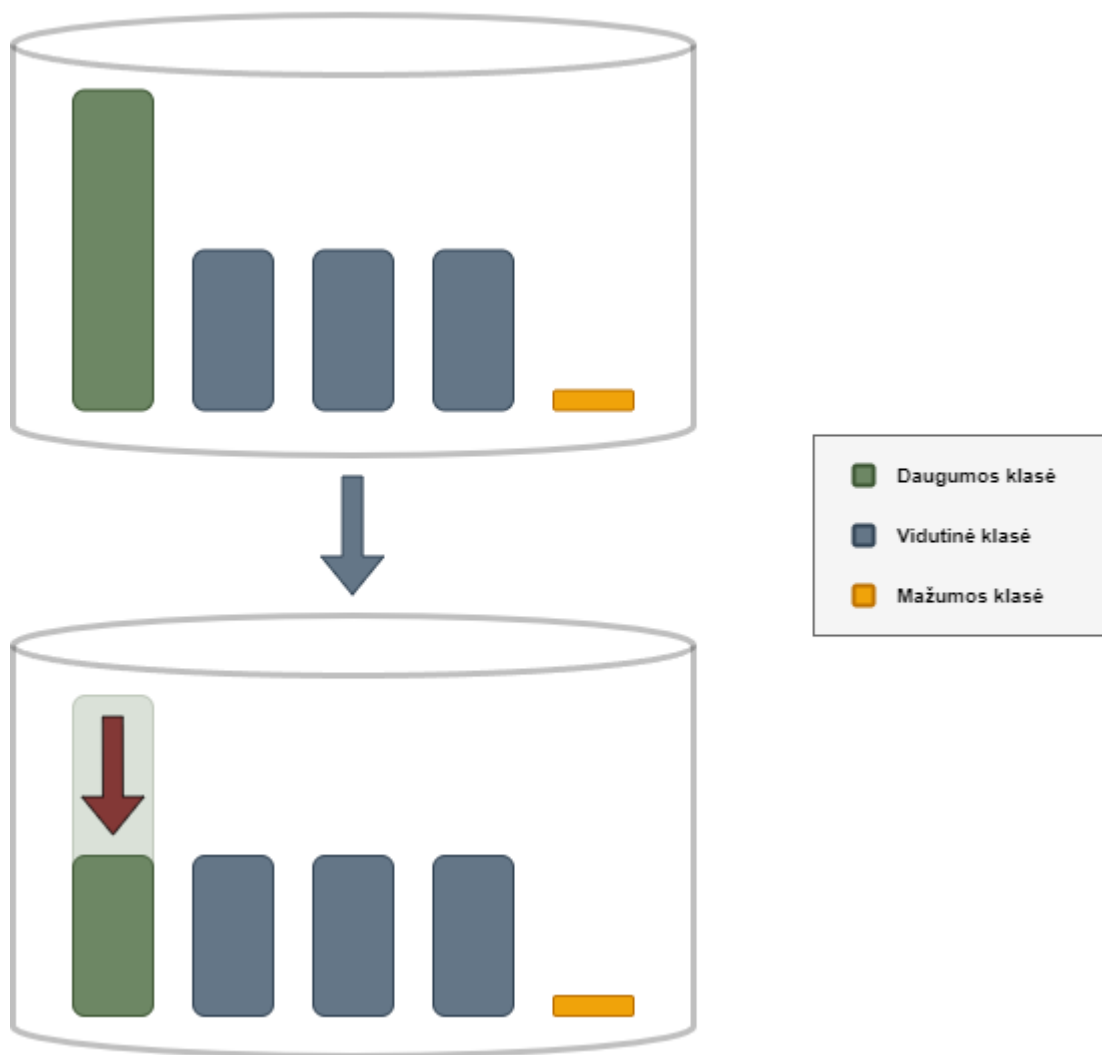
Svarbus „DeepSMOTE“ pranašumas prieš GAN pagrįstą pavyzdžių perteklinį įtraukimą yra tas, kad „DeepSMOTE“ metodui nereikia diskriminatoriaus. Taip pat šis metodas generuoja aukštos kokybės dirbtinius vaizdus, kuriuose gausu informacijos ir kurie yra tinkami vizualinei apžiūrai [DKC23].

2.3.2. Imties retinimas

Imties retinimas - didžiausiai klasei priklausančių pavyzdžių šalinimas, siekiant suvienodinti kiekvienos klasės pavyzdžių skaičių [FGG⁺18].

Imties retinimas daug duomenų turinčiai klasei yra gera priemonė padidinti klasifikatoriaus jautrumą mažai duomenų turinčiai klasei atpažinti.

Toliau pateikta keletas imties retinimo metodų.



6 pav. *Imties retinimas*

2.3.2.1. Atsitiktinis šalinimas Didžiausiai klasei taikomas atsitiktinis pavyzdžių šalinimas, kol mažiausia klasė tampa tam tikru nustatytu didžiausios klasės procentiniu dydžiu. Tai priverčia besimokantįjį modelį patirti įvairaus laipsnio pavyzdžių šalinimą ir esant didesniam pavyzdžių perteklinio įtraukimo laipsniui mažiausia klasė mokymo aibėje užima didesnę dalį [CBH⁺02]. Tai visiškai atsitiktinis procesas - nėra jokio konkretaus pasirinkimo, kuriuos pavyzdžius išsaugoti. Dėl šios priežasties, duomenų pasiskirstymas gali labai pasikeisti.

2.3.2.2. Klasteriais pagrįstas imties retinimas Klasteriais pagrįstas imties retinimas yra pažangus metodas, skirtas klasių disbalanso problemai spręsti, sumažinant didžiausios klasės pavyzdžių skaičių. Pagrindinis tikslas - išsaugoti didžiausios klasės įvairovę ir reprezentatyvumą, kartu veiksmingai sumažinant jos dydį, kad būtų subalansuotas klasės pasiskirstymas.

Viena iš klasteriais remto imties retinimo strategijų vykdoma taip (naudojamas k -vidurkių algoritmas) [LTH⁺17]:

1. k prilyginamas mažiausios klasės duomenų pavyzdžių skaičiui.

2. k klasterių centrai (arba centroidai) nustatomi k -vidurkių algoritmu pagal didžiausios klasės duomenų pavyzdžius.
3. Šie klasterių centrai naudojami visai didžiausios klasės duomenų aibei pakeisti.
4. Pakeitus didžiausios klasės duomenų aibę, tiek didžiausios tiek mažiausios klasės duomenų rinkinius sudaro tas pats duomenų pavyzdžių skaičius.

2.3.2.3. NearMiss NearMiss - tai imties retinimo technika, kuria didžiausios klasės pavyzdžių skaičius sumažinamas, jog būtų arčiau mažiausios klasės pavyzdžių skaičiaus. Tai palengvina klasifikavimą bet kuriuo algoritmu, naudojančiu erdvės atskyrimą arba dviejų klasių dimensinės erdvės padalijimą. Yra trys NearMiss versijos:

1. NearMiss-1: didžiausios klasės pavyzdžiai, kurių vidutinis atstumas iki trijų artimiausių mažiausios klasės pavyzdžių yra mažiausias.
2. NearMiss-2: didžiausios klasės pavyzdžiai su mažiausiu vidutiniu atstumu iki trijų tolimiausių mažiausios klasės pavyzdžių.
3. NearMiss-3: didžiausios klasės pavyzdžiai, kurių atstumas iki kiekvieno mažiausios klasės pavyzdžio yra mažiausias.

2.3.2.4. Tomek Links Tomek Links [Tom76] - tai arti viena kitos esančios priešingų klasių pavyzdžių poros. Pašalinus kiekvienos poros didžiausios klasės atvejus, padidėja erdvė tarp dviejų klasių ir palengvėja klasifikavimo procesas.

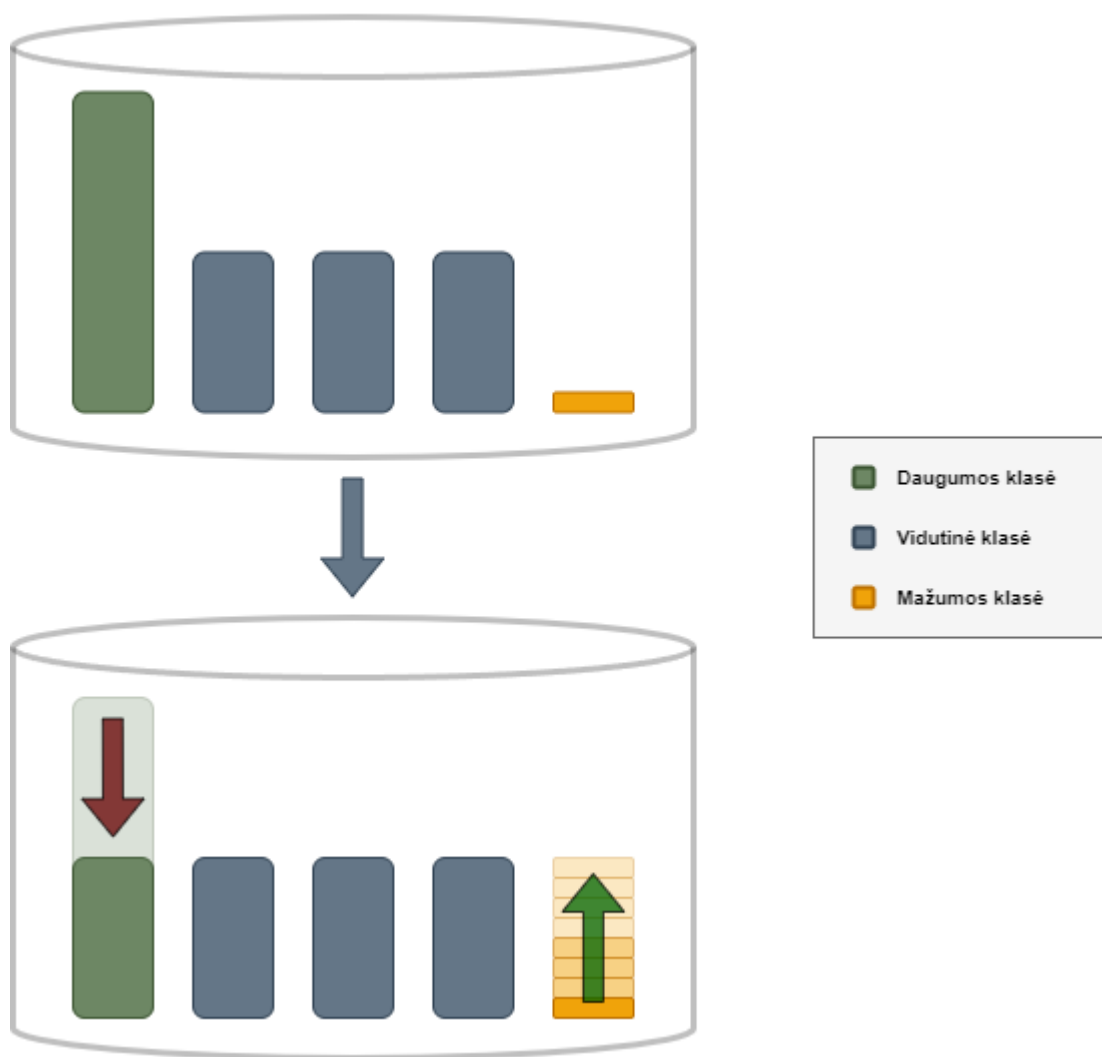
Pagrindinis Tomek Links naudojimo tikslas - pagerinti klasifikatorių veikimą, kad skiriamoji sritis tarp klasių būtų aiškesnė.

Tomek Links naudojimo trūkumas yra tas, kad pašalinus per daug atvejų iš didžiausios klasės, klasifikatorius gali būti nepakankamai gerai išmokytas ir todėl sumažėtų jo našumas.

2.3.2.5. Edited Nearest Neighbor (ENN) ENN [Wil72] yra imties retinimo technika, panašią į Tomek Links metodą. Tomek Links naudoja tik artimiausią kaimyną ryšiui apibrėžti ir tada jo atsisako, o ENN yra dinamiškesnis ir leidžia naudotojui nurodyti artimiausių kaimynų, į kuriuos reikia atsižvelgti, skaičių.

ENN gali prarasti daugiau informacijos nei Tomek Links, nes atmetami visi K artimiausi kaimynai, kurie yra artimi mažiausios klasės taškui, o ne tik vienas artimiausias kaimynas.

2.3.3. Hibridinis būdas



7 pav. Hibridinis būdas

Hibridinė technika kombinuoja tiek pavyzdžių perteklinį įtraukimą tiek imties retinimą. Tokiu būdu galima sulyginti visų klasių pavyzdžių kiekį mažinant didžiausios klasės pavyzdžių kiekį ir didinant mažiausios klasės duomenų kiekį. Vienas iš veiksmingų pavyzdžių yra SMOTE ir imties retinimo kombinacija, kuri duoda geresnius rezultatus negu tik imties retinimas [CBH⁺02].

2.3.4. Požymių atrankos ir požymių ištraukimo metodai

Esant nesubalansuotiems scenarijams, mažiausios klasės pavyzdžiai gali būti lengvai atmesti kaip triukšmas, tačiau jei iš požymių erdvės pašalinami nereikšmingi požymiai, ši rizika sumažėja [YHX⁺16].

Požymių atrankos tikslas - iš visos požymių erdvės atrinkti k požymių poaibį, kuris leistų klasifikatoriui pasiekti optimalų našumą, kai k yra naudotojo nustatytas arba adaptyviai parinktas parametras. Požymių atranka gali būti skirstoma į filtrus, apvalkalus ir įterptuosius metodus [HYS⁺17] [GE03].

Kitas būdas susidoroti su dimensiškumu yra požymių ištraukimas. Požymių ištraukimas yra susijęs su dimensiškumo mažinimu, kurio metu duomenys transformuojami į mažos dimensijos erdvę. Tačiau reikėtų pažymėti, kad požymių atrankos metodai skiriasi nuo požymių ištraukimo - požymių ištraukimas sukuria naujus požymius iš pradinių požymių, naudojant funkcinių atvaizdavimą, o požymių atranka grąžina pradinių požymių poaibį [ML02].

2.4. Algoritmų lygmens metodai

Priešingai nei anksčiau aptarti metodai, algoritmo lygmens sprendimai veikia tiesiogiai per nagrinėjamo klasifikatoriaus mokymo procedūrą. Dėl šios priežasties, jiems trūksta duomenų lygmens metodų teikiamo lankstumo, tačiau jie yra kompensuojami labiau tiesioginiu ir galingesniu būdu mažinant mokymosi algoritmo šališkumą. Be to, naudojant algoritmų lygmens metodus reikia nuodugniai suprasti, kaip vykdoma tam tikra mokymo procedūra ir kuri konkreiti jos dalis gali lemti didžiausios klasės šališkumą [DKC23].

Reikia pritaikyti metodą, kuris tiesiogiai spręstų disbalansą tarp klasių, pavyzdžiui pakeičiant klasių kaštus arba pakoreguojant tikimybės įvertinimą sprendimų medžio lapuose tam, kad tai pagerintų teigiamos klasės rezultatus [EYR12].

Daugelio klasių mokymasis laikomas sudėtinga užduotimi klasifikavimo algoritmams, nes daugelio klasių klasifikavimo rezultatai gali būti gerokai prastesni nei dviejų klasių klasifikavimo atveju. Ši problema tampa sudėtingesnė, kai susiduriama su nesubalansuotais duomenimis, nes ribos tarp klasių gali smarkiai persidengti [FLG⁺13].

2.4.1. Klasių svoriai

Naudojant nesubalansuotus duomenų rinkinius, klasifikatoriai linkę šališkai orientuotis į didžiausią klasę, todėl mažiausios klasės rezultatai dažnai būna prasti. Klasių svorių priskyrimas padeda neutralizuoti šį šališkumą, nes mokymo proceso metu mažiausiai klasei suteikiama daugiau reikšmės.

Klasių svoriai keičia mokymosi procesą, reguliuodami kiekvienos klasės svarbą nuostolių funkcijoje. Siekiama, kad už klaidingą mažiausios klasės klasifikavimą būtų baudžiama labiau nei už didžiausios klasės klasifikavimą. Tai skatina klasifikatorių daugiau dėmesio skirti mažiausiai klasei, todėl bendras rezultatas yra geresnis.

Paprasčiausias metodas - priskirti svorius, atvirkščiai proporcingus klasių dažniams. Jei n_i yra pavyzdžių skaičius klasėje i , N - bendras pavyzdžių skaičius, tada klasės i svorį galima apskaičiuoti taip:

$$Svoris = \frac{N}{n_i}$$

Taip užtikrinama, kad klasės, turinčios mažiau pavyzdžių, gautų didesnius svorius.

2.4.2. Kaštams jautrus mokymasis

Darant prielaidą, kad mažiausios klasės pavyzdžių klaidingo klasifikavimo kaina yra didesnė nei didžiausios klasės pavyzdžių, kaštams jautrų mokymąsi galima įtraukti tiek į duomenų (pvz.,

pakartotinė atranka ir požymių atranka), tiek į algoritmo lygmenį. Kainos dažnai nurodomos kaip kainų matricos, kur C_{ij} reiškia klaidingo klasifikavimo kainas, patiriamus priskiriant i klasei priklausančius pavyzdžius klasei j . Atsižvelgiant į konkrečią sritį, kainos matricos gali būti nustatomos remiantis ekspertų nuomone, o duomenų srauto scenarijuose jos gali skirtis kiekvienam įrašui arba kisti esant dinamiškai nesubalansuotai būklei [GMy13]. Lyginant su pakartotinės atrankos metodais, kaštams jautrus mokymasis skaičiavimo požiūriu yra efektyvesnis, todėl gali būti tinkamesnis dideliems duomenų srautams. Tačiau šis metodas yra daug mažiau populiarus nei pakartotinės atrankos metodai. Gali būti dvi galimos priežastys. Viena iš jų yra ta, kad kainų matricoje daugeliu atvejų yra sunku nustatyti vertes, kadangi neteisingo klasifikavimo kainos yra nežinomos iš duomenų ir ekspertas negali jos nurodyti [KWS14]. Kita priežastis, kuri gali būti labiau pagrįsta, yra ta, kad pakartotinė atranka yra įprastas praktinis pasirinkimas tiems tyrėjams, kurie nėra mašininio mokymosi ekspertai. Skirtingai nuo kaštams jautraus mokymosi, kuriam dažnai reikia modifikuoti mokymosi algoritmą, pakartotinės atrankos metodus daug lengviau tiesiogiai įgyvendinti tiek pavieniuose, tiek ansamblio modeliuose [HYS⁺17].

2.4.3. Ansamblio metodai

Dar vienas būdas yra klasifikatorių ansambliai, kurie tapo populiariu klasių disbalanso problemų sprendimo metodu. Ansambliu pagrįsti klasifikatoriai, dar vadinami kelių klasifikatorių sistemomis [KS13], yra žinomi kaip pagerinantys vieno klasifikatoriaus našumą, sujungiant kelis bazinius klasifikatorius, kurie yra pranašesni už kiekvieną nepriklausomą klasifikatorių [FLG⁺13].

2.5. Vertinimo metrikos

Teisingas išmokytojų modelių įvertinimas yra viena svarbiausių šablono atpažinimo problemų [FHM09].

Dauguma mašininio mokymosi algoritmų geriausiai veikia, kai kiekvienoje klasėje yra pakankamai vienodas duomenų kiekis. Taip yra todėl, nes dauguma algoritmų yra skirti maksimaliai padidinti vieną iš vertinime naudojamų metrikų – tikslumą (angl. *accuracy*) ir sumažinti paklaidos reikšmę.

Deja, bet naudojant pagrindines metrikas su nesubalansuotais duomenimis gali būti sukurti neoptimalūs klasifikavimo modeliai, o išvados gali būti klaidinančios, nes šie rodikliai nejautrūs iškreiptoms sritims [BTR15].

1 pavaizduota klasifikavimo lentelė.

1 lentelė. Klasifikavimo lentelė

	Klasė A	Klasė B
Progozė A	TP (True Positive)	FP (False Positive)
Progozė B	FN (False Negative)	TN (True Negative)

Išskirtos trys vertinimo metrikų grupės, naudojamos klasifikavime [Jap13]:

- Slenkstinių verčių metrikos (pvz., tikslumas ir F statistika),
- Rankingo metodai ir metrikos (pvz., ROC analizė ir AUC)
- Tikimybinės metrikos (pvz., kryžminė entropija)

2.5.1. Slenkstinių verčių metrikos

Tai yra metrikos, pagrįstos slenksčiu ir kokybinių klaidų supratimu. Šios priemonės naudojamos, kai norime, kad modelis kuo labiau sumažintų klaidų skaičių [FHM09].

Bene plačiausiai naudojama slenkstinė metrika yra klasifikavimo tikslumas. Tikslumo metrika skaičiuojama tikslų prognozių kiekį padalinus iš visų prognozių kiekio.

$$Acc = \frac{TP + TN}{TP + FP + FN + TN}$$

Tikslumo papildinys yra vadinamas paklaida.

Nesubalansuotų duomenų aibių atveju tikslumas akivaizdžiai netinka kaip geriausia matavimo metrika [YZ21] [CBH⁺02], nes ji matuoja visų teisingai nustatytų duomenų procentą ir neišskiria kiekvienos klasės atskirai.

2.5.1.1. Precizijos - atkūrimo metrikos Precizija nusako, koks procentas prognozių, jog tai buvo konkreti klasė, iš tikrųjų ir buvo prognozuotoji klasė.

$$Pr = \frac{TP}{TP + FP}$$

Atkūrimas nusako, koks procentas prognozių konkrečiai klasei buvo teisingas.

$$Re = \frac{TP}{TP + FN}$$

Kadangi tiek precizijos, tiek atkūrimo metrika nagrinėja vieną konkrečią klasę, todėl yra naudojamos nesubalansuotų duomenų aibių atveju, kai svarbu yra ne tik bendras visų klasių rezultatas, bet ir kiekvienos klasės atskirai.

