

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

Magistro baigiamasis darbas

# Elektroencefalogramų analizė konvoliuciniais neuroniniais

## tinklais

Analysis of Electroencephalograms Using Convolutional Neural Networks

Atliko: Monika Kisieliūtė

parašas

Vadovas: prof. dr. Tadas Meškauskas

Vilnius 2025

# Turinys

Su	tartir	nis terminų žodynas	4
Sa	ntrau	ka	5
Su	mma	ry	6
Įva	adas		7
1.	Susį	jusių darbų apžvalga	10
2.	Elek	troencefalogramos ir miego fazių specifika	12
	2.1.	Polisomnografijos: duomenų rinkimas bei žymėjimas	12
	2.2.	Miego fazės	12
	2.3.	Miegas ir amžius	13
3.	Laik	to – dažnio srities signalų reprezentacija	13
	3.1.	Vilnelės transformacija	13
	3.2.	Signalo skalograma	14
4.	Neu	roninių tinklų taikymas miego fazių klasifikacijai	14
	4.1.	Neuroniniai tinklai	14
		4.1.1. Konvoliuciniai neuroniniai tinklai	15
		4.1.2. ResNet tipo architektūros	16
		4.1.3. SqueezeNet architektūra	16
	4.2.	Neuroninių tinklų mokymas	17
		4.2.1. Persimokymas ir nepakankamas mokymasis	18
		4.2.2. Iš anksto apmokyti modeliai bei kontrastinis mokymas	19
		4.2.3. SimCLR karkasas	19
	4.3.	Duomenų augmentacijos	20
		4.3.1. Dažniausiai taikomos duomenų augmentacijos	20
		4.3.2. Skalogramų augmentacijų problematika	21
	4.4.	Miego fazių klasifikavimo ypatumai	22
		4.4.1. Kryžminės validacijos poreikis	22
		4.4.2. Klasifikavimo vertinimo metrikos	23
		4.4.3. Modelių paaiškinamumas GradCAM metodu	24
5.	Eksp	perimentų įgyvendinimas	25
	5.1.	Naudotos bibliotekos bei programavimo aplinka	25
	5.2.		26
		5.2.1. Duomenų skaidymas pagal amžiaus grupes	26
		5.2.2. Skalogramų sudarymas	27
	5.2	5.2.5. 5 grupių kryzminės validacijos taikymas	28
	5.3.		29
	э.4.	Konvolluciniai modellai ir apmokymo proceduros	50 21
		5.4.1. Bazimiai modellai	31 21
		5.4.2. Is anksto apmokyti modellal naudojant SIMCLR	51

		5.4.3.	GradCAM tyrimas	32	
6.	Rezu	ıltatai		33	
	6.1.	Bazini	ų modelių palyginimas	33	
		6.1.1.	Rezultatai su Sleep–EDF–20 duomenimis	34	
		6.1.2.	Rezultatai su Sleep–EDF–60 duomenimis	36	
		6.1.3.	Rezultatai su Sleep–EDF–80 duomenimis	37	
		6.1.4.	Rezultatai su Sleep–EDF–100 duomenimis	39	
	6.2.	SimCL	R bei augmentacijų taikymo rezultatai	41	
		6.2.1.	Bendri įverčiai	41	
		6.2.2.	Įverčiai kiekvienai miego fazei	42	
	6.3.	GradC	AM taikymas	44	
Išv	ados	ir reko	mendacijos	47	
At	eities	tyrimų	planas	49	
Lit	eratū	iros šalt	iniai	50	
Pr	iedai			54	
A.	EEG	Fpz-C	z kanalo fragmentai skirtingų miego fazių metu	55	
B.	Bazi	nių mo	delių metrikų medianų įverčiai	56	
C.	C. SimCLR bei augmentacijų scenarijų metrikų medianų įverčiai				
D.	Papi	ldomi (	FradCAM aktyvacijų žemėlapiai	63	

# Sutartinis terminų žodynas

- 1. EEG elektroencefalograma.
- 2. KNT konvoliucinis neuroninis tinklas.
- 3. VT vilnelių transformacija.
- Bazinis modelis toks KNT modelis, kurį apmokant nebuvo naudojamos duomenų augmentacijos ar išankstinis apmokymas.
- 5. ResNet<sub>1</sub> bazinis ResNet–18 architektūros modelis.
- ResNet<sub>2</sub> ResNet–18 architektūros modelis, kurį apmokant naudojamos augmentacijos, bet ne išankstinis mokymas.
- SimCLR<sub>1</sub> ResNet–18 architektūros pagrindo modelis, apmokytas su SimCLR kontrastinio mokymo procedūra (nenaudojant papildomų augmentacijų).
- 8. SimCLR<sub>2</sub> ResNet–18 architektūros pagrindo modelis, apmokytas su SimCLR kontrastinio mokymo procedūra (naudojamos papildomos augmentacijos).
- 9. Aktyvacijų žemėlapiai GradCAM metodu gautos paveiksliuko vietos, teigiamai veikiančios prognozę.
- 10. Kontrafaktiniai aktyvacijų žemėlapiai GradCAM metodu gautos paveiksliuko vietos, neigiamai veikiančios prognozę.

Kai kurios čia bei šiame rašto darbe esančios sąvokos verstos remiantis [41] žodynu. Papildomai, stengtasi laikytis lietuviškų vertimų, tačiau kai kur paprastumo dėlei naudotasi anglų kalbos trumpiniais (pvz.: Std. – standartinis nuokrypis, MAD – absoliutus medianos nuokrypis, TP – teisingai suklasifikuoti teigiami atvejai).