Precizija ir atkūrimą galima sujungti į vieną metriką, kuria siekiama subalansuoti abi metrikas. Tokia, nesubalansuotų duomenų klasifikavime populiari metrika yra vadinama F statistikos reikšme.

$$F = \frac{2 * Pr * Re}{Pr + Re}$$

F_β statistika yra F statistikos abstrakcija, kurioje precizijos ir atkūrimo pusiausvyrą apskaičiuojant harmoninį vidurkį kontroliuoja koeficientas, vadinamas beta.

$$F_\beta = \frac{(1 + \beta^2) * Pr * Re}{\beta^2 * Pr + Re}$$

2.5.1.2. Jautrumo - specifiškumo metrikos Jautrumas apibendrina, kaip gerai buvo prognozuota teigiama klasė. Ji skaičiuojama taip pačiai kaip atkūrimas.

$$Sen = Re = \frac{TP}{TP + FN}$$

Specifiškumas yra jautrumo papildinys arba tikrasis neigiamas rodiklis, kuris apibendrina, kaip gerai buvo prognozuota neigiama klasė.

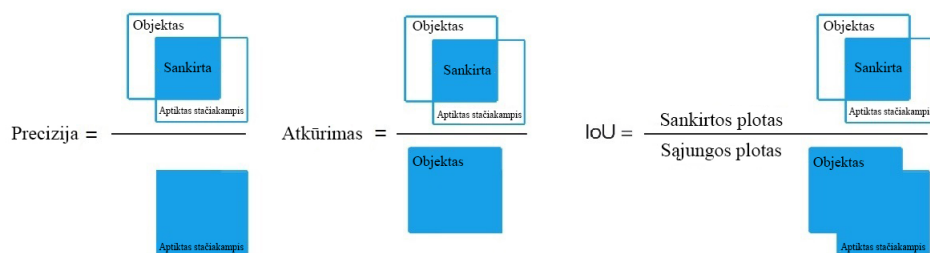
$$Spe = \frac{TN}{FP + TN}$$

Jautrumas ir specifiškumas gali būti sujungti į vieną statistiką, kuri suderina abu aspektus, vadinamą geometrinio vidurkiu arba G vidurkiu.

$$G = \sqrt{Sen * Spe}$$

2.5.1.3. Susikirtimas padalintas iš sąjungos (IoU) Kompiuterinės regos uždaviniuose tikimybė, jog prognozės regiono koordinatės atitiks etaloną (angl. *ground truth*) yra begalo maža ir nerealistiška. Dėl šios priežasties reikia naudoti metriką, kuri atvaizduotų prognozės ir tikrųjų reikšmių persidengimą. Tokia metrika yra vadinama IoU (angl. *intersection over union*, susikirtimas padalintas iš sąjungos), kuri yra populiariausia vertinimo metrika, naudojama objektų aptikimo uždaviniuose, ir yra išreiškiama kaip prognozės ir trokštamų reikšmių sankirtos ir sąjungos santykis. [RTG⁺19]

Geresniam įsivaizdavimui, precizijos, atkūrimo ir IoU metrikų vizualizacija, kompiuterinės regos uždaviniams spręsti, pavaizduota 8 paveikslėlyje ¹.



8 pav. Precizijos, atkūrimo ir IoU metrikų vizualizacija

Svarbus slenkstinių metrikų trūkumas yra tas, kad jos numato, jog visiškai žinomos sąlygos, kuriomis bus naudojamas klasifikatorius. Daroma prielaida, kad mokymo rinkinyje esantis klasių disbalansas yra toks pat, koks bus per visą klasifikatoriaus veikimo laiką [Jap13].

2.5.1.4. COCO metrikos COCO metrikos apjungia precizijos ir atkūrimo vertinimo metrikas. Jas sudaro 12 metrikų iš kurių 6 yra skirtos precizijai ir 6 atkūrimui įvertinti.

¹<https://github.com/AlexeyAB/darknet>

2 lentelė. COCO metrikos

Metrika	Aprašymas
Vidutinė precizija (AP)	
AP AP ^{IoU=0,50} AP ^{IoU=0,75}	AP esant IoU=0,50:0,05:0,95 (pagrindinė metrika) AP esant IoU=0,50 (tradicinė PASCAL VOC metrika) AP esant IoU=0,75 (griežta metrika)
AP skirtingose skalėse	
AP ^{small} AP ^{medium} AP ^{large}	AP mažiems objektams: plotas < 32 ² px AP vidutiniams objektams: 32 ² px < plotas < 96 ² px AP dideliems objektams: plotas > 96 ² px
Vidutinis atkūrimas (AR)	
AR ^{max=1} AR ^{max=10} AR ^{max=100}	AR su 1 aptikimas per vaizdą AR su 10 aptikimų per vaizdą AR su 100 aptikimų per vaizdą
AR skirtingose skalėse	
AR ^{small} AR ^{medium} AR ^{large}	AR mažiems objektams: plotas < 32 ² px AR vidutiniams objektams: 32 ² px < plotas < 96 ² px AR dideliems objektams: plotas > 96 ² px

AP ir AR yra kelių IoU reikšmių vidurkis. Yra naudojama 10 IoU slenksčių: .50:.05:.95, kas skiriasi nuo tradiciškai naudojamo vieno .50 IoU slenksčio. Vidurkio skaičiavimas iš keletos IoU reikšmių suteikia geresnę objektų lokalizaciją. AP čia yra visų kategorijų vidurkis. Tradiciškai tai žymima mAP (angl. *mean Average Precision*). Tas pat galioja ir atkūrimui, kur AR yra ir mAR (angl. *mean Average Recall*). COCO duomenų rinkinyje mažų objektų yra daugiau negu didelių, todėl yra tam specialiai išskirtos metrikos su small, medium ir large kategorijomis. Plotas yra pikselių skaičius segmentacijos kaukėje arba stačiakampyje. AR yra maksimalus atkūrimas duotam fiksuotam skaičiui aptikimų per vaizdą ir paėmus vidurkį per visas kategorijas ir IoU. Vertinimo metrikos objektų aptikimui ir segmentacijai visais atžvilgiais yra identiškos, išskyrus IoU skaičiavimą (kuris atliekamas atitinkamai stačiakampiems arba kaukėms).

2.5.2. Rankingo metodai ir metrikos

Rankingo metrikos labiau susiję su klasifikatorių vertinimu pagal tai, kaip veiksmingai jie atskiria klases. Rankingo metrikos yra pagrįstos tuo, kaip gerai modelis reitinguoja pavyzdžius. Jos svarbios daugeliui programų, kai klasifikatoriai naudojami atrinkti geriausius n duomenų rinkinio pavyzdžių arba kai labai svarbu gerai atskirti klases [FHM09].

Pagal šias metrikas reikalaujama, kad klasifikatorius numatytų statistikos vertę arba priklausymo klasei tikimybę. Pagal šią statistikos vertę galima taikyti įvairias ribas klasifikatorių veiksmingumui patikrinti. Tie modeliai, kurie išlaiko gerą rezultatą naudojant įvairius slenksčius, gerai atskiria klases ir bus geriau įvertinti.

Turint klasifikatorių, kuris duoda skaitinę statistinę vertę, kad pavyzdys būtų priskirtas teigiamai klasei, vietoj paprasto teigiamo arba neigiamo prognozavimo, statistinė vertė šiuo atveju suteikia tam tikrą detalumo lygį [FGG⁺18].

Dažniausiai naudojama rankingo metrika yra ROC (Receiver Operating Characteristic) kreivė arba ROC analizė.

2.5.2.1. ROC ROC apibendrina binarinių klasifikatorių analizės sritį pagal jų gebėjimą atskirti klases.

ROC kreivė - tai diagnostinis grafikas, skirtas modelio elgsenai apibendrinti, apskaičiuojant klaidingai teigiamų (angl. *False Positive Rate*) ir tikrų teigiamų (angl. *True Positive Rate*) rezultatų dažnį modelio prognozių rinkiniui pagal skirtingus slenskičius.

Tikrų teigiamų rezultatų dažnis yra skaičiuojamas taip pat kaip ir atkūrimas bei jautrumas:

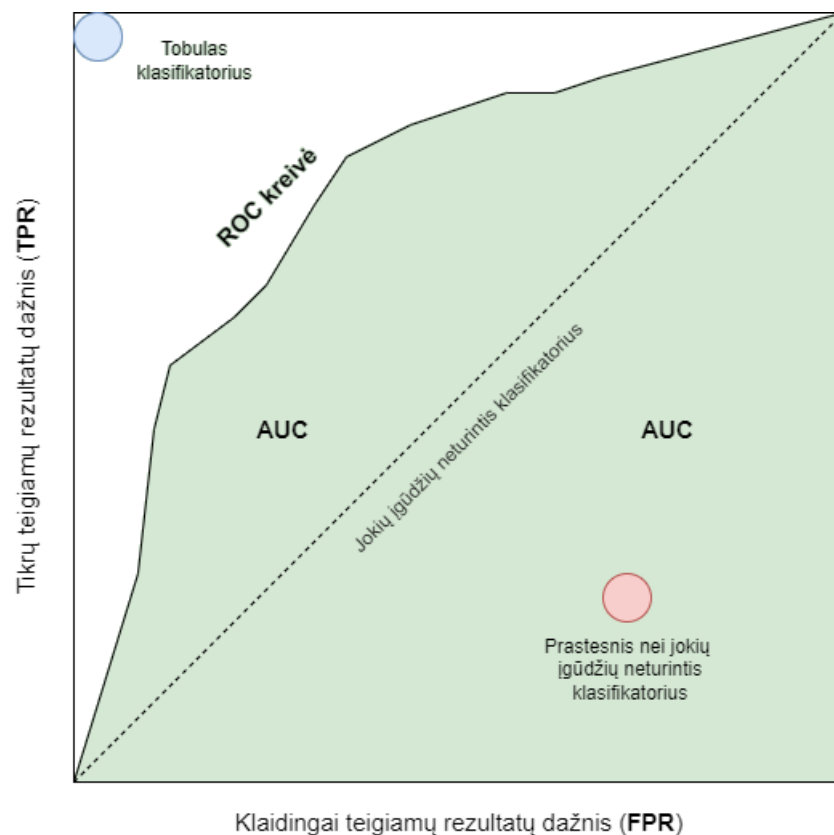
$$TPR = Sen = Re = \frac{TP}{TP + FN}$$

Klaidingai teigiamų rezultatų dažnis apskaičiuojamas taip:

$$FPR = \frac{FP}{FP + TN}$$

Kiekvienas slenkstis yra taškas grafike ir jie sujungiami į kreivę. Klasifikatorius, kuris neturi įgūdžių (pvz., prognozuoti dažniausią klasę su visomis riboms), bus pavaizduotas įstriža linija iš apatinio kairiojo kampo į viršutinį dešinįjį.

Bet kurie taškai, esantys žemiau šios linijos, turi prastesnius įgūdžius nei jokio įgūdžio. Tobulas modelis bus taškas viršutiniame kairiajame grafiko kampe.



9 pav. ROC kreivės grafikas

ROC kreivė yra naudinga diagnostika vienam modeliui.

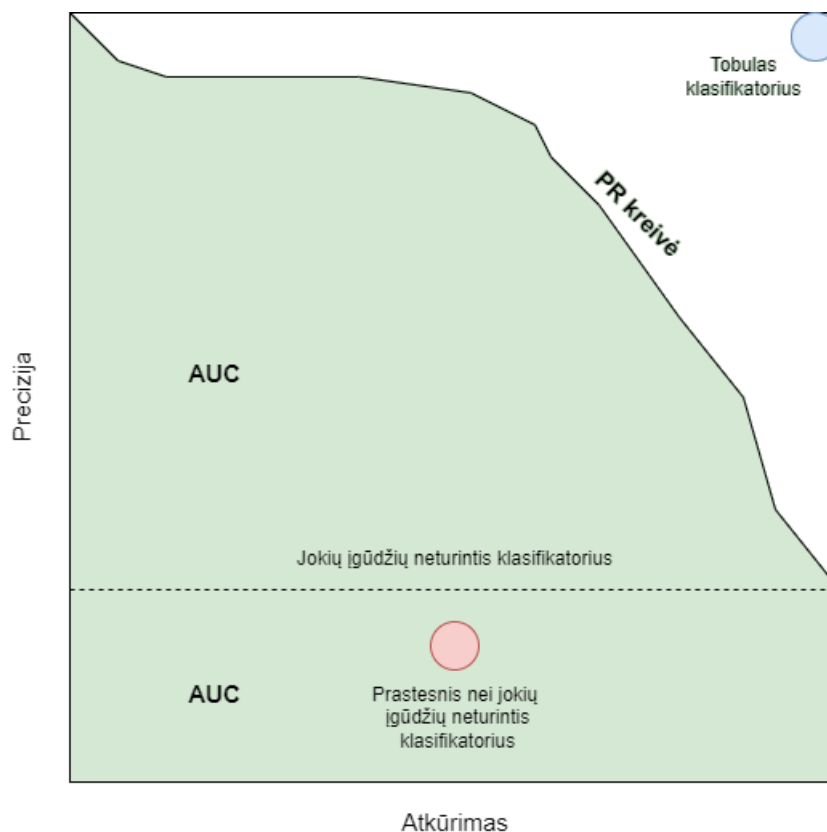
Galima apskaičiuoti plotą po ROC kreive ir gauti vieną statistinę vertę, apibendrinančią grafiką, kurią galima naudoti modeliams palyginti. Jokių įgūdžių neturinčio klasifikatoriaus rezultatas bus 0,5, o tobulo klasifikatoriaus - 1,0.

Nors paprastai veiksmingi, esant dideliame klasės disbalansui ROC kreivė ir ROC AUC (Area Under Curve) gali būti optimistiniai, ypač kai rečiausios klasės pavyzdžių skaičius yra mažas.

Alternatyva ROC kreivei yra precizijos - atkūrimo (PR - Precision-Recall) kreivė, kurią galima naudoti panašiai, tačiau joje daugiausia dėmesio skiriama klasifikatoriaus rezultatui mažiausiai klasei.

Vėlgi, modelio prognozių rinkiniui taikomi skirtingi slenksčiai, ir šiuo atveju apskaičiuojama precizija ir atkūrimas. Taškai sudaro kreivę, o klasifikatoriai, kurių rezultatai yra geresni esant skirtingiems slenksčiams bus geriau įvertinti.

Neįgudęs klasifikatorius bus horizontali linija diagramoje, kurios precizija bus proporcinga teigiamų pavyzdžių skaičiui duomenų rinkinyje. Subalansuotam duomenų rinkiniui statistinė vertė bus 0,5. Tobulas klasifikatorius reprezentuojamas tašku viršutiniame dešiniajame kampe.



10 pav. Precizijos - atkūrimo (PR) kreivės grafikas

Kaip ir ROC kreivė, precizijos - atkūrimo kreivė yra naudinga diagnostinė priemonė vienam klasifikatoriui vertinti, tačiau kelia iššūkių lyginant klasifikatorius.

Kaip ir ROC AUC, plotą po kreive galime apskaičiuoti kaip statistinę vertę ir naudoti ją klasifikatoriams palyginti. Šiuo atveju, dėl dėmesio mažiausiai klasei, precizijos - atkūrimo AUC yra naudingesnis būdas nesubalansuotų duomenų klasifikavimo problemoms spręsti.

Yra ir kitų rankingo metrikų, kurios naudojamos rečiau, pavyzdžiui, ROC kreivės modifikacija, skirta nesubalansuotam klasifikavimui ir kaštų kreivės.

2.5.3. Tikimybinių metrikos

Tikimybinių metrikos yra specialiai sukurtos klasifikatoriaus prognozių neapibrėžtumui kiekybiškai įvertinti. Tai naudinga sprendžiant uždavinius, kai mus mažiau domina neteisingų ir teisingų klasių prognozių santykis ir labiau domina modelio prognozių neapibrėžtumas ir baudimas už tas prognozes, kurios yra neteisingos, bet labai pasikliaujamos.

Šios metrikos yra pagrįstos tikimybinio paklaidos supratimu, t. y. nuokrypio nuo tikrosios tikimybės matavimu. Tai naudinga, kai norime įvertinti klasifikatorių patikimumą - ne tik įvertinti, kada jie nesėkmingai veikia, bet ir tai, ar su didele, ar su maža tikimybe jie pasirinko neteisingą klasę [FHM09].

Vertinant modelį pagal prognozuojamas tikimybes, reikia, kad tikimybės būtų kalibruotos. Kai kurie klasifikatoriai mokomi naudojant tikimybinių sistemą, pavyzdžiui, didžiausio tikėtinumo įvertinimą, t. y. jų tikimybės jau yra sukalibruotos. Kaip pavyzdys galėtų būti logistinė regresija. Daugelis netiesinių klasifikatorių nėra apmokyti pagal tikimybinių sistemą, todėl prieš juos vertinant tikimybine metrika, jų tikimybes reikia kalibruoti pagal duomenų rinkinį. Pavyzdžiai gali būti atraminių vektorių mašinos ir k-artimiausių kaimynų metodas.

Bene labiausiai paplitusi prognozuojamų tikimybių vertinimo metrika yra dvejetainio klasifikavimo logaritminis nuostolis (arba neigiamas logaritminis tikėtinumas), arba, geriau žinoma kaip kryžminė entropija bei kita populiari tikimybinių metrika - Brier statistikos vertė.

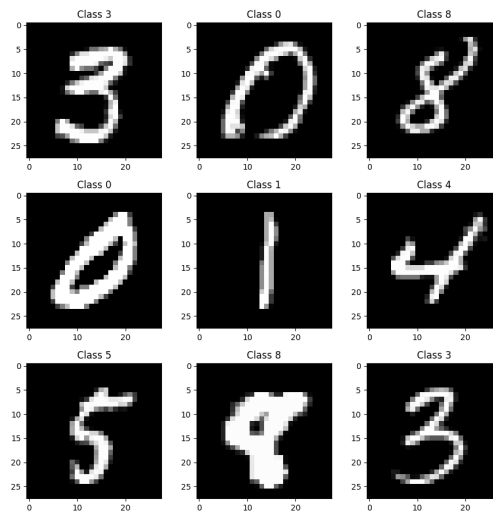
Nors ir populiarios sprendžiant subalansuotų duomenų klasifikavimo uždavinius, tikimybinių metrikos rečiau taikomos sprendžiant klasifikavimo uždavinius, kai klasių pasiskirstymas yra netolygus.

2.6. Uždaviniai

Kompiuterinės regos srityje išskiriamos trys pagrindinės užduotys: klasifikavimas, objektų aptikimas ir segmentavimas.

2.6.1. Klasifikavimo uždavinys

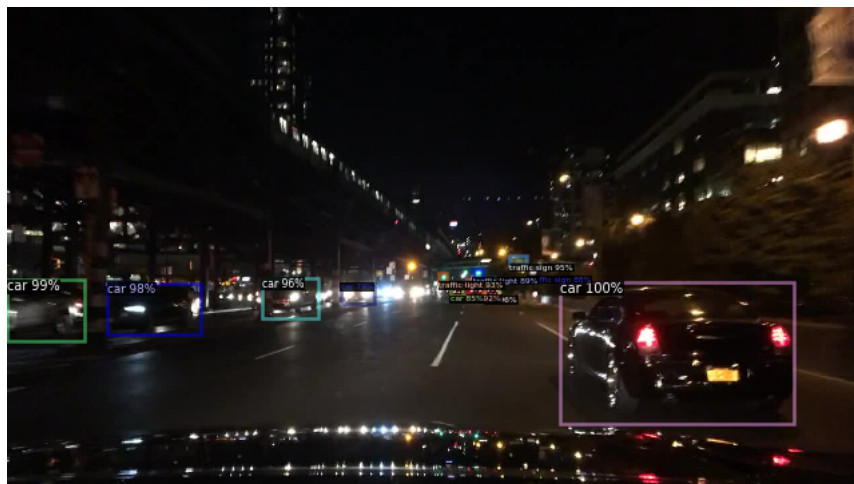
Vaizdų klasifikavimas yra bene pats paprasčiausias kompiuterinės regos uždavinys. Jis apima duomenų žymos priskyrimą arba klasės priskyrimą visam vaizdui pagal jo turinį. Pavyzdys pateikiamas 11 paveikslėlyje.



11 pav. Klasifikavimo pavyzdys

2.6.2. Objektų aptikimo uždavinys

Objektų aptikimas – technologija, leidžianti vaizde atpažinti konkrečios klasės objekto pavyzdžius. Ši kompiuterinėje regoje plačiai naudojama technologija pasitelkiama vaizdų anotavimui, transporto priemonių skaičiavimui, veido aptikimui bei atpažinimui ir kitiems uždaviniams, kuomet svarbu sužinoti objektų kiekį ir lokaciją. Pavyzdys pateikiamas 12 paveikslėlyje.



12 pav. Objektų aptikimo pavyzdys

2.6.3. Segmentacijos uždavinys

Segmentacija – procesas, kurio metu skaitmeninis vaizdas padalinamas į tam tikrus segmentus, išskiriami objektai ir jų ribos, vadinamosios kaukės. Tai yra objektų atpažinimo ir semantinės segmentacijos kombinacija, kuri sujungia elementus iš klasikinės kompiuterinės regos užduočių, susijusių su objektų aptikimu ir semantiniu segmentavimu (klasifikuojamas kiekvieną pikselis į fiksuotą kategorijų rinkinį neišskiriant objektų). Segmentavimas yra itin svarbus ir naudojamas įvairiose srityse, kaip pavyzdžiui, medicininių vaizdų analizėje (naviko ribų nustatymas, audinių

matavimas), autonominiame vairavime (pvz., važiuojamosios dalies nustatyme ir pėsčiųjų aptikime), vaizdo stebėjime, papildytoje realybėje ir dar daug kur kitur [HGD⁺18] [HB20] [MBP⁺20].

13 paveikslėlyje yra išskirtas objekto stačiakampis (lokalizacija), objekto klasė (klasifikavimas) ir objektą ribojantis poligonas (segmentacija).



13 pav. Vaizdų segmentacijos pavyzdys

3. Eksperimentas

Eksperimentui įvykdyti buvo reikalinga sukurti programinį kodą, kuris paruoštų duomenų rinkinius, modelius bei visą mokymosi bei vertinimo procesą.

Naudotas „Google Colab“ žiniatinklio IDE, skirtą Python kalbai.

Programinis kodas².

3.1. Duomenų rinkiniai

Nuspręsta eksperimentą atlikti su trimis duomenų rinkiniais: MNIST [LBB⁺98], Fashion MNIST [XRV17] (toliau FMNIST) ir SVHN (angl. *The Street View House Numbers*) .

Su duomenų rinkiniais buvo atliekamas klasifikavimo uždavinys. Kadangi duomenų rinkiniai nėra natūraliai nesubalansuoti, tai reikėjo atlikti dirbtiniu būdu.

3.1.1. MNIST

MNIST (angl. *Modified National Institute of Standards and Technology*)³ duomenų rinkinys sudarytas iš ranka rašytų skaitmenų 28x28 px dydžio vaizdų. Duomenų rinkinys turi 60 tūkst. apmokymo duomenų pavyzdžių ir 10 tūkst. testavimo duomenų pavyzdžių. Tai didesnio rinkinio, kurį galima gauti iš NIST, poaibis. Skaičiai yra normalizuoti pagal dydį ir centruoti fiksuoto dydžio paveikslėlyje.

²https://colab.research.google.com/drive/1byR6t7weH_kfX3CnG_7-GND08jBhH6gR?usp=sharing

³<http://yann.lecun.com/exdb/mnist/>

Žymė	Pavadinimas	Pavyzdžiai
0	Skaitmuo nulis	
1	Skaitmuo vienas	
2	Skaitmuo du	
3	Skaitmuo trys	
4	Skaitmuo keturi	
5	Skaitmuo penki	
6	Skaitmuo šeši	
7	Skaitmuo septyni	
8	Skaitmuo aštuoni	
9	Skaitmuo devyni	

14 pav. MNIST duomenų rinkinys

3.1.2. FMNIST

FMNIST⁴ remiasi „Zalando“ interneto svetainės mados produktų asortimentu. Kiekvienas pavyzdys yra 28x28 px pilkos skalės vaizdas, susietas su žyme iš 10 klasių. Siekiama, kad FMNIST būtų tiesioginis originalaus MNIST duomenų rinkinio pakaitalas mašininio mokymosi algoritmų lyginamajai analizei. Jis turi tą patį vaizdų dydį ir tą patį apmokymo ir testavimo rinkinių dydį bei santykį.

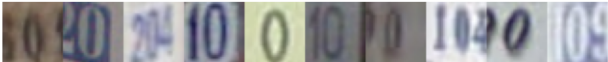
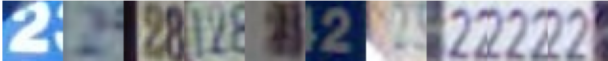
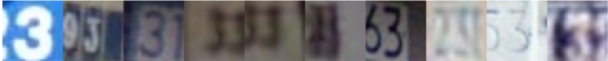
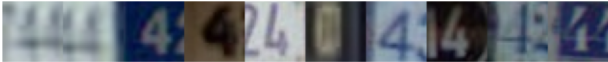
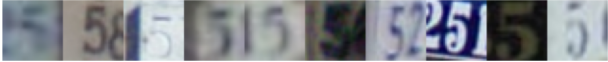
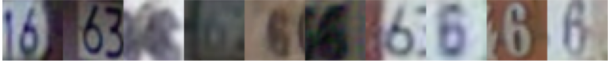
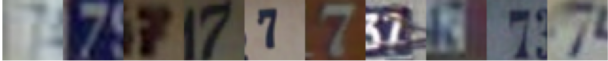
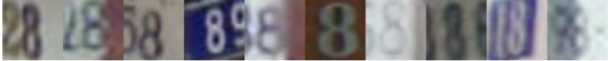
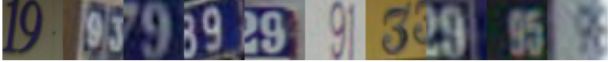
⁴<https://github.com/zalandoresearch/fashion-mnist>

Žymė	Pavadinimas	Pavyzdžiai
0	Marškinėliai	
1	Kelnės	
2	Megztinis	
3	Suknelė	
4	Paltas	
5	Sandalai	
6	Marškiniai	
7	Sportbačiai	
8	Krepšys	
9	Batai iki kulkšnies	

15 pav. FMNIST duomenų rinkinys

3.1.3. SVHN

SVHN yra realaus pasaulio vaizdų duomenų rinkinys, skirtas mašininio mokymosi ir objektų atpažinimo algoritmams kurti, o duomenų pirminiam apdorojimui ir formatavimui keliama minimalūs reikalavimai. Jis gali būti panašus į MNIST (pvz., vaizdai yra su mažais apkarpytais skaitmenimis), tačiau jame yra daugiau sužymėtų duomenų (daugiau kaip 600 tūkst. skaitmeninių vaizdų). Taip pat, SVHN duomenų rinkinys yra sudarytas iš gerokai sudėtingesnės, neišspręstos realaus pasaulio problemos (skaitmenys ir skaičių atpažinimas natūralios scenos vaizduose). SVHN gautas iš namų numerių „Google Street View“ vaizduose [NWC⁺11].

Žymė	Pavadinimas	Pavyzdžiai
0	Skaitmuo nulis	
1	Skaitmuo vienas	
2	Skaitmuo du	
3	Skaitmuo trys	
4	Skaitmuo keturi	
5	Skaitmuo penki	
6	Skaitmuo šeši	
7	Skaitmuo septyni	
8	Skaitmuo aštuoni	
9	Skaitmuo devyni	

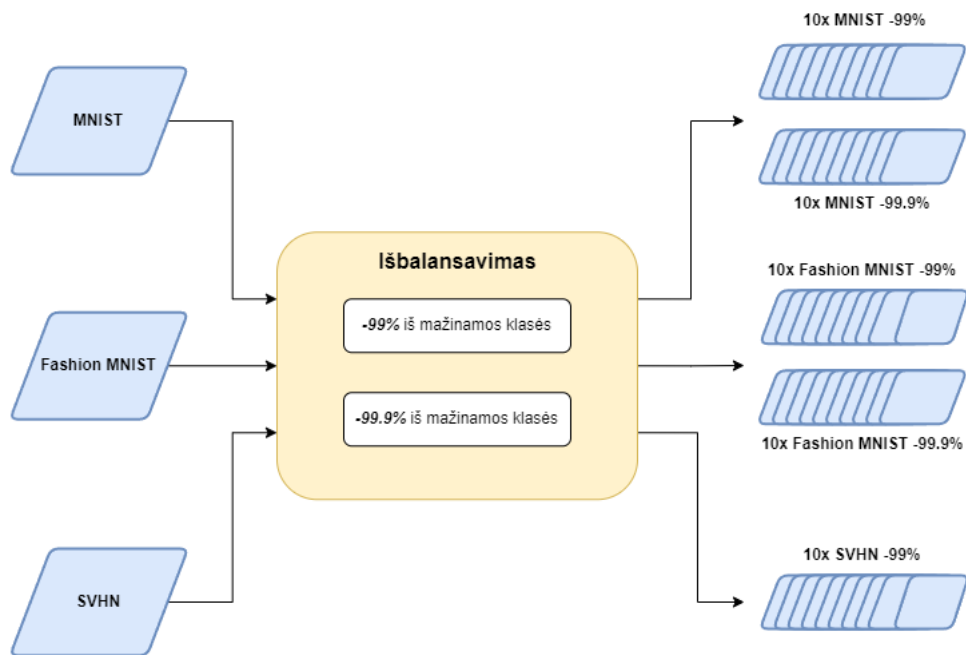
16 pav. SVHN duomenų rinkinys

3.2. Išbalansuotų duomenų rinkinių kūrimas

Pasikrovus duomenų rinkinius, buvo nuspręsta tiek MNIST tiek FMNIST duomenų rinkiniui mažinti vieną klasę iš jos pašalinant 99% ir 99,9% pavyzdžių. Bandymai su 50%, 75%, 90% neišbalansavo duomenų rinkinio tiek, kad tai svariai pablogintų mažinamos klasės rezultatus. SVHN duomenų rinkiniui pasirinktas 99% dydžio pavyzdžių šalinimas. Atsitiktiniu būdu pašalinus pasirinktos klasės pavyzdžius iš pasirinkto duomenų rinkinio gauta 50 duomenų rinkinių paruoštų eksperimentui:

- 10x MNIST -99%
- 10x MNIST -99,9%
- 10x FMNIST -99%
- 10x FMNIST -99,9%
- 10x SVHN -99%

Duomenų rinkinių koregavimo procesas pavaizduotas 17 paveikslėlyje.



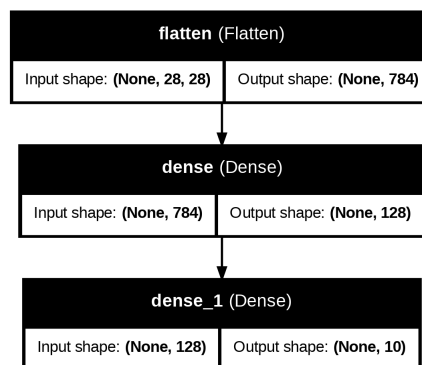
17 pav. Koreguotų duomenų rinkinių kūrimas

Šie duomenų rinkiniai buvo išsaugoti, kad būtų lengvai pasiekiami ir šalinimo nereiktų kartoti kartojant skaičiavimus.

3.3. Modeliai

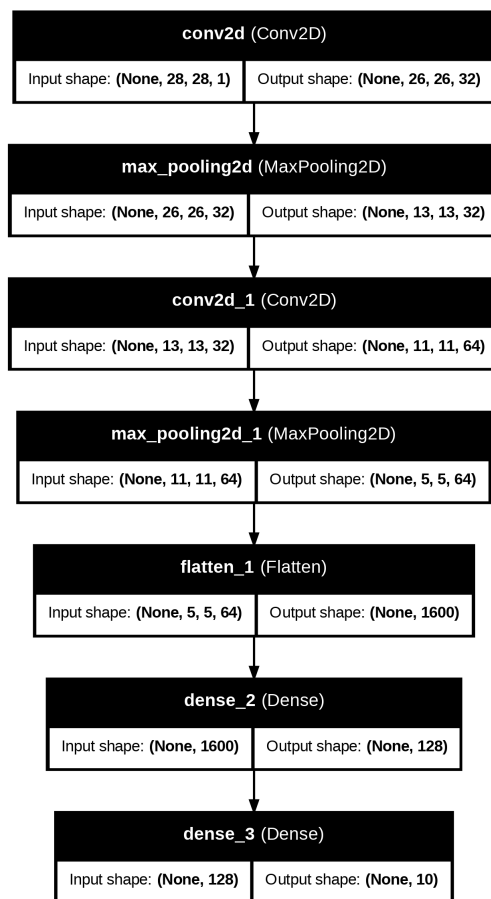
Sukurti 3 skirtingo gylio modeliai.

Pirmasis modelis turi 102 tūkst. (101770) parametrų. Jo architektūra pavaizduota 18 paveikslėlyje.



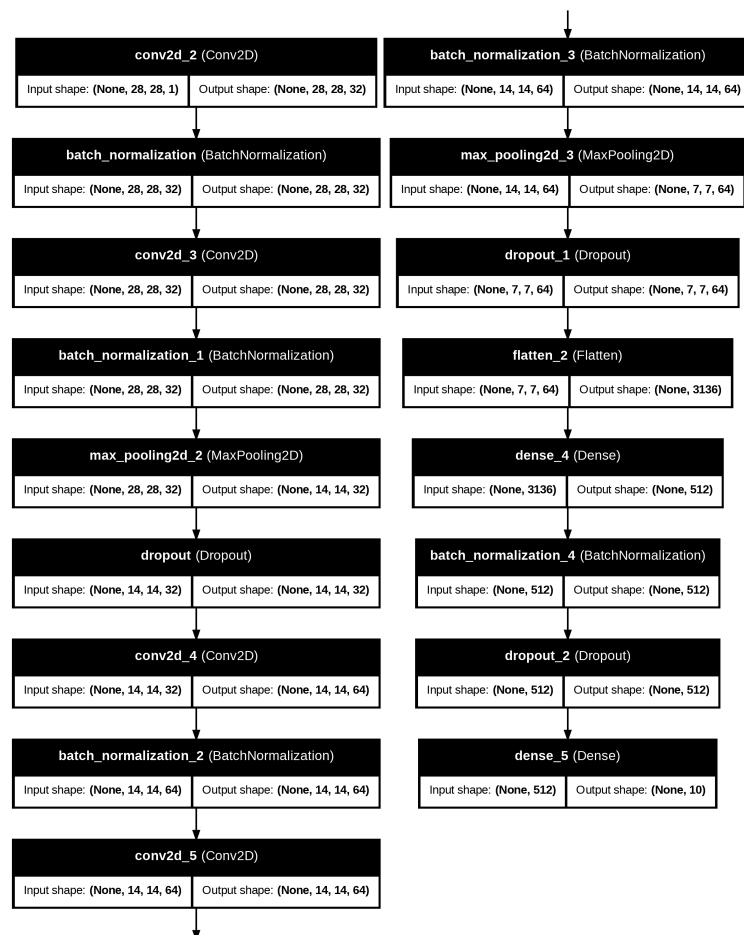
18 pav. Modelis nr. 1 (M1)

Antrasis modelis turi 225 tūkst. (225034) parametrų. Jo architektūra pavaizduota 19 paveikslėlyje.



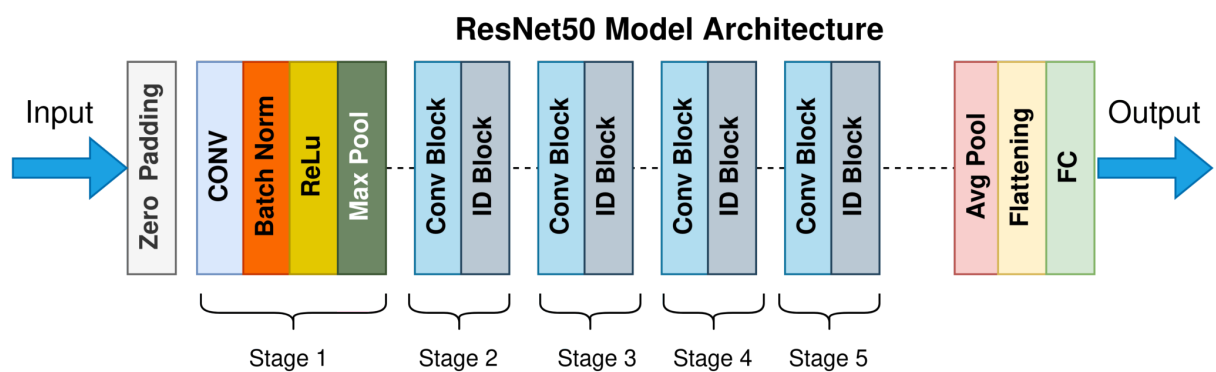
19 pav. Modelis nr. 2 (M2)

Trečiasis modelis turi 1,68 mln. (1679082) parametrų. Jo architektūra pavaizduota 20 paveikslėlyje.



20 pav. Modelis nr. 3 (M3)

Trečiasis modelis turi 25,7 mln. (25696138) parametrų. Jo architektūra yra ResNet50, kuri⁵ pavaizduota 21 paveikslėlyje⁶.



21 pav. Modelis nr. 4 (M4)

3.4. Metrikos

Naudojamos atkūrimo, vidutinės precizijos (AP), F1 ir ROC AUC metrikos.

⁵<https://keras.io/api/applications/resnet/#resnet50-function>

⁶<https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758/>

3.5. Pradiniai rezultatai

Duomenų rinkinio MNIST rezultatų pokytis su skirtingais modeliais po 99% duomenų pašalinimo iš mažinamos klasės atitinkamai pavaizduotas 12, 13 ir 14 lentelėse.

Duomenų rinkinio MNIST rezultatų pokytis su skirtingais modeliais po 99,9% duomenų pašalinimo iš mažinamos klasės atitinkamai pavaizduotas 15, 16 ir 17 lentelėse.

Duomenų rinkinio FMNIST rezultatų pokytis su skirtingais modeliais po 99% duomenų pašalinimo iš mažinamos klasės atitinkamai pavaizduotas 18, 19 ir 20 lentelėse.

Duomenų rinkinio FMNIST rezultatų pokytis su skirtingais modeliais po 99,9% duomenų pašalinimo iš mažinamos klasės atitinkamai pavaizduotas 21, 22 ir 23 lentelėse.

3.6. Sprendimo būdai

Pasirinkti trys skirtingi sprendimo būdai - DeepSmote, augmentacija ir atsitiktinis perteklinis įtraukimas.

3.6.1. DeepSmote

Nauji įrašai duomenų rinkinyje naudojant DeepSmote būdą buvo sukurti naudojant DeepSmote įgyvendinimą su Pytorch [PGM⁺19] ⁷. Pateikiamas kodas, skirtas modeliams apmokyti ir pavyzdžiams pagal apmokytą modelį generuoti.

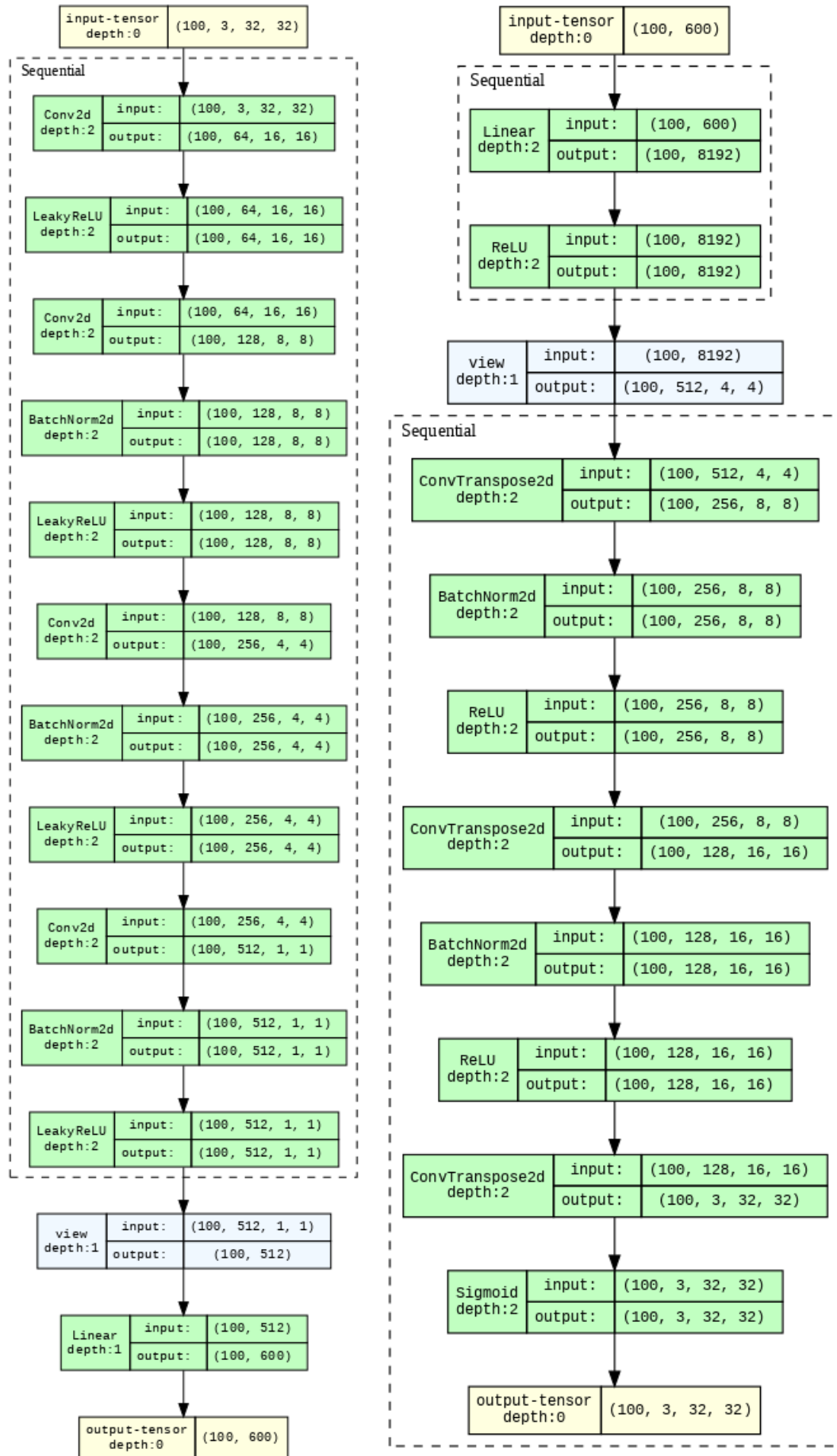
Sukurta 50 modelių:

- 10 modelių MNIST su viena 99% sumažinta klase
- 10 modelių MNIST su viena 99,9% sumažinta klase
- 10 modelių FMNIST su viena 99% sumažinta klase
- 10 modelių FMNIST su viena 99,9% sumažinta klase
- 10 modelių SVHN su viena 99% sumažinta klase

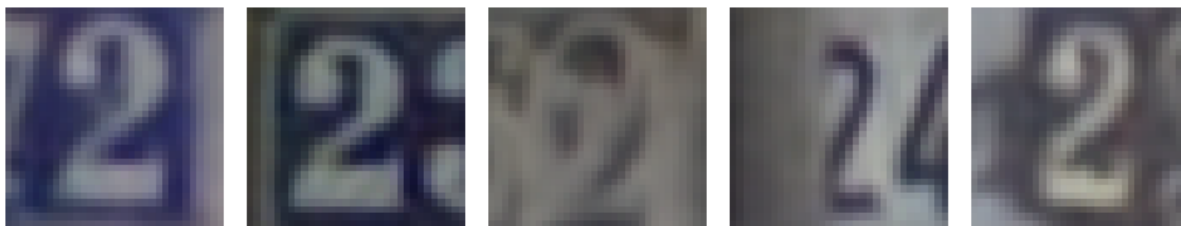
Sukūrus modelius, jie buvo naudojami naujų pavyzdžių generavimui.

Deepsmote enkoderio/dekoderio architektūra pavaizduota 22 paveikslėlyje.

⁷<https://github.com/dd1github/DeepSMOTE>



22 pav. DeepSmote enkoderis/dekoderis



23 pav. SVHN duomenų rinkinio pavyzdžiai sukurti naudojant DeepSmote metodą

3.6.2. Augmentacija

Nauji įrašai duomenų rinkinyje naudojant augmentacijos būdą sukurti naudojant *imgaug* [JWC⁺20] biblioteką skirtą vaizdų augmentacijai mašininio mokymosi eksperimentuose. Pasirinktos šios duomenų augmentacijos priemonės:

- GaussianBlur (suliejimas)
- LinearContrast (kontrasto pakeitimas)
- Multiply (šviesumo pakeitimas)
- Affine (afininė transformacija)
 - *scale*="x": (0.8, 1.2), "y": (0.8, 1.2),
 - *translate_percent*="x": (-0.2, 0.2), "y": (-0.2, 0.2),
 - *rotate*=(-25, 25),
 - *shear*=(-8, 8)



24 pav. MNIST duomenų rinkinio pavyzdžiai sukurti naudojant augmentaciją

3.6.3. Atsitiktins perteklinis įtraukimas

Nauji įrašai duomenų rinkinyje naudojant atsitiktinį perteklinį įtraukimą sukurti naudojant *imblearn* [LNA17] bibliotekos *RandomOverSampler* klasę.

3.6.4. Kombinuoti sprendimo būdai

Taip pat sukurtos keturios šių sprendimo būdų kombinacijos. Nekeičiant sprendimo būdų eiliškumo kombinuojant būdus gautas sąrašas iš 7 elementų:

1. DeepSmote (**D**)
2. Augmentacija (**A**)
3. Atsitiktinis perteklinis įtraukimas (**R**)
4. DeepSmote + Augmentacija (**D+A**)
5. DeepSmote + Atsitiktinis perteklinis įtraukimas (**D+R**)
6. Augmentacija + Atsitiktinis perteklinis įtraukimas (**A+R**)
7. DeepSmote + Augmentacija + Atsitiktinis perteklinis įtraukimas (**D+A+R**)

Kiekvienam iš būdų pasirinkti 4 skirtingi mažiausios klasės didinimo dydžiai - 25%, 50%, 75% ir 100% nuo didžiausios klasės. Tokiu būdu galima stebėti, kuris dydis duoda geriausių rezultatus palyginus su likusiais. Kombinuotiems sprendimo būdams, kiekvieno atskiro sprendimo būdo dydis yra lygus bendram didinamam procentui padalintam iš kombinuotame būde esančių atskirų būdų kiekiui (pvz. pasirinkus 50% didinimą ir *DeepSmote + Augmentacija + Atsitiktinis perteklinis įtraukimas* metodą, iš pradžių bus atliekamas DeepSmote metodas iki 16,66%, tuomet Augmentacija iki 33,33% ir Atsitiktinis perteklinis įtraukimas iki 50%).

3.7. Eksperimento rezultatai

Iš viso buvo atlikta 4200 skaičiavimų. MNIST ir FMNIST duomenų rinkiniams atlikta po 1680 skaičiavimų (2 (mažinimo dydžiai) * 10 (klasės) * 3 (modeliai) * 7(sprendimo būdai) * 4 (didinimo dydžiai)). SVHN duomenų rinkiniui atlikta 840 skaičiavimų (1 (mažinimo dydis) * 10 (klasės) * 3 (modeliai) * 7(sprendimo būdai) * 4 (didinimo dydžiai)).

SVHN duomenų rinkinio atveju pirmasis modelis, dėl savo paprastos architektūros, negebėjo tinkamai mokytis ir rezultatai nėra įtraukiami į tolimesnį darbą.

Rezultatai kiekvienai modelio bei klasės kombinacijai naudojantis atkūrimo metrika sprendimo būdų efektyvumui pamatuoti atvaizduojami grafikais:

- MNIST -99% duomenų rinkinių rezultatai: grafikas 25
- MNIST -99,9% duomenų rinkinių rezultatai: grafikas 30
- FMNIST -99% duomenų rinkinių rezultatai: grafikas 35
- FMNIST -99,9% duomenų rinkinių rezultatai: grafikas 40
- SVHN -99% duomenų rinkinių rezultatai: grafikas 45

Tai leidžia vizualiai palyginti rezultatus.

Taip pat pateikiami visų klasių rezultatai kiekvienai klasės bei sprendimo būdų kombinacijai naudojantis f1 metrika. Tai atvaizduota grafikais:

- MNIST -99% duomenų rinkinių rezultatai su skirtingais modeliais: 26, 27 ir 28.
- MNIST -99,9% duomenų rinkinių rezultatai su skirtingais modeliais: 31, 32 ir 33.
- FMNIST -99% duomenų rinkinių rezultatai su skirtingais modeliais: 36, 37 ir 38.
- FMNIST -99,9% duomenų rinkinių rezultatai su skirtingais modeliais: 41, 42 ir 43.
- SVHN -99% duomenų rinkinių rezultatai su skirtingais modeliais: 46 ir 47.

Visų sumažintų klasių rezultatų vidurkiai atvaizduoti 3, 4, 5, 6 ir 7 lentelėse. Šie rezultatai taip pat atvaizduoti ir grafikais - 29, 34, 39, 44, 49.

3 lentelė. MNIST -99% f1 rezultatų vidurkiai

Spalvų reikšmės: **žalia** – pagerėję rezultatai, **raudona** – suprastėję rezultatai, **mėlyna** – geriausias rezultatas vienai eksperimento kombinacijai (stulpelis), **paryškintas tekstas** – geriausias modelio rezultatas visame eksperimente. **P** - pradinių rezultatų vidurkis, **P-99%** - sumažintų klasių pradinių rezultatų vidurkis.

	M1	M1	M1	M1	M2	M2	M2	M2	M3	M3	M3	M3
Metodas	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%
P	0,9752	0,9752	0,9752	0,9752	0,9911	0,9911	0,9911	0,9911	0,9841	0,9841	0,9841	0,9841
P-99%	0,7672	0,7672	0,7672	0,7672	0,9008	0,9008	0,9008	0,9008	0,8414	0,8414	0,8414	0,8414
D	0,7779	0,7802	0,7659	0,7807	0,9493	0,9246	0,9389	0,9433	0,8831	0,8969	0,8700	0,8599
A	0,8082	0,7936	0,8105	0,8263	0,9189	0,9265	0,9191	0,9374	0,8832	0,7432	0,8631	0,8671
R	0,8234	0,8462	0,8346	0,8573	0,9294	0,9471	0,9397	0,9383	0,9647	0,9653	0,9708	0,9704
D+A	0,8447	0,8414	0,8683	0,8540	0,9367	0,9506	0,9461	0,9352	0,9217	0,7468	0,8648	0,8764
D+R	0,7875	0,8170	0,7985	0,8053	0,9280	0,9386	0,9505	0,9423	0,9060	0,9433	0,7963	0,9420
A+R	0,7739	0,8139	0,8420	0,8467	0,9298	0,9322	0,9464	0,9098	0,8920	0,9373	0,8559	0,9275
D+A+R	0,8442	0,8503	0,8545	0,8613	0,9515	0,9508	0,9446	0,9448	0,8702	0,7429	0,8795	0,9132

4 lentelė. MNIST -99,9% f1 rezultatų vidurkiai

Spalvų reikšmės: **žalia** – pagerėję rezultatai, **raudona** – suprastėję rezultatai, **mėlyna** – geriausias rezultatas vienai eksperimento kombinacijai (stulpelis), **paryškintas tekstas** – geriausias modelio rezultatas visame eksperimente. **P** - pradinių rezultatų vidurkis, **P-99%** - sumažintų klasių pradinių rezultatų vidurkis.

	M1	M1	M1	M1	M2	M2	M2	M2	M3	M3	M3	M3
Metodas	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%
P	0,9752	0,9752	0,9752	0,9752	0,9911	0,9911	0,9911	0,9911	0,9841	0,9841	0,9841	0,9841
P-99,9%	0,1800	0,1800	0,1800	0,1800	0,4802	0,4802	0,4802	0,4802	0,4608	0,4608	0,4608	0,4608
D	0,3561	0,4010	0,3875	0,3830	0,6781	0,6543	0,6661	0,7322	0,7536	0,8128	0,7487	0,8080
A	0,5662	0,5911	0,5879	0,6210	0,7210	0,8057	0,7938	0,8105	0,6218	0,6482	0,6425	0,6305
R	0,4092	0,3874	0,4222	0,3943	0,6224	0,6984	0,6857	0,7296	0,8322	0,7846	0,7993	0,8203
D+A	0,5098	0,6128	0,5713	0,6051	0,8062	0,7676	0,8128	0,8172	0,5334	0,5552	0,5485	0,5999
D+R	0,3813	0,3626	0,3470	0,3873	0,7141	0,7305	0,6990	0,7088	0,6927	0,8409	0,8345	0,8179
A+R	0,4868	0,5205	0,6156	0,5856	0,7385	0,7977	0,7685	0,8413	0,7592	0,6431	0,6451	0,5128
D+A+R	0,4896	0,5722	0,6329	0,6465	0,7604	0,8192	0,7842	0,8382	0,6708	0,7393	0,6301	0,5529

5 lentelė. FMNIST -99% f1 rezultatų vidurkiai

Spalvų reikšmės: **žalia** – pagerėję rezultatai, **raudona** – suprastėję rezultatai, **mėlyna** – geriausias rezultatas vienai eksperimento kombinacijai (stulpelis), **paryškintas tekstas** – geriausias modelio rezultatas visame eksperimente. **P** - pradinių rezultatų vidurkis, **P-99%** - sumažintų klasių pradinių rezultatų vidurkis.

	M1	M1	M1	M1	M2	M2	M2	M2	M3	M3	M3	M3
Metodas	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%
P	0,8678	0,8678	0,8678	0,8678	0,9054	0,9054	0,9054	0,9054	0,9161	0,9161	0,9161	0,9161
P-99%	0,3544	0,3544	0,3544	0,3544	0,4257	0,4257	0,4257	0,4257	0,5108	0,5108	0,5108	0,5108
D	0,6330	0,6307	0,6049	0,6306	0,5939	0,6256	0,5892	0,5904	0,7155	0,5498	0,5791	0,5404
A	0,4220	0,4753	0,4392	0,4763	0,5306	0,4660	0,5831	0,5066	0,5148	0,5056	0,5287	0,5497
R	0,7013	0,6598	0,6785	0,6969	0,6826	0,6660	0,6686	0,6590	0,7505	0,7573	0,7532	0,7653
D+A	0,6416	0,7088	0,7208	0,6611	0,6526	0,6345	0,6724	0,6044	0,5564	0,5969	0,5428	0,5385
D+R	0,6644	0,6590	0,6460	0,6519	0,6128	0,5915	0,6302	0,6698	0,6719	0,6182	0,6795	0,6131
A+R	0,5140	0,5253	0,4836	0,5308	0,6152	0,5765	0,5309	0,5269	0,6180	0,5617	0,5488	0,5752
D+A+R	0,6655	0,6552	0,6853	0,7118	0,6315	0,6270	0,6123	0,6091	0,6563	0,6936	0,6735	0,5294

6 lentelė. FMNIST -99,9% f1 rezultatų vidurkiai

Spalvų reikšmės: **žalia** – pagerėję rezultatai, **raudona** – suprastėję rezultatai, **mėlyna** – geriausias rezultatas vienai eksperimento kombinacijai (stulpelis), **paryškintas tekstas** – geriausias modelio rezultatas visame eksperimente. **P** - pradinių rezultatų vidurkis, **P-99%** - sumažintų klasių pradinių rezultatų vidurkis.

	M1	M1	M1	M1	M2	M2	M2	M2	M3	M3	M3	M3
Metodas	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%
P	0,8678	0,8678	0,8678	0,8678	0,9054	0,9054	0,9054	0,9054	0,9161	0,9161	0,9161	0,9161
P-99,9%	0,0701	0,0701	0,0701	0,0701	0,1045	0,1045	0,1045	0,1045	0,0737	0,0737	0,0737	0,0737
D	0,2022	0,2008	0,1872	0,2018	0,2968	0,2746	0,2357	0,2467	0,3225	0,3123	0,2575	0,3660
A	0,2491	0,2798	0,2610	0,2026	0,3309	0,3434	0,3999	0,3814	0,2839	0,3011	0,3058	0,3078
R	0,2202	0,2425	0,2464	0,1965	0,2814	0,2800	0,2813	0,2848	0,3679	0,4073	0,3790	0,3773
D+A	0,2722	0,2751	0,3590	0,3386	0,3428	0,3615	0,3497	0,3400	0,2850	0,3179	0,3128	0,2985
D+R	0,1818	0,2200	0,2348	0,2062	0,2744	0,2546	0,2664	0,2869	0,2857	0,3347	0,3273	0,3177
A+R	0,1510	0,2338	0,2646	0,2508	0,3294	0,3727	0,3665	0,3602	0,2559	0,2701	0,2884	0,3966
D+A+R	0,2896	0,2885	0,3155	0,3231	0,3156	0,3234	0,3463	0,3053	0,3831	0,2398	0,2747	0,2562

7 lentelė. SVHN -99% f1 rezultatų vidurkiai

Spalvų reikšmės: **žalia** – pagerėję rezultatai, **raudona** – suprastėję rezultatai, **mėlyna** – geriausias rezultatas vienai eksperimento kombinacijai (stulpelis), **paryškintas tekstas** – geriausias modelio rezultatas visame eksperimente. **P** - pradinių rezultatų vidurkis, **P-99%** - sumažintų klasių pradinių rezultatų vidurkis.

	M2	M2	M2	M2	M3	M3	M3	M3	M4	M4	M4	M4
Metodas	25%	50%	75%	100%	25%	50%	75%	100%	25%	50%	75%	100%
P	0,8690	0,8690	0,8690	0,8690	0,9235	0,9235	0,9235	0,9235	0,8923	0,8923	0,8923	0,8923
P-99%	0,3391	0,3391	0,3391	0,3391	0,4216	0,4216	0,4216	0,4216	0,3137	0,3137	0,3137	0,3137
D	0,4219	0,4360	0,4317	0,4203	0,4960	0,5041	0,5032	0,5190	0,5034	0,5392	0,5225	0,5375
A	0,4955	0,4597	0,4795	0,5210	0,4435	0,4332	0,4525	0,4503	0,5528	0,4967	0,5070	0,5107
R	0,4849	0,4982	0,4918	0,4615	0,5403	0,5250	0,5622	0,5218	0,5514	0,5170	0,5520	0,5638
D+A	0,5016	0,5037	0,4930	0,5361	0,4692	0,5055	0,5538	0,5559	0,5582	0,6031	0,5750	0,5458
D+R	0,4559	0,4347	0,3979	0,4235	0,5324	0,5070	0,5078	0,5155	0,5891	0,5431	0,5600	0,5706
A+R	0,4339	0,4464	0,4887	0,4710	0,3986	0,4206	0,4821	0,4772	0,5587	0,4895	0,5470	0,5036
D+A+R	0,4624	0,4689	0,4836	0,5028	0,4955	0,5555	0,5549	0,5387	0,4982	0,6225	0,6190	0,5775

8 lentelės rezultatai buvo skaičiuojami kiekvienai duomenų rinkinio, disbalanso dydžio ir metodo grupei pagal aritmetinį vidurkį iš trijų modelių (M1, M2 ir M3 arba M2, M3 ir M4) ir keturių mažiausios klasės didinimo dydžių (25%, 50%, 75%, 100%). Taigi, kiekvienos grupės F1 statistikos reikšmių vidurkis yra 12 verčių vidurkis (3 modeliai × 4 mažiausios klasės didinimo dydžiai).

8 lentelė. Vidutinės F1 statistikos reikšmės kiekvienam metodui skirtinguose duomenų rinkiniuose ir disbalanso lygmenyse

Spalvų reikšmės: **žalia** – pagerėję rezultatai, **mėlyna** – geriausias rezultatas duomenų rinkiniui (eilutė). **P** - pradinių rezultatų vidurkis, **P-disbalansas %** - sumažintų klasių pradinių rezultatų vidurkis.

Duom. rink.	P	P-disbalansas %	D	A	R	D+A	D+R	A+R	D+A+R	Geriausias
MNIST -99%	0,983	0,837	0,840	0,837	0,925	0,875	0,876	0,879	0,890	+0,088
MNIST -99,9%	0,983	0,374	0,615	0,670	0,632	0,645	0,626	0,660	0,678	+0,304
FMNIST -99%	0,896	0,430	0,607	0,500	0,703	0,628	0,642	0,551	0,646	+0,273
FMNIST -99,9%	0,896	0,083	0,259	0,304	0,297	0,321	0,266	0,295	0,305	+0,238
SVHN -99%	0,896	0,380	0,486	0,484	0,522	0,533	0,503	0,476	0,531	+0,153

3.8. Palyginimas su algoritmo lygmens metodu

Eksperimento metu taip pat buvo išbandytas ir algoritmų lygmens, svorių priskyrimo klasėms metodas. Tai buvo daroma, norint palyginti nagrinėtų duomenų lygmens metodų efektyvumą su algoritmo lygmens metodu.

Svoriai buvo apskaičiuoti naudojant scikit-learn bibliotekos [PVG⁺11] funkciją `compute_class_weight`⁸ su *balanced* strategija, kuri priskiria svorius atvirkščiai proporcingai klasių dažnims.

Kad būtų išvengta pernelyg didelės įtakos gradientams dėl mažiausių klasių, apskaičiuoti svoriai buvo normalizuoti taip, kad mažiausia klasė gautų maksimalų 10,0 svorį, o likusios klasės būtų proporcingai perskaičiuotos ir apribotos iki maksimalios 1,0 reikšmės. Toks subalansuotas svorių paskirstymas leido modeliams geriau atpažinti mažumos klases, nepažeidžiant mokymo stabilumo.

Klasių svorių metodo rezultatų palyginimas su geriausiais eksperimento rezultatais pateikiamas 9, 10, 11 lentelėse.

9 lentelė. MNIST duomenų rinkinio klasių svorių metodo rezultatų palyginimas su geriausiu eksperimento rezultatu

Spalvų reikšmės: **mėlyna** – geresnis rezultatas lyginant geriausių eksperimento metodą ir svorių metodą, **paryškintas tekstas** – geriausias rezultatas duomenų rinkiniui visame palyginime. **P** - pradinių rezultatų vidurkis, **P-99%** ir **P-99,9%**- sumažintų klasių pradinių rezultatų vidurkis.

Metodas / Modelis	M1	M2	M3
P	0,9752	0,9911	0,9841
P-99%	0,7672	0,9008	0,8414
Geriausias eksperimento metodas	0,8613	0,9515	0,9708
Svorių metodas	0,8105	0,9394	0,9502
P-99,9%	0,1800	0,4802	0,4608
Geriausias eksperimento metodas	0,6465	0,8413	0,8409
Svorių metodas	0,3262	0,6500	0,6697

⁸https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html

10 lentelė. FMNIST duomenų rinkinio klasių svorių metodo rezultatų palyginimas su geriausiu eksperimento rezultatu

Spalvų reikšmės: mėlyna – geresnis rezultatas lyginant geriausių eksperimento metodą ir svorių metodą, paryškintas tekstas – geriausias rezultatas duomenų rinkiniui visame palyginime. **P** - pradinio rezultatų vidurkis, **P-99%** ir **P-99,9%**- sumažintų klasių pradinio rezultatų vidurkis.

Metodas / Modelis	M1	M2	M3
P	0,8678	0,9054	0,9161
P-99%	0,3544	0,4257	0,5108
Geriausias eksperimento metodas	0,7208	0,6826	0,7653
Svorių metodas	0,6279	0,6749	0,6701
P-99,9%	0,0701	0,1045	0,0737
Geriausias eksperimento metodas	0,3386	0,3999	0,4073
Svorių metodas	0,2280	0,2567	0,2832

11 lentelė. SVHN duomenų rinkinio klasių svorių metodo rezultatų palyginimas su geriausiu eksperimento rezultatu

Spalvų reikšmės: mėlyna – geresnis rezultatas lyginant geriausių eksperimento metodą ir svorių metodą, paryškintas tekstas – geriausias rezultatas duomenų rinkiniui visame palyginime. **P** - pradinio rezultatų vidurkis, **P-99%** - sumažintų klasių pradinio rezultatų vidurkis.

Metodas / Modelis	M2	M3	M4
P	0,8690	0,9235	0,8923
P-99%	0,3391	0,4216	0,3137
Geriausias metodas	0,5316	0,5622	0,6225
Svorių metodas	0,5592	0,6079	0,6335

Pastebėta, kad nors svorių priskyrimo metodas kartais gali dar labiau pagerinti mažiausios klasės rezultatus, dažnai tai daroma kitų klasių rezultatų sąskaita. Eksperimentas su mažiausios klasės svorio pakeitimu iki pradinio dydžio (svoris 100,0) labai stipriai iškraipė rezultatus ir kai kuriais atvejais modelis negebėdavo mokytis. Dėl svorių priskyrimo padidėja modelio poliarizacija, o tai gali būti nepageidautina programose, kuriose reikia subalansuoto tikslumo arba teisingumo visose klasėse.

Rezultatai ir išvados

Rezultatai

- Darbo metu buvo tiriama mašininio mokymo nesubalansuotų vaizdinių duomenų aibių problema. Pasitelkti skirtingi duomenų rinkiniai, išbalansavimo santykiai, modeliai bei sprendimo būdai. Tai leido atlikti eksperimentus bei gauti rezultatus įvairioms duomenų rinkinių, išbalansavimo santykių, modelių bei sprendimo būdų kombinacijoms.
- Pasirinkti MNIST, FMNIST ir SVHN duomenų rinkiniai, 2 išbalansavimo santykiai, 4 modeliai bei 3 sprendimo būdai.
- Iš 3 sprendimo būdų gauti dar 4 kombinuoti sprendimo būdai.
- Sukurtas programinis kodas eksperimentui įvykdyti.
- Atliktas eksperimentas su 4200 skirtingų skaičiavimų.
- Gauti rezultatai pateikti lentelėse ir grafikuose eksperimento dalyje bei prieduose.

Išvados

- Pasirinkti sprendimo būdai **98,1%** (412 iš 420) atvejų padeda spręsti nesubalansuotų vaizdinių duomenų aibių problemą.
- Geriausius rezultatus duodantys sprendimo būdai ženkliai pagerina rezultatus (nuo 4,4 iki **46,65** procentinio punkto) lyginant su rezultatais nepritaikius sprendimo būdų.
- Pastebėta, jog naudojant DeepSmote sprendimo būdą ne kaip vienintelį, o pirmąjį kombinuotame sprendimo būde, jis duoda geresnius rezultatus ir laimi **60%** (36 iš 60) eksperimentų. Palyginimui nekombinuotas DeepSmote nelaimėjo nė vieno.
- Kombinuoti sprendimo būdai dažniau davė geresnius rezultatus ir laimėjo **65%** (39 iš 60) eksperimentų.
- Modeliai su gilesne architektūra ir daugiau parametrų duoda geresnius rezultatus sprendžiant nesubalansuotų vaizdinių duomenų aibių problemą.
- Siekiant išgauti geresnius rezultatus, rekomenduojama sprendimo būdus kombinuoti bei taikyti atsižvelgiant į konkretaus duomenų rinkinio ypatybes, pasirinkto modelio struktūrą ir duomenų išbalansavimo laipsnį.
- Rekomenduojama taikyti iteratyvų požiūrį – pritaikyti vieną sprendimo būdą, įvertinti jo veiksmingumą, koreguoti parametrus arba išbandyti kitus metodus bei juos kombinuoti, kol bus rastas tinkamiausias sprendimas konkrečiai situacijai.

Tyrimas gali būti tęsiamas pasitelkiant naujus sprendimo būdus, kurie nebuvo analizuojami šiame darbe. Galima išbandyti kitas modelių architektūras, išbalansavimo santykius ar duomenų rinkinius. Taip pat galėtų būti atliekama detalesnė rezultatų analizė. Esamų sprendimo būdų gerinimo pavyzdžiai galėtų būti:

- Kitokios augmentacijos būdo priemonės
- DeepSmote enkoderio/dekoderio architektūros keitimas
- Kitokie sprendimo būdų kombinavimo variantai

Kadangi geriausi rezultatai gauti kombinuojant kelis sprendimo būdus, padidinus sprendimo būdų kiekį būtų galima kombinuoti daugiau variantų ir galbūt taip pasiekti geresnius rezultatus.

Literatūra ir šaltiniai

- [AH17] S. Ando, C.-Y. Huang. *Deep Over-sampling Framework for Classifying Imbalanced Data*. 2017.
- [BDK⁺20] F. Bao, Y. Deng, Y. Kong, Z. Ren, J. Suo, Q. Dai. „Learning Deep Landmarks for Imbalanced Classification“. Iš: *IEEE Transactions on Neural Networks and Learning Systems* 31.8 (2020), puslapiai 2691–2704. <https://doi.org/10.1109/TNNLS.2019.2927647>.
- [BF15] C. Beyan, R. Fisher. „Classifying imbalanced data sets using similarity based hierarchical decomposition“. Iš: *Pattern Recognition* 48.5 (2015), puslapiai 1653–1672. issn: 0031-3203. <https://doi.org/https://doi.org/10.1016/j.patcog.2014.10.032>. URL: <https://www.sciencedirect.com/science/article/pii/S003132031400449X>.
- [BYM⁺20] L. A. Bugnon, C. Yones, D. H. Milone, G. Stegmayer. „Deep Neural Architectures for Highly Imbalanced Data in Bioinformatics“. Iš: *IEEE Transactions on Neural Networks and Learning Systems* 31.8 (2020), puslapiai 2857–2867. <https://doi.org/10.1109/TNNLS.2019.2914471>.
- [Bro19] J. Brownlee. *A Gentle Introduction to Generative Adversarial Networks (GANs)*. <https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/>. 2019.
- [Bro21] J. Brownlee. *Random Oversampling and Undersampling for Imbalanced Classification*. <https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/>. 2021.
- [BTR15] P. Branco, L. Torgo, R. Ribeiro. *A Survey of Predictive Modelling under Imbalanced Distributions*. 2015.
- [BTR16] P. Branco, L. Torgo, R. P. Ribeiro. „A Survey of Predictive Modeling on Imbalanced Domains“. Iš: *ACM Comput. Surv.* 49.2 (2016). issn: 0360-0300. <https://doi.org/10.1145/2907070>. URL: <https://doi.org/10.1145/2907070>.
- [CBH⁺02] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer. „SMOTE: Synthetic Minority Over-sampling Technique“. Iš: *Journal of Artificial Intelligence Research* 16 (2002), puslapiai 321–357. issn: 1076-9757. <https://doi.org/10.1613/jair.953>. URL: <http://dx.doi.org/10.1613/jair.953>.
- [DKC23] D. Dablain, B. Krawczyk, N. V. Chawla. „DeepSMOTE: Fusing Deep Learning and SMOTE for Imbalanced Data“. Iš: *IEEE Transactions on Neural Networks and Learning Systems* 34.9 (2023), puslapiai 6390–6404. <https://doi.org/10.1109/TNNLS.2021.3136503>.

- [DT10] M. Denil, T. Trappenberg. „Overlap versus Imbalance“. Iš: *Advances in Artificial Intelligence*. Sudarė A. Farzindar, V. Kešelj. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, puslapiai 220–231. ISBN: 978-3-642-13059-5.
- [EYR12] R. E., C. Y., B. R. „SMOTE-RSB *: a hybrid preprocessing approach based on over-sampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory“. Iš: *Knowl Inf Syst* (2012). <https://doi.org/https://doi.org/10.1007/s10115-011-0465-6>.
- [FGG⁺18] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, F. Herrera. „Learning from Imbalanced Data Sets“. Iš: *Cambridge International Law Journal*. 2018. URL: <https://api.semanticscholar.org/CorpusID:53046396>.
- [FGH⁺18] A. Fernández, S. Garcia, F. Herrera, N. Chawla. „SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary“. Iš: *Journal of Artificial Intelligence Research* 61 (2018), puslapiai 863–905. <https://doi.org/10.1613/jair.1.11192>.
- [FHM09] C. Ferri, J. Hernández-Orallo, R. Modroi. „An experimental comparison of performance measures for classification“. Iš: *Pattern Recognition Letters* 30.1 (2009), puslapiai 27–38. ISSN: 0167-8655. <https://doi.org/https://doi.org/10.1016/j.patrec.2008.08.010>. URL: <https://www.sciencedirect.com/science/article/pii/S0167865508002687>.
- [FLG⁺13] A. Fernández, V. López, M. Galar, M. J. del Jesus, F. Herrera. „Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches“. Iš: *Knowledge-Based Systems* 42 (2013), puslapiai 97–110. ISSN: 0950-7051. <https://doi.org/https://doi.org/10.1016/j.knosys.2013.01.018>. URL: <https://www.sciencedirect.com/science/article/pii/S0950705113000300>.
- [Gan12] V. Ganganwar. „An overview of classification algorithms for imbalanced datasets“. Iš: 2012. URL: <https://api.semanticscholar.org/CorpusID:7033031>.
- [GBC16a] I. Goodfellow, Y. Bengio, A. Courville. *Deep Learning*. <http://www.deeplearningbook.org> 13-14 p. MIT Press, 2016.
- [GBC16b] I. Goodfellow, Y. Bengio, A. Courville. *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press, 2016.
- [GE03] I. Guyon, A. Elisseeff. „An Introduction of Variable and Feature Selection“. Iš: *J. Machine Learning Research Special Issue on Variable and Feature Selection* 3 (2003), puslapiai 1157–1182. <https://doi.org/10.1162/153244303322753616>.
- [GMy13] A. Ghazikhani, R. Monsefi, H. sadoghi yazdi. „Online cost-sensitive neural network classifiers for non-stationary and imbalanced data streams“. Iš: *Neural Computing and Applications* 23 (2013), puslapiai 1283–1295. <https://doi.org/10.1007/s00521-012-1071-6>.

- [GPM⁺14] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio. *Generative Adversarial Networks*. 2014.
- [HB20] A. M. Hafiz, G. M. Bhat. „A survey on instance segmentation: state of the art“. Iš: *International Journal of Multimedia Information Retrieval* 9.3 (2020), puslapiai 171–189. ISSN: 2192-662X. <https://doi.org/10.1007/s13735-020-00195-x>. URL: <http://dx.doi.org/10.1007/s13735-020-00195-x>.
- [HB97] T. Ha, H. Bunke. „Off-line, handwritten numeral recognition by perturbation method“. Iš: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.5 (1997), puslapiai 535–539. <https://doi.org/10.1109/34.589216>.
- [HC13] T. R. Hoens, N. V. Chawla. „Imbalanced Datasets: From Sampling to Classifiers“. Iš: *Imbalanced Learning*. John Wiley Sons, Ltd, 2013. Skyrius 3, puslapis 47. ISBN: 9781118646106. <https://doi.org/https://doi.org/10.1002/9781118646106.ch3>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118646106.ch3>.
- [HGD⁺18] K. He, G. Gkioxari, P. Dollár, R. Girshick. *Mask R-CNN*. 2018.
- [HYS⁺17] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, G. Bing. „Learning from class-imbalanced data: Review of methods and applications“. Iš: *Expert Systems with Applications* 73 (2017), puslapiai 220–239. ISSN: 0957-4174. <https://doi.org/https://doi.org/10.1016/j.eswa.2016.12.035>. URL: <https://www.sciencedirect.com/science/article/pii/S0957417416307175>.
- [YHX⁺16] L. Yijing, G. Haixiang, L. Xiao, L. Yanan, L. Jinling. „Adapted ensemble classification algorithm based on multiple classifier system and feature selection for classifying multi-class imbalanced data“. Iš: *Knowledge-Based Systems* 94 (2016), puslapiai 88–104. ISSN: 0950-7051. <https://doi.org/https://doi.org/10.1016/j.knosys.2015.11.013>. URL: <https://www.sciencedirect.com/science/article/pii/S0950705115004566>.
- [YZ21] L. Yu, N. Zhou. *Survey of Imbalanced Data Methodologies*. 2021.
- [Jap13] N. Japkowicz. „Assessment Metrics for Imbalanced Learning“. Iš: *Imbalanced Learning*. John Wiley Sons, Ltd, 2013. Skyrius 8. ISBN: 9781118646106. <https://doi.org/https://doi.org/10.1002/9781118646106.ch8>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118646106.ch8>.
- [JHC⁺15] P. Jun, P. Huang, N. Christou, F. P. Schoenberg, Y. Wu. „Classification of Imbalanced Data Using Synthetic Over-Sampling Techniques“. Iš: 2015. URL: <https://api.semanticscholar.org/CorpusID:55362853>.
- [JJ04] T. Jo, N. Japkowicz. „Class imbalances versus small disjuncts“. Iš: *SIGKDD Explor. Newsl.* 6.1 (2004), puslapiai 40–49. ISSN: 1931-0145. <https://doi.org/10.1145/1007730.1007737>. URL: <https://doi.org/10.1145/1007730.1007737>.

- [JK19] J. Johnson, T. Khoshgoftaar. „Survey on deep learning with class imbalance“. Iš: *Journal of Big Data* 6 (2019), puslapis 27. <https://doi.org/10.1186/s40537-019-0192-5>.
- [JWC⁺20] A. B. Jung, K. Wada, J. Crall, S. Tanaka ir kiti. *imgaug*. <https://github.com/aleju/imgaug>. Online; accessed 01-Feb-2020. 2020.
- [KKW19] M. Koziarski, B. Krawczyk, M. Woźniak. „Radial-Based oversampling for noisy imbalanced data classification“. Iš: *Neurocomputing* 343 (2019). Learning in the Presence of Class Imbalance and Concept Drift, puslapiai 19–33. ISSN: 0925-2312. <https://doi.org/https://doi.org/10.1016/j.neucom.2018.04.089>. URL: <https://www.sciencedirect.com/science/article/pii/S0925231219301596>.
- [KS13] B. Krawczyk, G. Schaefer. „An improved ensemble approach for imbalanced classification problems“. Iš: *2013 IEEE 8th International Symposium on Applied Computational Intelligence and Informatics (SACI)*. 2013, puslapiai 423–426. <https://doi.org/10.1109/SACI.2013.6609011>.
- [KW17] M. Koziarski, M. Wozniak. „CCR: A combined cleaning and resampling algorithm for imbalanced data classification“. Iš: *International Journal of Applied Mathematics and Computer Science* 27 (2017), puslapiai 727–736. <https://doi.org/10.1515/amcs-2017-0050>.
- [KWS14] B. Krawczyk, M. Woźniak, G. Schaefer. „Cost-sensitive decision tree ensembles for effective imbalanced classification“. Iš: *Applied Soft Computing* 14 (2014), puslapiai 554–562. ISSN: 1568-4946. <https://doi.org/https://doi.org/10.1016/j.asoc.2013.08.014>. URL: <https://www.sciencedirect.com/science/article/pii/S1568494613002895>.
- [LBB⁺98] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner. „Gradient-based learning applied to document recognition“. Iš: *Proceedings of the IEEE* 86.11 (1998), puslapiai 2278–2324. <https://doi.org/10.1109/5.726791>.
- [LFG⁺13] V. López, A. Fernández, S. García, V. Palade, F. Herrera. „An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics“. Iš: *Information Sciences* 250 (2013), puslapiai 113–141. ISSN: 0020-0255. <https://doi.org/https://doi.org/10.1016/j.ins.2013.07.007>. URL: <https://www.sciencedirect.com/science/article/pii/S0020025513005124>.
- [LJL⁺20] X. Liang, A. Jiang, T. Li, Y. Xue, G. Wang. „LR-SMOTE — An improved unbalanced data set oversampling based on K-means and SVM“. Iš: *Knowledge-Based Systems* 196 (2020), puslapis 105845. ISSN: 0950-7051. <https://doi.org/https://doi.org/10.1016/j.knosys.2020.105845>. URL: <https://www.sciencedirect.com/science/article/pii/S0950705120302148>.

- [LLM20] E. Lashgari, D. Liang, U. Maoz. „Data augmentation for deep-learning-based electroencephalography“. Iš: *Journal of Neuroscience Methods* 346 (2020), puslapiai 108885. issn: 0165-0270. <https://doi.org/https://doi.org/10.1016/j.jneumeth.2020.108885>. URL: <https://www.sciencedirect.com/science/article/pii/S0165027020303083>.
- [LMC⁺16] O. Loyola-González, J. F. Martínez-Trinidad, J. A. Carrasco-Ochoa, M. García-Borroto. „Study of the impact of resampling methods for contrast pattern based classifiers in imbalanced databases“. Iš: *Neurocomputing* 175 (2016), puslapiai 935–947. issn: 0925-2312. <https://doi.org/https://doi.org/10.1016/j.neucom.2015.04.120>. URL: <https://www.sciencedirect.com/science/article/pii/S0925231215015908>.
- [LNA17] G. Lemaître, F. Nogueira, C. K. Aridas. „Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning“. Iš: *Journal of Machine Learning Research* 18.17 (2017), puslapiai 1–5. URL: <http://jmlr.org/papers/v18/16-365.html>.
- [LTH⁺17] W.-C. Lin, C.-F. Tsai, Y.-H. Hu, J.-S. Jhang. „Clustering-based undersampling in class-imbalanced data“. Iš: *Information Sciences* 409-410 (2017), puslapiai 17–26. issn: 0020-0255. <https://doi.org/https://doi.org/10.1016/j.ins.2017.05.008>. URL: <https://www.sciencedirect.com/science/article/pii/S0020025517307235>.
- [MBP⁺20] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, D. Terzopoulos. *Image Segmentation Using Deep Learning: A Survey*. 2020.
- [MDD20] S. S. Mullick, S. Datta, S. Das. *Generative Adversarial Minority Oversampling*. 2020.
- [Meh21] B. Mehlig. „Machine Learning with Neural Networks“. Iš: (2021). <https://doi.org/10.1017/9781108860604>. URL: <http://dx.doi.org/10.1017/9781108860604>.
- [ML02] H. Motoda, H. Liu. „Feature selection, extraction and construction“. Iš: *Communication of IICM (Institute of Information and Computing Machinery, Taiwan)* 5 (2002), puslapiai 67–72.
- [MMM23] T. Mūžas, A. V. Misiukas Misiūnas, T. Meškauskas. „Large Scale Study of Binary Galaxy Image Classification and the Impact of Image Augmentation Techniques“. Iš: *Computational Science and Its Applications – ICCSA 2023*. Sudarė O. Gervasi, B. Murgante, D. Tanar, B. O. Apduhan, A. C. Braga, C. Garau, A. Stratigea. Cham: Springer Nature Switzerland, 2023, puslapiai 402–412. ISBN: 978-3-031-36808-0.
- [NWC⁺11] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Ng. „Reading Digits in Natural Images with Unsupervised Feature Learning“. Iš: *NIPS* (2011).

- [PGM⁺19] A. Paszke, S. Gross, F. Massa, A. Lerer ir kiti. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. 2019. URL: <https://arxiv.org/abs/1912.01703>.
- [PVG⁺11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel ir kiti. „Scikit-learn: Machine Learning in Python“. Iš: *Journal of Machine Learning Research* 12 (2011), puslapiai 2825–2830.
- [RHW19] Y. Roh, G. Heo, S. E. Whang. *A Survey on Data Collection for Machine Learning: a Big Data – AI Integration Perspective*. 2019.
- [RTG⁺19] H. Rezatofghi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese. *Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression*. 2019.
- [Tom76] I. Tomek. „Two Modifications of CNN“. Iš: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-6.11 (1976), puslapiai 769–772. <https://doi.org/10.1109/TSMC.1976.4309452>.
- [WC10] M. Wasikowski, X.-w. Chen. „Combating the Small Sample Class Imbalance Problem Using Feature Selection“. Iš: *IEEE Transactions on Knowledge and Data Engineering* 22.10 (2010), puslapiai 1388–1400. <https://doi.org/10.1109/TKDE.2009.187>.
- [Wil72] D. L. Wilson. „Asymptotic Properties of Nearest Neighbor Rules Using Edited Data“. Iš: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-2.3 (1972), puslapiai 408–421. <https://doi.org/10.1109/TSMC.1972.4309137>.
- [XRV17] H. Xiao, K. Rasul, R. Vollgraf. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. 2017. URL: <https://arxiv.org/abs/1708.07747>.

Sąvokų apibrėžimai

Didžiausia klasė (angl. *Majority class*)
Mažiausia klasė (angl. *Minority class*)
Neigiama klasė (angl. *Negative class*)
Teigiama klasė (angl. *Positive class*)
Neigiami pavyzdžiai (angl. *Negative instances*)
Teigiami pavyzdžiai (angl. *Positive instances*)
Perteklinis įtraukimas (angl. *Oversampling*)
Imties retinimas (angl. *Undersampling*)
Kaštams jautrus mokymasis (angl. *Cost-sensitive learning*)
Ansamblis (angl. *Ensemble*)
Požymių erdvė (angl. *Feature space*)
Priešiškas mokymas (angl. *Adversarial learning*)
Perstatymas (angl. *Permute*)
Bauda (angl. *Penalty*)

1 priedas.

Eksperimento failai

Eksperimento rezultatų failas bei programinis kodas yra pateikti paskirtame MIF Drive aplanke atitinkamai .json bei .ipynb formatais.

2 priedas.

MNIST -99% rezultatai

12 lentelė. MNIST rezultatai po 99% duomenų pašalinimo iš klasės su pirmuoju modeliu

Klasė	Atkūrimas	AP	F1	ROC AUC
0	0,9878 → 0,8173	0,9719 → 0,8322	0,9852 → 0,8980	0,9929 → 0,9085
1	0,9921 → 0,7304	0,9782 → 0,7610	0,9886 → 0,8442	0,9951 → 0,8652
2	0,9690 → 0,6260	0,9513 → 0,6636	0,9737 → 0,7695	0,9833 → 0,8129
3	0,9911 → 0,5545	0,9351 → 0,5985	0,9662 → 0,7129	0,9922 → 0,7772
4	0,9796 → 0,4409	0,9491 → 0,4958	0,9732 → 0,6120	0,9880 → 0,7205
5	0,9697 → 0,5841	0,9494 → 0,6178	0,9730 → 0,7359	0,9837 → 0,7919
6	0,9781 → 0,7244	0,9648 → 0,7487	0,9812 → 0,8392	0,9882 → 0,8621
7	0,9718 → 0,6294	0,9538 → 0,6665	0,9751 → 0,7721	0,9847 → 0,8146
8	0,9528 → 0,5862	0,9383 → 0,6255	0,9662 → 0,7387	0,9753 → 0,7931
9	0,9732 → 0,6016	0,9434 → 0,6388	0,9699 → 0,7498	0,9847 → 0,8006

13 lentelė. MNIST rezultatai po 99% duomenų pašalinimo iš klasės su antruoju modeliu

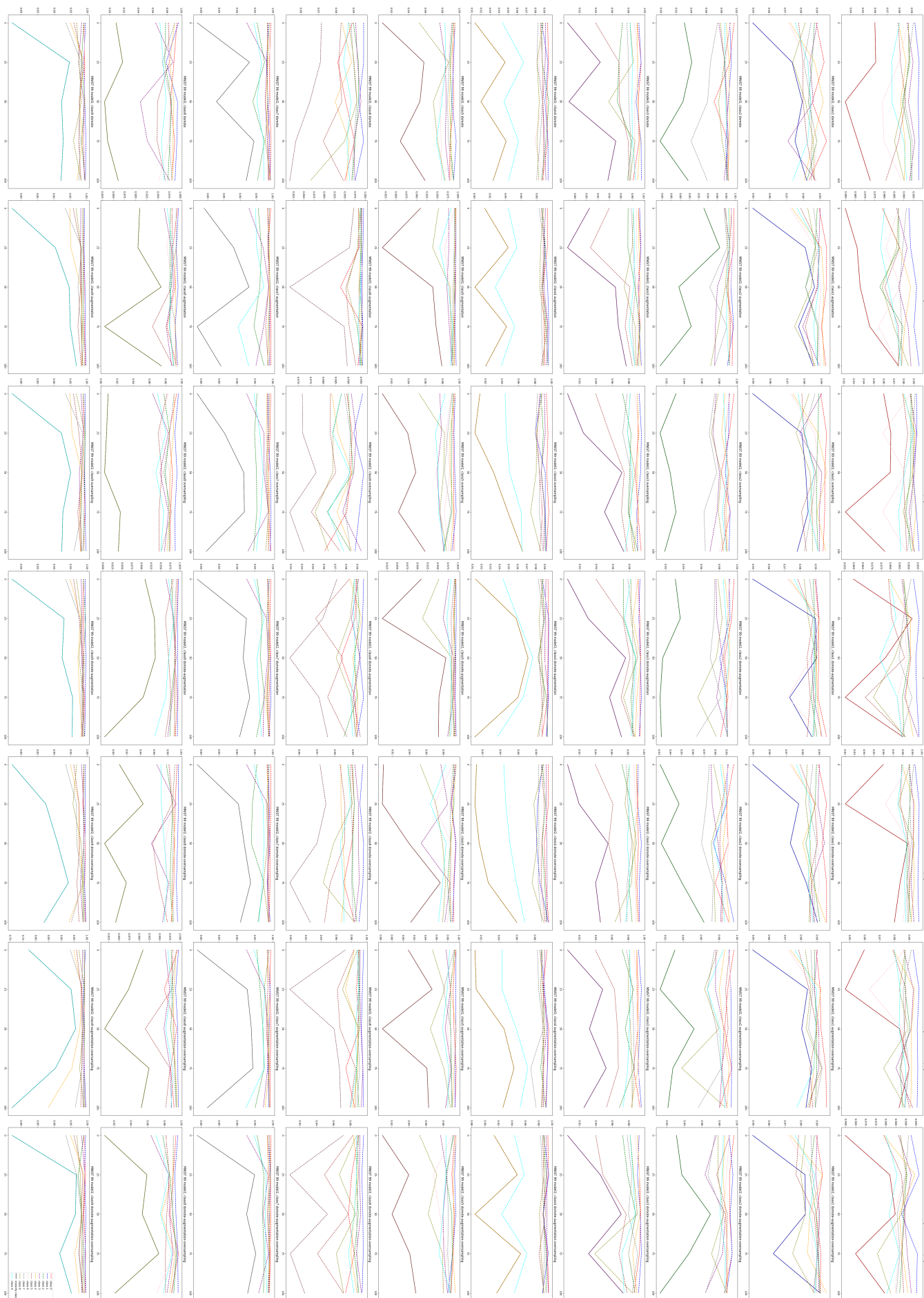
Klasė	Atkūrimas	AP	F1	ROC AUC
0	0,9980 → 0,9235	0,9911 → 0,9300	0,9954 → 0,9597	0,9986 → 0,9617
1	0,9991 → 0,9040	0,9896 → 0,9149	0,9947 → 0,9496	0,9989 → 0,9520
2	0,9942 → 0,8721	0,9842 → 0,8834	0,9918 → 0,9307	0,9965 → 0,9359
3	0,9950 → 0,8267	0,9848 → 0,8442	0,9921 → 0,9051	0,9969 → 0,9134
4	0,9969 → 0,8411	0,9822 → 0,8557	0,9909 → 0,9132	0,9976 → 0,9205
5	0,9910 → 0,8341	0,9731 → 0,8478	0,9861 → 0,9090	0,9946 → 0,9170
6	0,9896 → 0,9478	0,9741 → 0,9497	0,9865 → 0,9716	0,9939 → 0,9737
7	0,9874 → 0,6508	0,9838 → 0,6867	0,9912 → 0,7885	0,9934 → 0,8254
8	0,9908 → 0,8326	0,9866 → 0,8489	0,9928 → 0,9087	0,9951 → 0,9163
9	0,9921 → 0,6283	0,9792 → 0,6658	0,9891 → 0,7718	0,9953 → 0,8142

14 lentelė. MNIST rezultatai po 99% duomenų pašalinimo iš klasės su trečiuoju modeliu

Klasė	Atkūrimas	AP	F1	ROC AUC
0	0,9939 → 0,8408	0,9924 → 0,8544	0,9959 → 0,9125	0,9968 → 0,9203
1	0,9982 → 0,9286	0,9888 → 0,9359	0,9943 → 0,9626	0,9985 → 0,9643
2	0,9893 → 0,7238	0,9895 → 0,7523	0,9942 → 0,8398	0,9946 → 0,8619
3	0,9990 → 0,8168	0,9942 → 0,8334	0,9970 → 0,8982	0,9992 → 0,9083
4	0,9450 → 0,8564	0,9504 → 0,8695	0,9717 → 0,9221	0,9725 → 0,9282
5	0,9944 → 0,7791	0,9816 → 0,7988	0,9905 → 0,8759	0,9965 → 0,8896
6	0,9864 → 0,9530	0,9856 → 0,9565	0,9921 → 0,9754	0,9931 → 0,9765
7	0,9893 → 0,7033	0,9846 → 0,7328	0,9917 → 0,8253	0,9943 → 0,8516
8	0,9938 → 0,8624	0,9301 → 0,8738	0,9637 → 0,9251	0,9932 → 0,9311
9	0,9078 → 0,1606	0,9132 → 0,2453	0,9497 → 0,2767	0,9537 → 0,5803



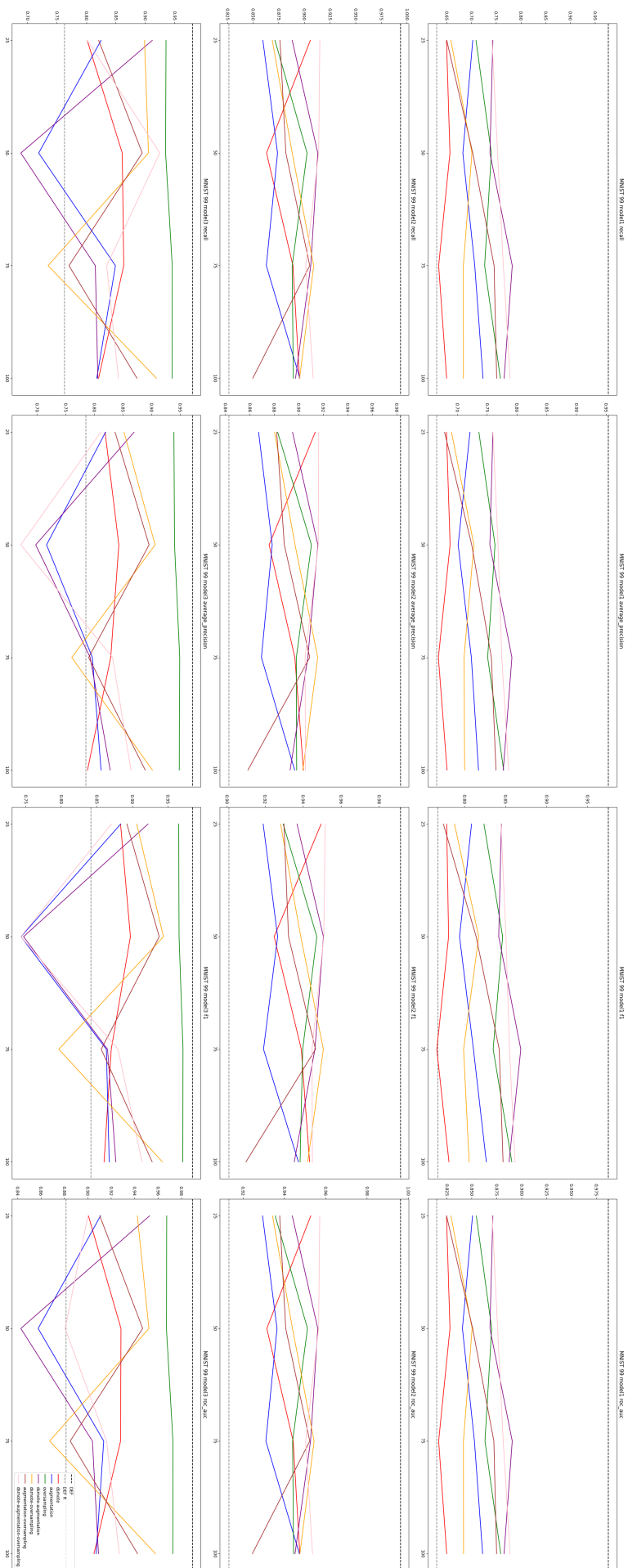
26 pav. MNIST -99% Modelis 1 visų klasių F1 rezultatai



27 pav. MNIST -99% Modelis 2 visų klasių F1 rezultatai



28 pav. MNIST -99% Modelis 3 visų klasių F1 rezultatai



29 pav. MNIST -99% rezultatų vidurkliai

3 priedas.

MNIST -99,9% rezultatai

15 lentelė. MNIST rezultatai po 99,9% duomenų pašalinimo iš klasės su pirmuoju modeliu

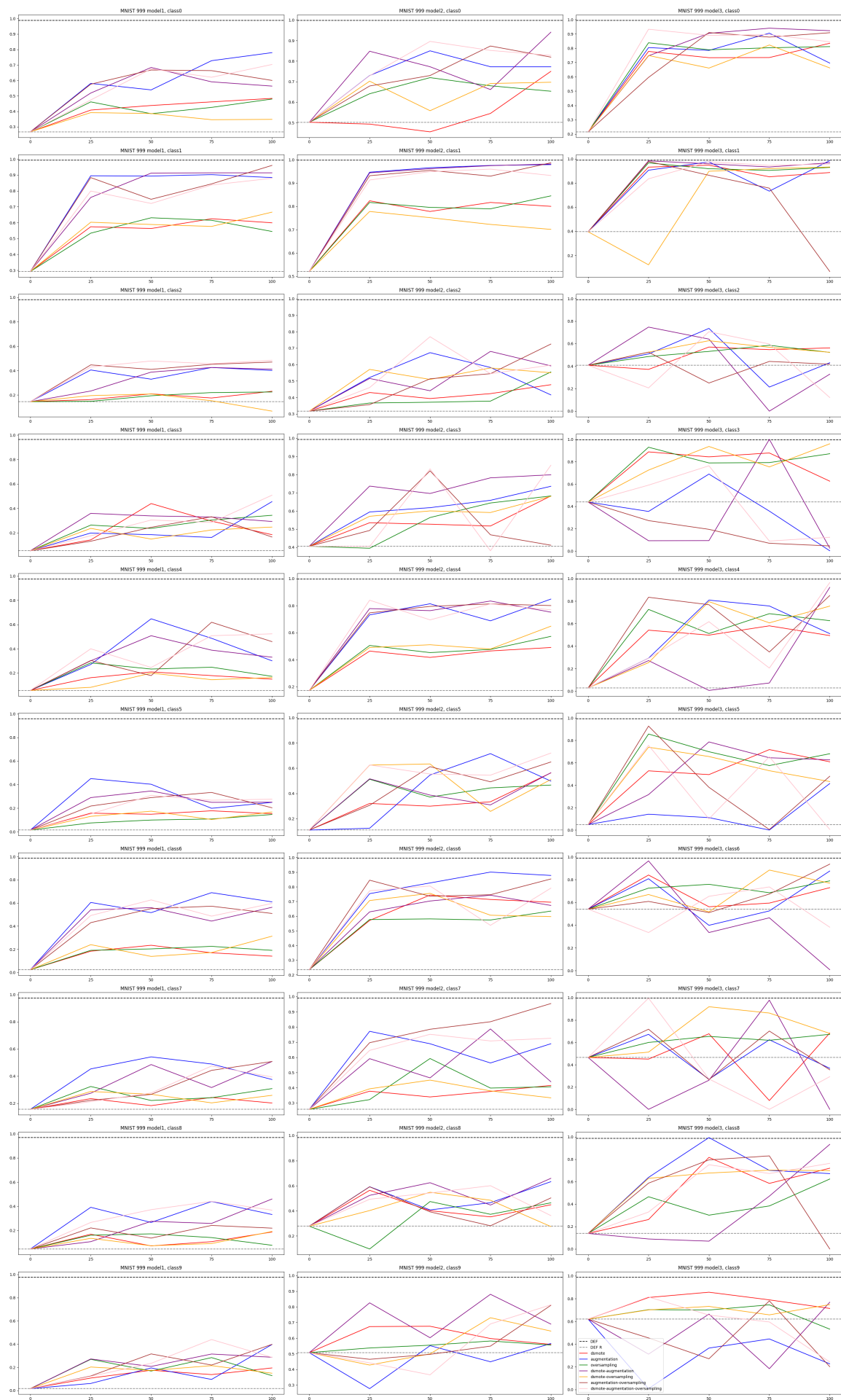
Klasė	Atkūrimas	AP	F1	ROC AUC
0	0,9878 → 0,2673	0,9719 → 0,3391	0,9852 → 0,4219	0,9929 → 0,6337
1	0,9921 → 0,2925	0,9782 → 0,3728	0,9886 → 0,4526	0,9951 → 0,6463
2	0,9690 → 0,1453	0,9513 → 0,2335	0,9737 → 0,2538	0,9833 → 0,5727
3	0,9911 → 0,0535	0,9351 → 0,1491	0,9662 → 0,1015	0,9922 → 0,5267
4	0,9796 → 0,0570	0,9491 → 0,1496	0,9732 → 0,1079	0,9880 → 0,5285
5	0,9697 → 0,0135	0,9494 → 0,1015	0,9730 → 0,0265	0,9837 → 0,5067
6	0,9781 → 0,0240	0,9648 → 0,1175	0,9812 → 0,0469	0,9882 → 0,5120
7	0,9718 → 0,1556	0,9538 → 0,2424	0,9751 → 0,2694	0,9847 → 0,5778
8	0,9528 → 0,0462	0,9383 → 0,1391	0,9662 → 0,0883	0,9753 → 0,5231
9	0,9732 → 0,0159	0,9434 → 0,1152	0,9699 → 0,0312	0,9847 → 0,5079

16 lentelė. MNIST rezultatai po 99,9% duomenų pašalinimo iš klasės su antruoju modeliu

Klasė	Atkūrimas	AP	F1	ROC AUC
0	0,9980 → 0,5020	0,9911 → 0,5508	0,9954 → 0,6685	0,9986 → 0,7510
1	0,9991 → 0,5207	0,9896 → 0,5751	0,9947 → 0,6848	0,9989 → 0,7604
2	0,9942 → 0,3169	0,9842 → 0,3874	0,9918 → 0,4812	0,9965 → 0,6584
3	0,9950 → 0,4050	0,9848 → 0,4651	0,9921 → 0,5765	0,9969 → 0,7025
4	0,9969 → 0,1721	0,9822 → 0,2534	0,9909 → 0,2937	0,9976 → 0,5860
5	0,9910 → 0,1099	0,9731 → 0,1893	0,9861 → 0,1980	0,9946 → 0,5549
6	0,9896 → 0,2349	0,9741 → 0,3082	0,9865 → 0,3804	0,9939 → 0,6174
7	0,9874 → 0,2578	0,9838 → 0,3341	0,9912 → 0,4099	0,9934 → 0,6289
8	0,9908 → 0,2782	0,9866 → 0,3485	0,9928 → 0,4353	0,9951 → 0,6391
9	0,9921 → 0,5074	0,9792 → 0,5571	0,9891 → 0,6732	0,9953 → 0,7537

17 lentelė. MNIST rezultatai po 99,9% duomenų pašalinimo iš klasės su trečiuoju modeliu

Klasė	Atkūrimas	AP	F1	ROC AUC
0	0,9939 → 0,2153	0,9924 → 0,2922	0,9959 → 0,3543	0,9968 → 0,6077
1	0,9982 → 0,3965	0,9888 → 0,4650	0,9943 → 0,5678	0,9985 → 0,6982
2	0,9893 → 0,4079	0,9895 → 0,4690	0,9942 → 0,5795	0,9946 → 0,7040
3	0,9990 → 0,4386	0,9942 → 0,4953	0,9970 → 0,6098	0,9992 → 0,7193
4	0,9450 → 0,0295	0,9504 → 0,1248	0,9717 → 0,0574	0,9725 → 0,5148
5	0,9944 → 0,0493	0,9816 → 0,1341	0,9905 → 0,0940	0,9965 → 0,5247
6	0,9864 → 0,5407	0,9856 → 0,5847	0,9921 → 0,7019	0,9931 → 0,7704
7	0,9893 → 0,4650	0,9846 → 0,5190	0,9917 → 0,6344	0,9943 → 0,7324
8	0,9938 → 0,1386	0,9301 → 0,2225	0,9637 → 0,2435	0,9932 → 0,5693
9	0,9078 → 0,6194	0,9132 → 0,6578	0,9497 → 0,7650	0,9537 → 0,8097



30 pav. MNIST -99,9% atkūrimas



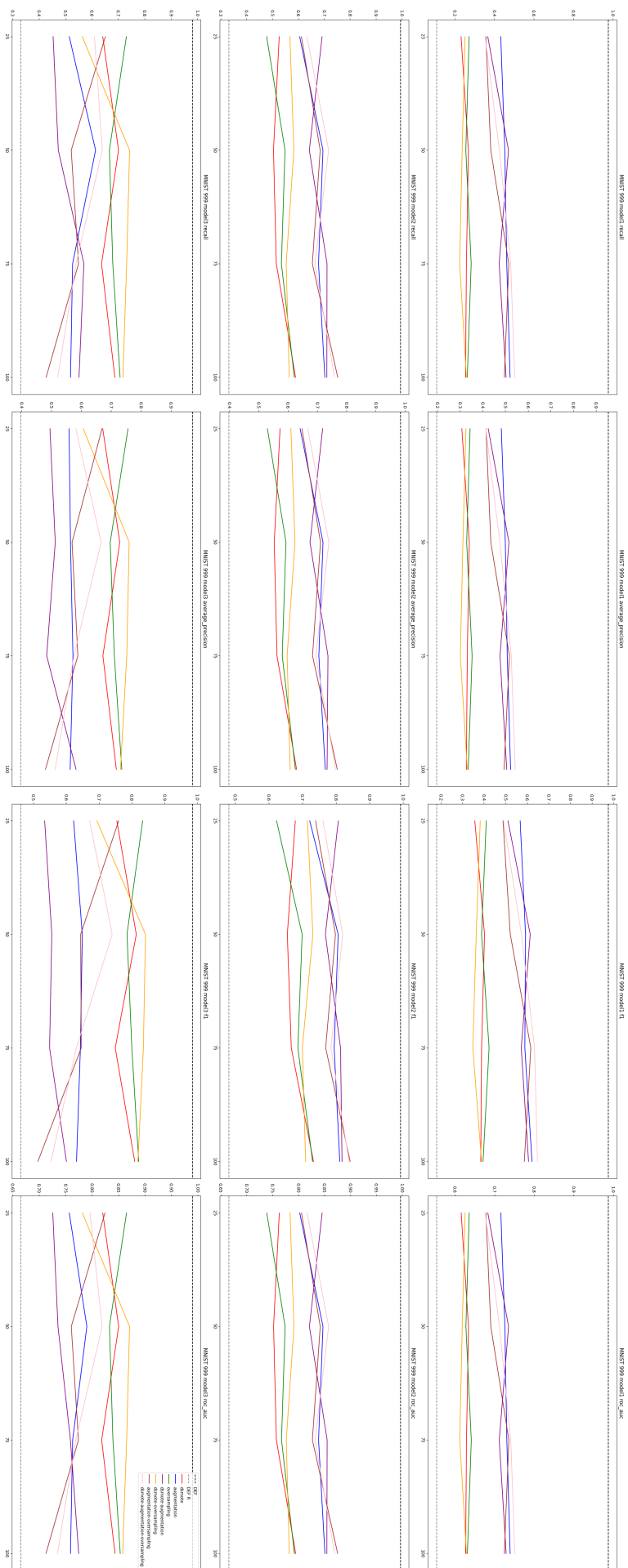
31 pav. MNIST -99,9% Modelis 1 visų klasių F1 rezultatai



32 pav. MNIST -99,9% Modelis 2 visų klasių F1 rezultatai



33 pav. MNIST -99,9% Modelis 3 visų klasių F1 rezultatai



34 pav. MNIST -99,9% rezultatų vidurkiai

4 priedas.

FMNIST -99% rezultatai

18 lentelė. FMNIST rezultatai po 99% duomenų pašalinimo iš klasės su pirmuoju modeliu

Klasė	Atkūrimas	AP	F1	ROC AUC
0	0,8130 → 0,1290	0,6897 → 0,2122	0,8191 → 0,2277	0,8969 → 0,5643
1	0,9750 → 0,8620	0,9579 → 0,8758	0,9774 → 0,9259	0,9864 → 0,9310
2	0,8520 → 0,0060	0,6347 → 0,1054	0,7849 → 0,0119	0,9083 → 0,5030
3	0,8890 → 0,3270	0,7747 → 0,3846	0,8737 → 0,4892	0,9364 → 0,6629
4	0,8650 → 0,0000	0,6547 → 0,1000	0,7983 → 0,0000	0,9157 → 0,5000
5	0,9630 → 0,3640	0,9274 → 0,4276	0,9611 → 0,5337	0,9792 → 0,6820
6	0,5220 → 0,0000	0,4563 → 0,1000	0,6263 → 0,0000	0,7529 → 0,5000
7	0,8790 → 0,1670	0,8593 → 0,2503	0,9195 → 0,2862	0,9377 → 0,5835
8	0,9750 → 0,6250	0,9465 → 0,6605	0,9716 → 0,7683	0,9857 → 0,8124
9	0,9750 → 0,1770	0,8976 → 0,2593	0,9457 → 0,3008	0,9827 → 0,5885

19 lentelė. MNIST rezultatai po 99% duomenų pašalinimo iš klasės su antruoju modeliu

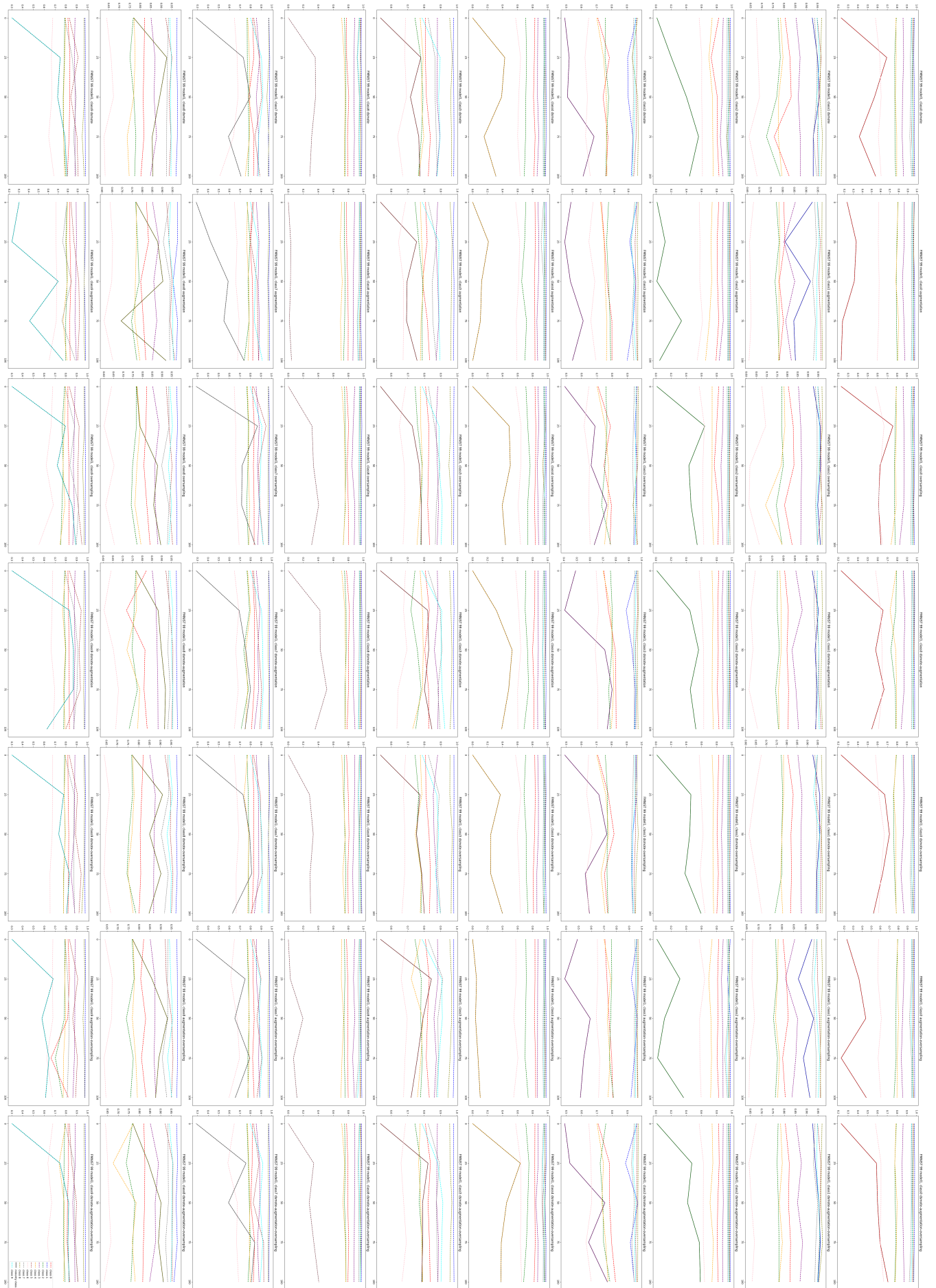
Klasė	Atkūrimas	AP	F1	ROC AUC
0	0,8110 → 0,0370	0,7323 → 0,1333	0,8439 → 0,0714	0,8993 → 0,5185
1	0,9810 → 0,7980	0,9740 → 0,8182	0,9859 → 0,8877	0,9900 → 0,8990
2	0,8480 → 0,0980	0,7401 → 0,1834	0,8514 → 0,1777	0,9160 → 0,5487
3	0,9270 → 0,1970	0,8203 → 0,2763	0,9013 → 0,3289	0,9563 → 0,5984
4	0,8820 → 0,0130	0,7443 → 0,1117	0,8555 → 0,0257	0,9310 → 0,5065
5	0,9730 → 0,7640	0,9619 → 0,7856	0,9794 → 0,8652	0,9857 → 0,8819
6	0,7430 → 0,0000	0,5653 → 0,1000	0,7346 → 0,0000	0,8559 → 0,5000
7	0,9610 → 0,2660	0,9201 → 0,3394	0,9572 → 0,4202	0,9779 → 0,6330
8	0,9790 → 0,5450	0,9615 → 0,5905	0,9795 → 0,7055	0,9884 → 0,7725
9	0,9550 → 0,6330	0,9370 → 0,6687	0,9656 → 0,7748	0,9762 → 0,8164

20 lentelė. FMNIST rezultatai po 99% duomenų pašalinimo iš klasės su trečiuoju modeliu

Klasė	Atkūrimas	AP	F1	ROC AUC
0	0,8110 → 0,1580	0,7513 → 0,2412	0,8546 → 0,2726	0,9007 → 0,5789
1	0,9770 → 0,8900	0,9753 → 0,9010	0,9864 → 0,9418	0,9883 → 0,9450
2	0,8870 → 0,0330	0,7926 → 0,1297	0,8839 → 0,0639	0,9368 → 0,5165
3	0,9450 → 0,4170	0,8576 → 0,4704	0,9229 → 0,5865	0,9668 → 0,7082
4	0,9050 → 0,0390	0,7793 → 0,1351	0,8769 → 0,0751	0,9437 → 0,5195
5	0,9810 → 0,6660	0,9576 → 0,6964	0,9776 → 0,7981	0,9891 → 0,8328
6	0,7180 → 0,0020	0,5879 → 0,1011	0,7475 → 0,0040	0,8477 → 0,5009
7	0,9650 → 0,5010	0,9357 → 0,5509	0,9655 → 0,6676	0,9806 → 0,7505
8	0,9730 → 0,7410	0,9523 → 0,7649	0,9745 → 0,8503	0,9852 → 0,8704
9	0,9710 → 0,7380	0,9457 → 0,7622	0,9710 → 0,8483	0,9839 → 0,8689



35 pav. FMNIST -99% atkūrimas



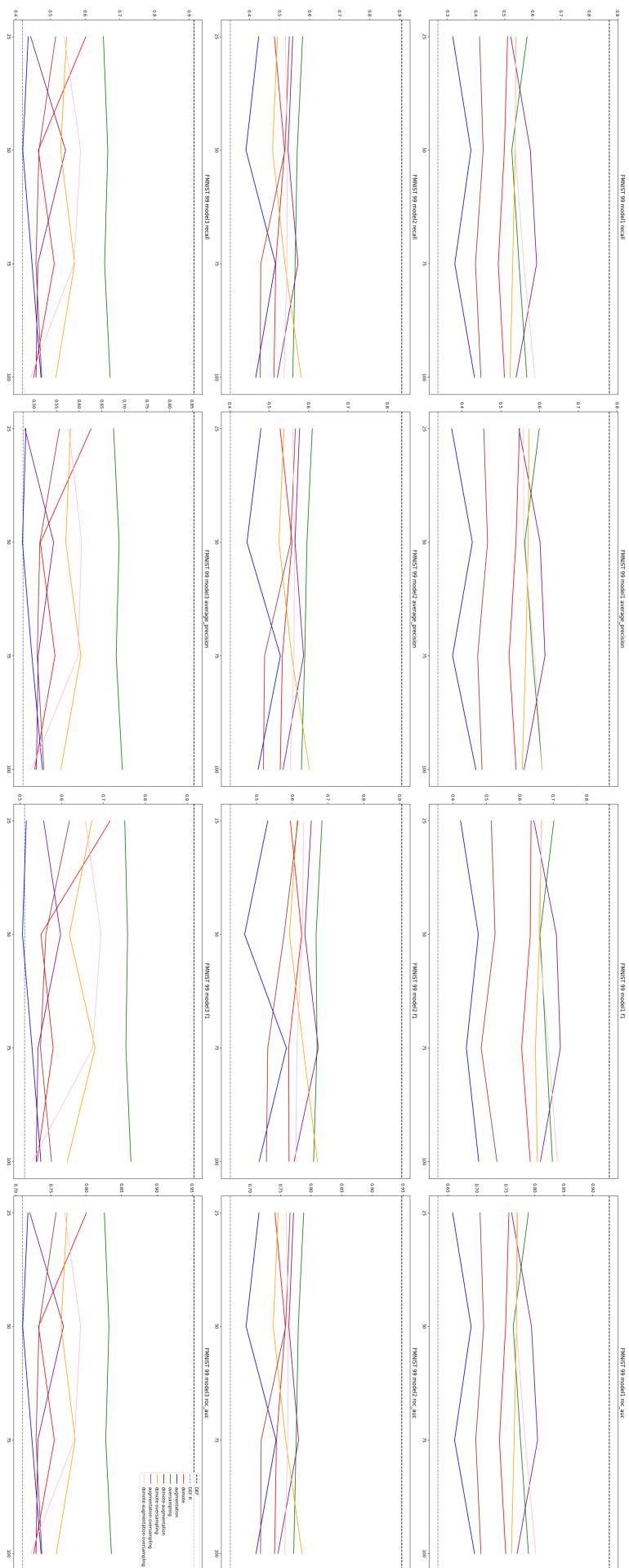
36 pav. FMNIST -99% Modelis 1 visų klasių F1 rezultatai



37 pav. FMNIST -99% Modelis 2 visų klasių F1 rezultatai



38 pav. FMNIST -99% Modelis 3 visų klasių F1 rezultatai



39 pav. FMNIST -99% rezultatų vidurkiai

5 priedas.

FMNIST -99,9% rezultatai

21 lentelė. FMNIST rezultatai po 99,9% duomenų pašalinimo iš klasės su pirmuoju modeliu

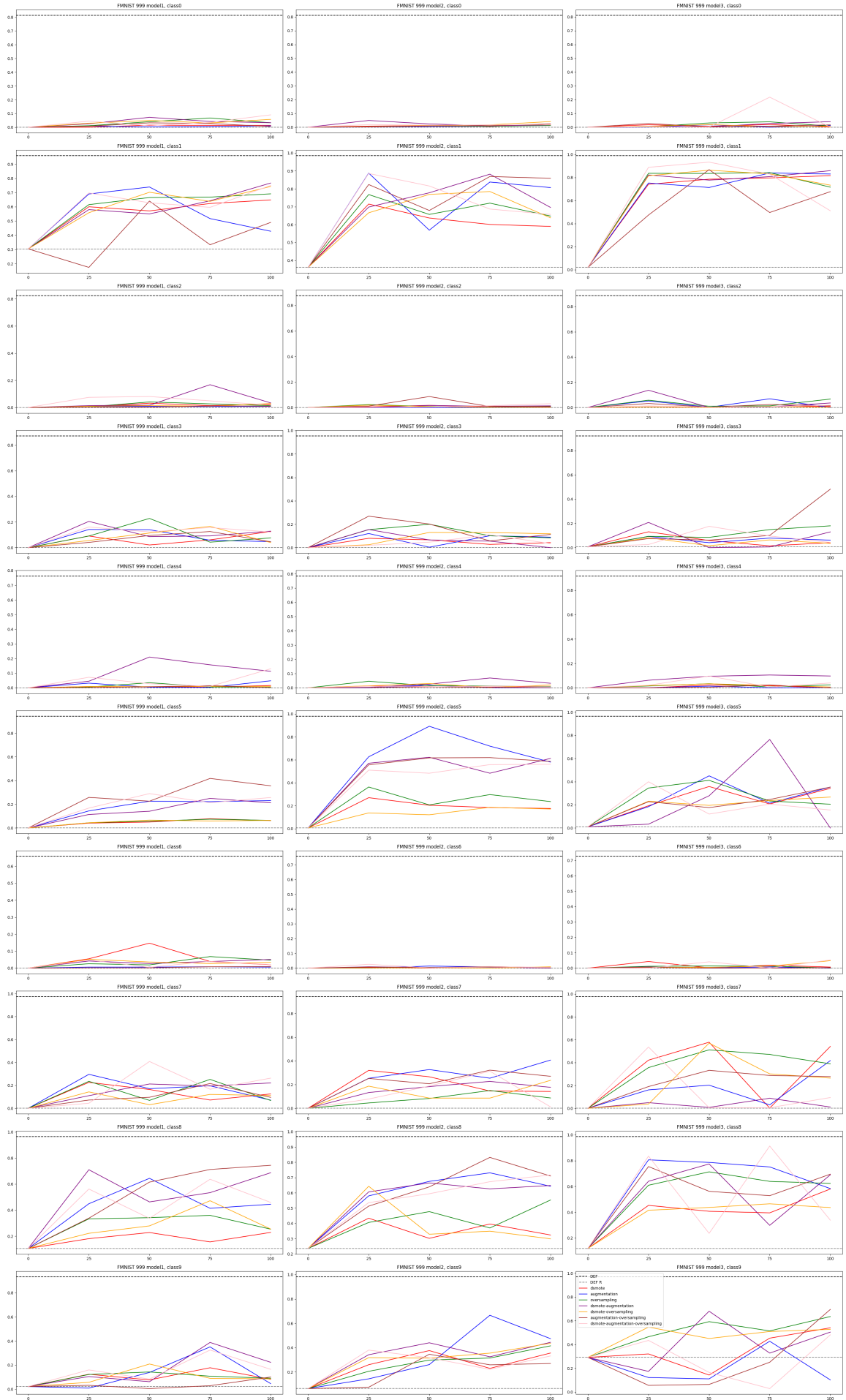
Klasė	Atkūrimas	AP	F1	ROC AUC
0	0,8130 → 0,0000	0,6897 → 0,1000	0,8191 → 0,0000	0,8969 → 0,5000
1	0,9750 → 0,3040	0,9579 → 0,3726	0,9774 → 0,4659	0,9864 → 0,6519
2	0,8520 → 0,0000	0,6347 → 0,1000	0,7849 → 0,0000	0,9083 → 0,5000
3	0,8890 → 0,0000	0,7747 → 0,1000	0,8737 → 0,0000	0,9364 → 0,5000
4	0,8650 → 0,0000	0,6547 → 0,1000	0,7983 → 0,0000	0,9157 → 0,5000
5	0,9630 → 0,0000	0,9274 → 0,1000	0,9611 → 0,0000	0,9792 → 0,5000
6	0,5220 → 0,0000	0,4563 → 0,1000	0,6263 → 0,0000	0,7529 → 0,5000
7	0,8790 → 0,0000	0,8593 → 0,1000	0,9195 → 0,0000	0,9377 → 0,5000
8	0,9750 → 0,1060	0,9465 → 0,1954	0,9716 → 0,1917	0,9857 → 0,5530
9	0,9750 → 0,0220	0,8976 → 0,1198	0,9457 → 0,0431	0,9827 → 0,5110

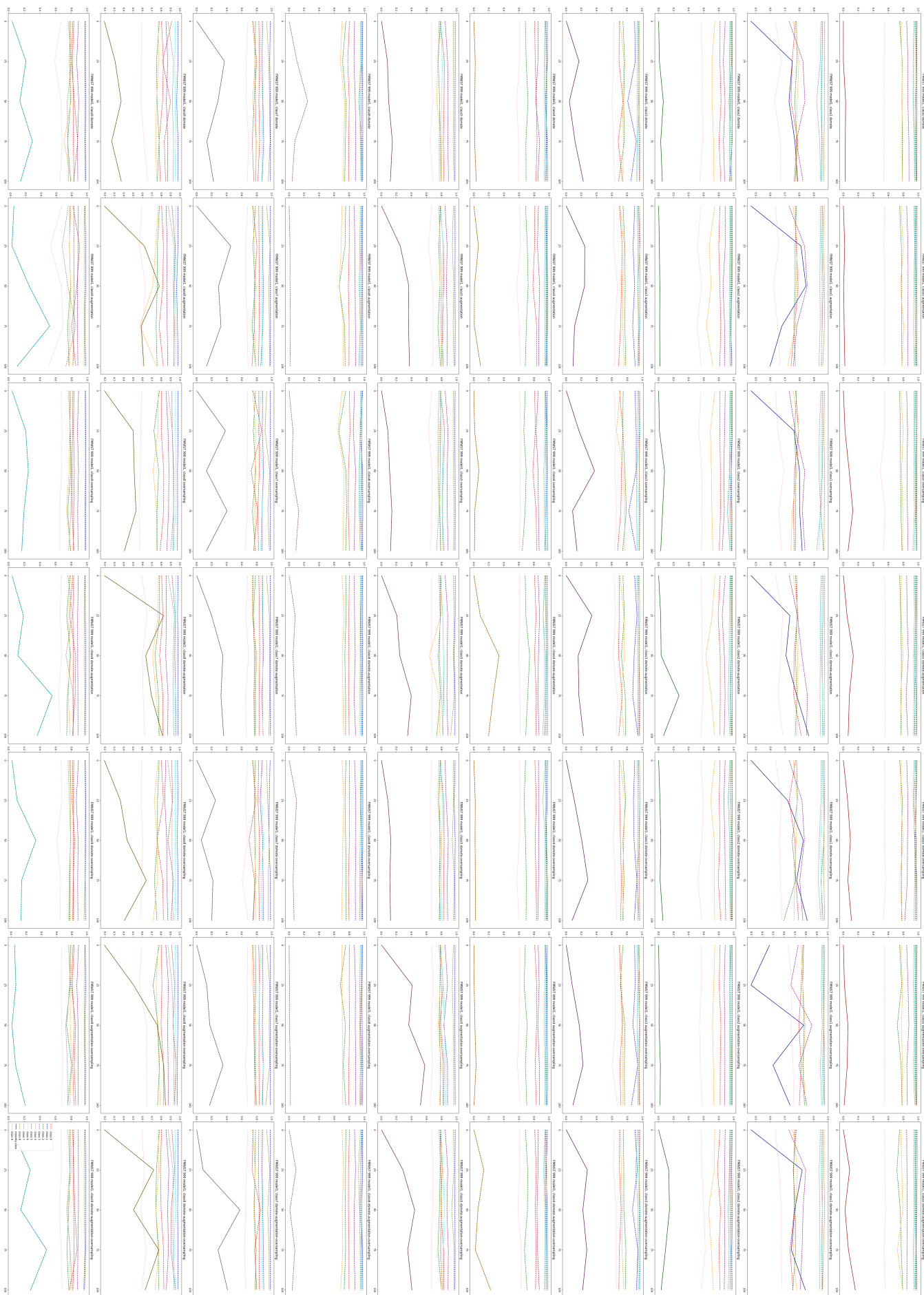
22 lentelė. FMNIST rezultatai po 99,9% duomenų pašalinimo iš klasės su antruoju modeliu

Klasė	Atkūrimas	AP	F1	ROC AUC
0	0,8110 → 0,0000	0,7323 → 0,1000	0,8439 → 0,0000	0,8993 → 0,5000
1	0,9810 → 0,3610	0,9740 → 0,4249	0,9859 → 0,5305	0,9900 → 0,6805
2	0,8480 → 0,0000	0,7401 → 0,1000	0,8514 → 0,0000	0,9160 → 0,5000
3	0,9270 → 0,0000	0,8203 → 0,1000	0,9013 → 0,0000	0,9563 → 0,5000
4	0,8820 → 0,0000	0,7443 → 0,1000	0,8555 → 0,0000	0,9310 → 0,5000
5	0,9730 → 0,0060	0,9619 → 0,1054	0,9794 → 0,0119	0,9857 → 0,5030
6	0,7430 → 0,0000	0,5653 → 0,1000	0,7346 → 0,0000	0,8559 → 0,5000
7	0,9610 → 0,0000	0,9201 → 0,1000	0,9572 → 0,0000	0,9779 → 0,5000
8	0,9790 → 0,2360	0,9615 → 0,3124	0,9795 → 0,3819	0,9884 → 0,6180
9	0,9550 → 0,0640	0,9370 → 0,1576	0,9656 → 0,1203	0,9762 → 0,5320

23 lentelė. FMNIST rezultatai po 99,9% duomenų pašalinimo iš klasės su trečiuoju modeliu

Klasė	Atkūrimas	AP	F1	ROC AUC
0	0,8110 → 0,0000	0,7513 → 0,1000	0,8546 → 0,0000	0,9007 → 0,5000
1	0,9770 → 0,0200	0,9753 → 0,1180	0,9864 → 0,0392	0,9883 → 0,5100
2	0,8870 → 0,0000	0,7926 → 0,1000	0,8839 → 0,0000	0,9368 → 0,5000
3	0,9450 → 0,0070	0,8576 → 0,1063	0,9229 → 0,0139	0,9668 → 0,5035
4	0,9050 → 0,0000	0,7793 → 0,1000	0,8769 → 0,0000	0,9437 → 0,5000
5	0,9810 → 0,0100	0,9576 → 0,1057	0,9776 → 0,0197	0,9891 → 0,5047
6	0,7180 → 0,0000	0,5879 → 0,1000	0,7475 → 0,0000	0,8477 → 0,5000
7	0,9650 → 0,0000	0,9357 → 0,1000	0,9655 → 0,0000	0,9806 → 0,4967
8	0,9730 → 0,1190	0,9523 → 0,2071	0,9745 → 0,2127	0,9852 → 0,5595
9	0,9710 → 0,2920	0,9457 → 0,3618	0,9710 → 0,4517	0,9839 → 0,6459

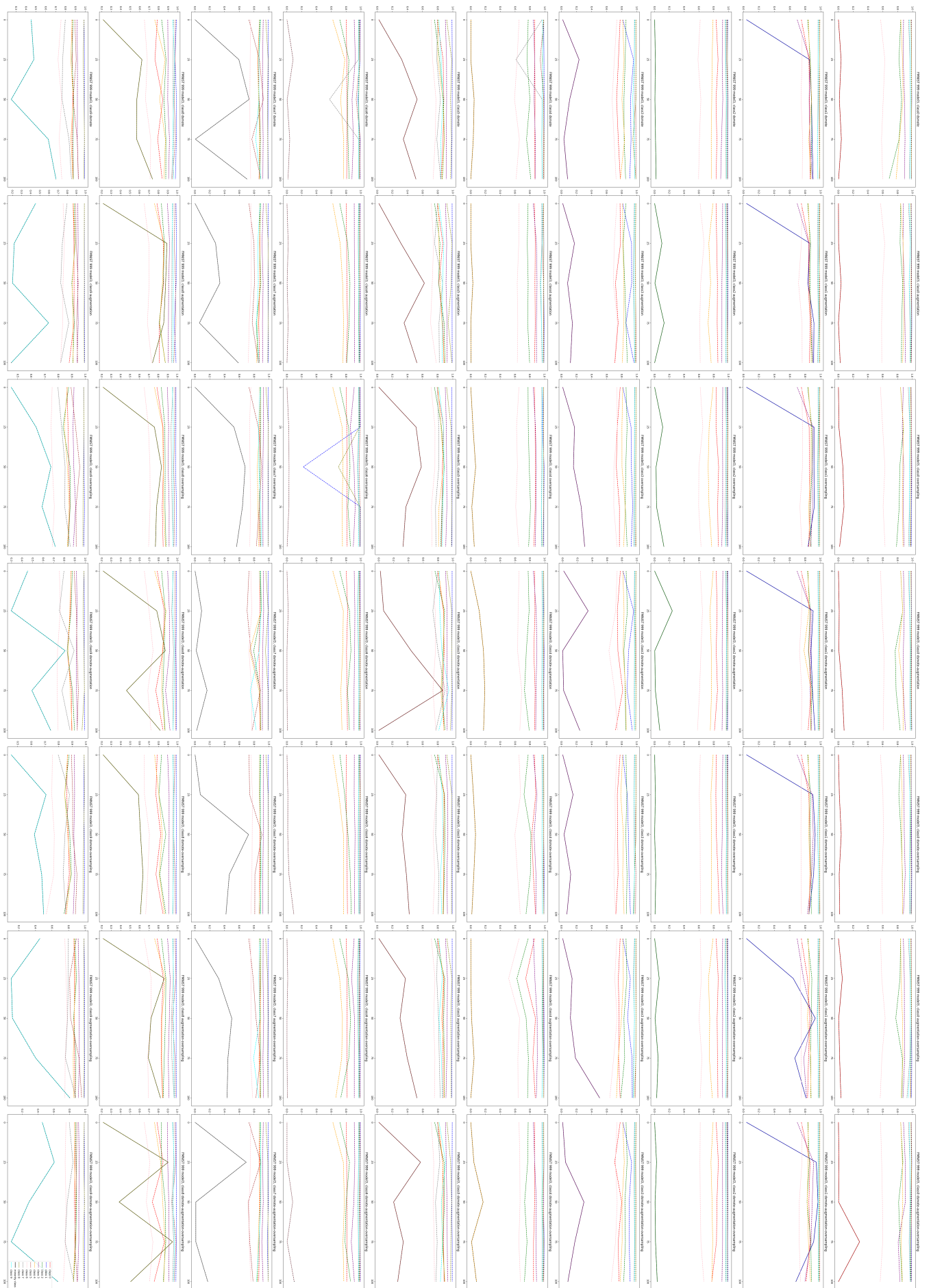




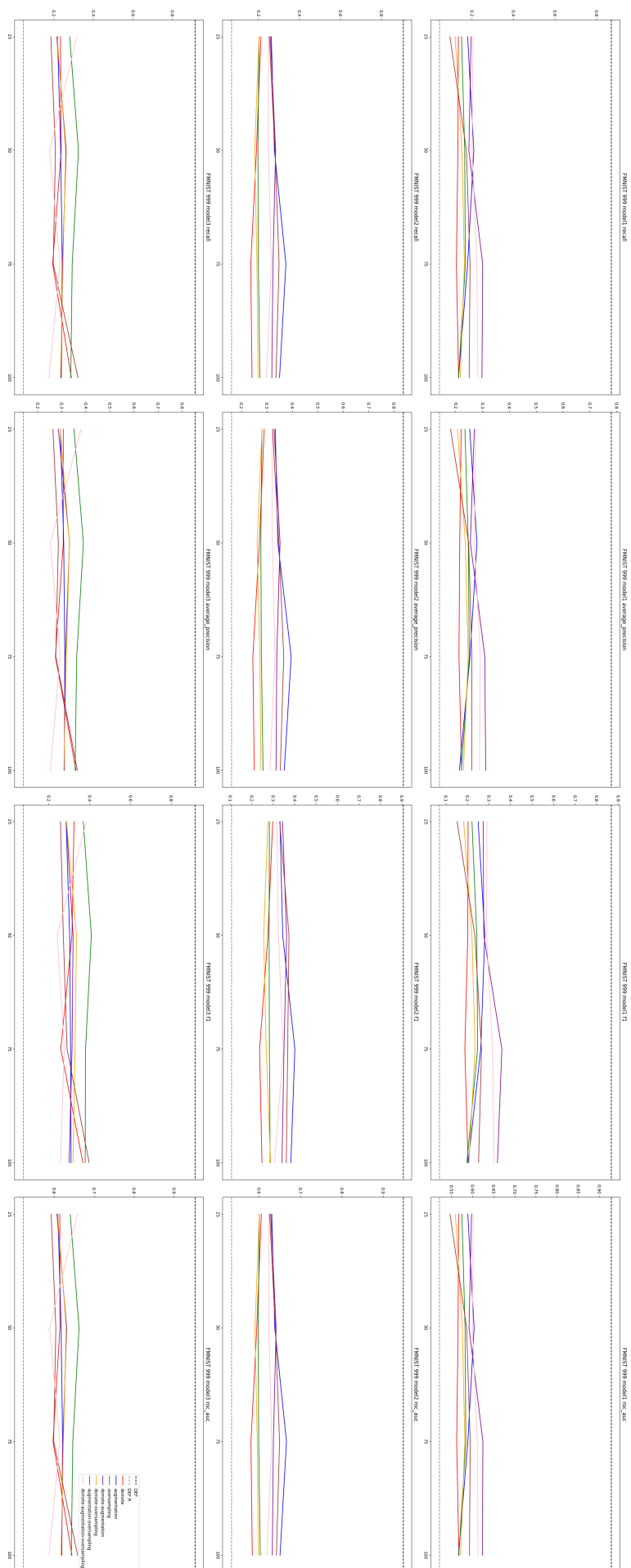
41 pav. FMNIST -99,9% Modelis 1 visų klasių F1 rezultatai



42 pav. FMNIST -99,9% Modelis 2 visų klasių F1 rezultatai



43 pav. FMNIST -99,9% Modelis 3 visų klasių F1 rezultatai



44 pav. FMNIST -99,9% rezultatų vidurkiai

6 priedas.

SVHN -99% rezultatai

24 lentelė. SVHN rezultatai po 99% duomenų pašalinimo iš klasės su antruoju modeliu

Klasė	Atkūrimas	AP	F1	ROC AUC
0	0,8956 → 0,2959	0,7952 → 0,3380	0,8878 → 0,4548	0,9434 → 0,6478
1	0,9208 → 0,3677	0,8763 → 0,4885	0,9278 → 0,5365	0,9526 → 0,6835
2	0,9152 → 0,4512	0,8726 → 0,5344	0,9268 → 0,6200	0,9519 → 0,7252
3	0,8123 → 0,1558	0,7422 → 0,2475	0,8485 → 0,2692	0,8998 → 0,5778
4	0,9294 → 0,2156	0,7880 → 0,2912	0,8827 → 0,3546	0,9553 → 0,6078
5	0,8289 → 0,1917	0,7478 → 0,2636	0,8552 → 0,3212	0,9089 → 0,5957
6	0,8862 → 0,1199	0,6837 → 0,1857	0,8193 → 0,2139	0,9317 → 0,5599
7	0,8851 → 0,2011	0,8209 → 0,2611	0,9009 → 0,3343	0,9392 → 0,6005
8	0,8012 → 0,0464	0,6729 → 0,1066	0,8125 → 0,0886	0,8948 → 0,5232
9	0,8470 → 0,1097	0,6967 → 0,1636	0,8288 → 0,1976	0,9171 → 0,5548

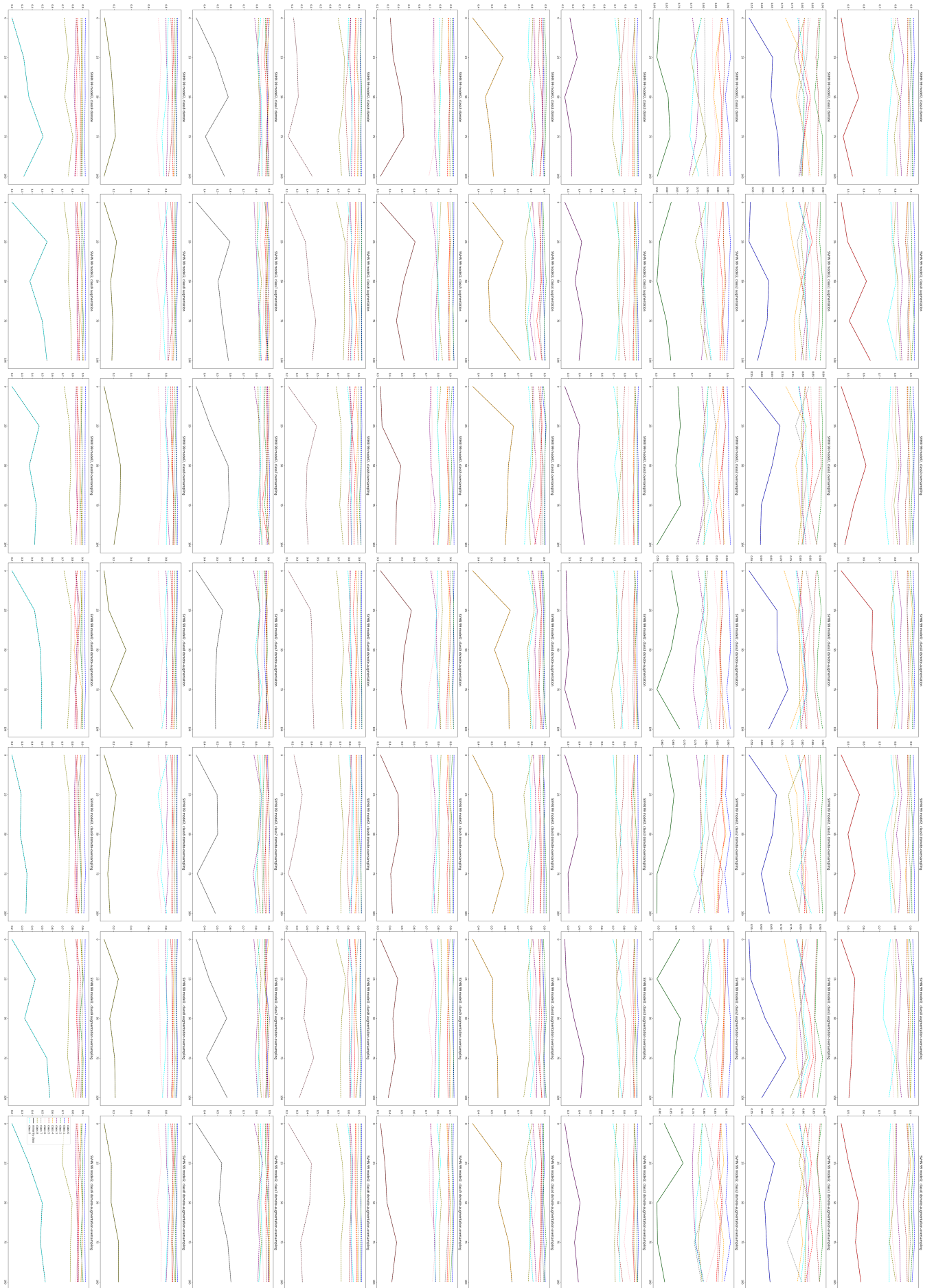
25 lentelė. SVHN rezultatai po 99% duomenų pašalinimo iš klasės su trečiuoju modeliu

Klasė	Atkūrimas	AP	F1	ROC AUC
0	0,9478 → 0,4593	0,8454 → 0,4904	0,9171 → 0,6273	0,9696 → 0,7295
1	0,9447 → 0,4328	0,9127 → 0,5424	0,9496 → 0,6035	0,9669 → 0,7162
2	0,9593 → 0,4510	0,9210 → 0,5370	0,9563 → 0,6210	0,9752 → 0,7253
3	0,8987 → 0,1693	0,8311 → 0,2609	0,9054 → 0,2895	0,9440 → 0,5846
4	0,9275 → 0,4055	0,8990 → 0,4599	0,9443 → 0,5757	0,9618 → 0,7026
5	0,9325 → 0,1745	0,8627 → 0,2497	0,9255 → 0,2970	0,9621 → 0,5872
6	0,9145 → 0,1103	0,8340 → 0,1773	0,9097 → 0,1985	0,9533 → 0,5551
7	0,9128 → 0,3264	0,8834 → 0,3762	0,9360 → 0,4912	0,9548 → 0,6631
8	0,8928 → 0,0861	0,8161 → 0,1444	0,8995 → 0,1586	0,9432 → 0,5431
9	0,9223 → 0,2157	0,8000 → 0,2606	0,8912 → 0,3539	0,9563 → 0,6077

26 lentelė. SVHN rezultatai po 99% duomenų pašalinimo iš klasės su ketvirtuoju modeliu

Klasė	Atkūrimas	AP	F1	ROC AUC
0	0,9243 → 0,0912	0,8306 → 0,1509	0,9084 → 0,1669	0,9582 → 0,5455
1	0,9386 → 0,3409	0,8777 → 0,4674	0,9304 → 0,5074	0,9597 → 0,6701
2	0,9614 → 0,2673	0,8758 → 0,3838	0,9321 → 0,4218	0,9711 → 0,6336
3	0,8557 → 0,1999	0,7946 → 0,2847	0,8820 → 0,3321	0,9226 → 0,5997
4	0,9703 → 0,3151	0,7761 → 0,3772	0,8751 → 0,4776	0,9719 → 0,6573
5	0,8754 → 0,3352	0,8212 → 0,3931	0,8996 → 0,5009	0,9341 → 0,6674
6	0,8452 → 0,0228	0,7731 → 0,0970	0,8721 → 0,0445	0,9188 → 0,5114
7	0,8207 → 0,1124	0,8106 → 0,1813	0,8894 → 0,2021	0,9093 → 0,5562
8	0,8500 → 0,0458	0,7606 → 0,1066	0,8664 → 0,0876	0,9212 → 0,5229
9	0,8232 → 0,2476	0,7661 → 0,2913	0,8678 → 0,3962	0,9092 → 0,6237

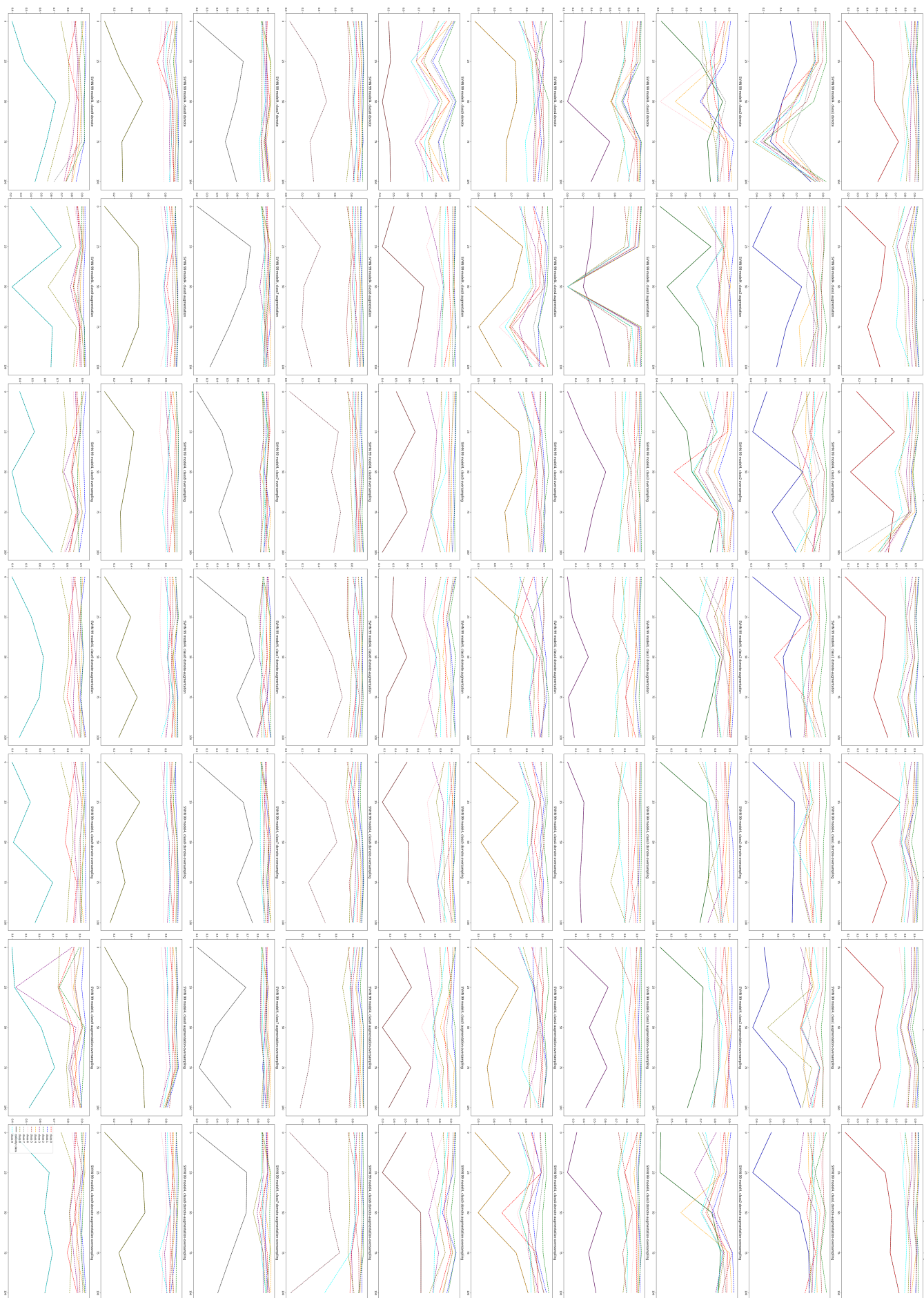




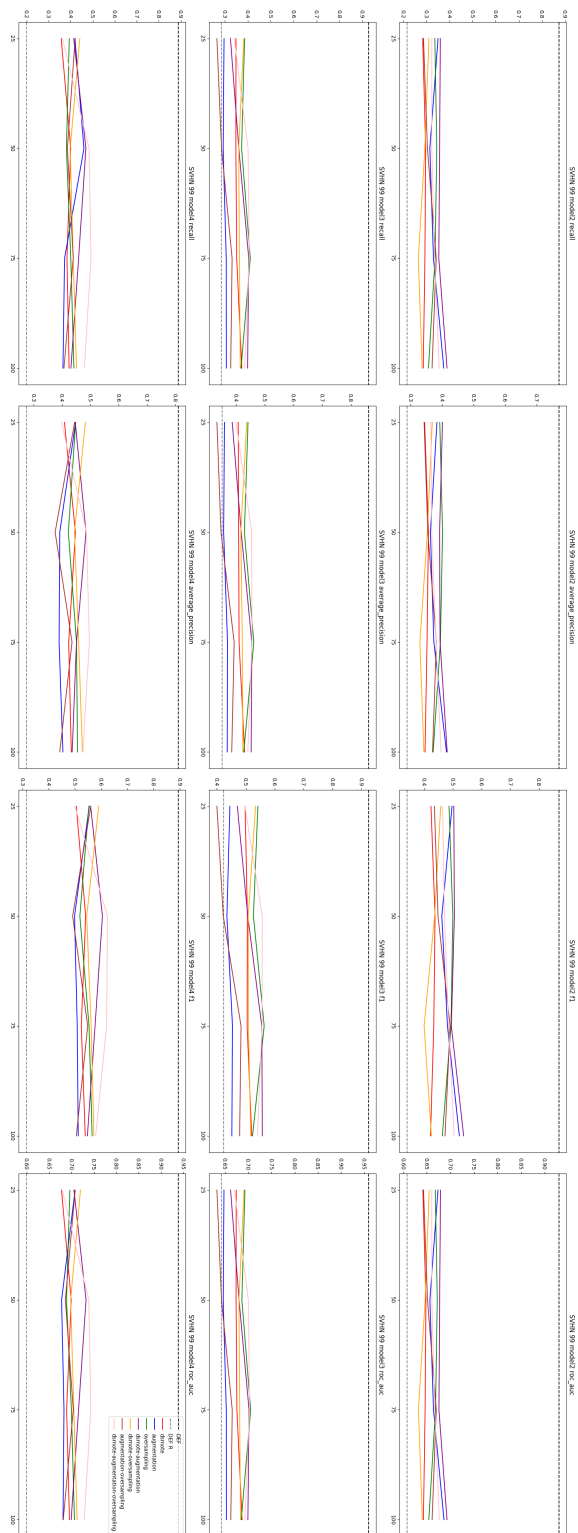
46 pav. SVHN -99% Modelis 2 visų klasių F1 rezultatai



47 pav. SVHN -99% Modelis 3 visų klasių F1 rezultatai



48 pav. SVHN -99% Modelis 4 visų klasių F1 rezultatai



49 pav. SVHN -99% rezultatų vidurkiai