### Santrauka

Šiame darbe analizuojamas miego fazių klasifikavimas konvoliuciniais neuroniniais tinklais naudojant tik vieno EEG kanalo skalogramas. Lyginamos Morse, kompleksinės Morlet, kompleksinės Gauso bei Mexican Hat vilnelėmis sudarytos skalogramos bei dvi KNT architektūros (SqueezeNet bei ResNet–18). Rezultatams gerinti, parinkus geresnę modelio bei vilnelės kombinaciją, pritaikomas kontrastinio mokymo SimCLR karkasas bei duomenų augmentacijos. Siekiant prisidėti prie modelių paaiškinamumo, įgyvendintas GradCAM metodas, pavaizduoti KNT sprendimui aktualūs skalogramų regionai. Randama, jog miego fazių klasifikavimas labai priklauso nuo pacientų amžiaus: geriausiai klasifikuojami jauniausių, blogiausiai – vyriausių pacientų duomenys. Rasta, jog Morse bei kompleksinė Morlet vilnelės tam tikrais atvejais yra pranašesnės už kitas, ResNet–18 architektūra stabilesnė. Papildomai, kontrastinis mokymas pagerina REM miego fazės klasifikavimą beveik visoms amžiaus grupėms. Pagerintas ir I miego fazės klasifikavimo pablogėjimas. Tai galima sieti su nekorektiškomis augmentacijomis būtent šiai amžiaus grupei. Šie radiniai nauji, be to, jie paryškina, jog klasifikuojant miego fazes svarbu atsižvelgti į pacientų amžiaus statistikas.

### **Summary**

#### Analysis of Electroencephalograms Using Convolutional Neural Networks

Sleep staging is a time–consuming and resource–intensive process. Automating this task could significantly enhance clinicians' productivity, allowing them to focus on diagnosing and treating patients. Classifying complex, non–stationary signals is inherently challenging, and convolutional neural networks (CNNs) are widely used for this purpose. Recent research has shifted focus toward classifying the scalogram of EEG signals, rather than the raw signal itself. Wavelet transformation is particularly relevant for constructing these scalograms, as it enables the representation of the signal in both time and frequency domains. However, there has been limited research on how the choice of mother wavelet affects classification performance.

Additionally, contrastive learning frameworks and data augmentation techniques hold potential for improving predictions of sleep stages, but no studies have applied these methods to scalogram data for sleep stage classification. This study aims to examine how the choice of wavelet influences sleep stage classification. Furthermore, the SimCLR contrastive learning framework was implemented, simple data augmentation techniques were tested, and results were compared with a baseline model. Notably, this study uses data from a single electroencephalogram (EEG) channel, aligning with the goal of developing portable, cost–effective polysomnographic devices for home diagnostics. Additionally, the study compares classification results across four distinct age groups, highlighting the impact of age on classification performance.

Based on a comprehensive review of relevant research, the following wavelets were selected: Morse, complex Morlet, complex Gaussian, and Mexican Hat. Two CNN architectures, Squeeze-Net and ResNet–18, were evaluated to determine the optimal baseline model. The combination of ResNet-18 and the complex Morlet wavelet was chosen for further improvement using the SimCLR pretraining framework and data augmentation. To enhance interpretability, the GradCAM method was applied to visualize the scalogram regions influencing the model's decisions.

The study found that sleep stage classification metrics varied with patient age, the youngest patients achieving the highest classification metrics and the oldest patients the lowest. In specific cases, using Morse and/or complex Morlet wavelet showed improved metrics compared to other wavelets. Additionally, ResNet–18 architecture has been shown to be more stable. Contrastive learning proved beneficial, especially for classifying REM sleep stage across most age groups, except for the oldest patients. This discrepancy may be attributed to the inappropriate selection of data augmentation techniques for this age group. Additionally, SimCLR with augmentations enhanced the classification of sleep stage I, but only for one specific age group. These findings underscore the importance of considering patient age when improving sleep stage classification methods, particularly in developing more accurate and personalized diagnostic tools.

### Įvadas

Miegas – viena svarbiausių žmogaus biologinių funkcijų. Esant miego sutrikimams ar tiesiog suprastėjus jo kokybei, sutrinka žmogaus gyvenimo kokybė, tačiau klinikinė diagnostika pacientams ne visada yra prieinama ar patraukli. Fiziologinių rodiklių stebėjimas miego metu (kitaip – polisomnografijų rinkimas) – labiausiai patikimas diagnostinis metodas. Tam tikri miego įvykiai bei dėsningumai matomi tik elektroencefalogramose (toliau – EEG), todėl yra poreikis neapsiriboti vien elektrokardiogramų ar kitų lengviau surenkamų duomenų analize.

Kita vertus, miego EEG duomenų rinkimas ganėtinai sudėtingas procesas, jis nepatogus tiek pacientui, tiek medikams. Dažniausiai pacientas turi atsigulti į ligoninę ir ten yra stebimas viso miego metu. Dar daugiau, netgi tokia diagnostika ne visada yra prieinama dėl ilgų eilių. Iš kitos pusės, medicinos personalas taip pat yra apkraunamas – EEG duomenis reikia analizuoti. Net pirminė analizė (miego fazių sužymėjimas) reikalauja ganėtinai daug laiko bei kruopštumo – miego EEG įrašai ilgi, o miego fazes rekomenduojama žymėti kas 30 sekundžių, įvertinant dominuojančią miego fazę bei atsižvelgiant į prieš tai buvusias fazes [22].

Kaip alternatyva siūlomi mobilūs EEG prietaisai, su kuriais galima diagnostika namuose, jie galimai išspręstų problemas, su kuriomis susiduria pacientai – namuose atliekami tyrimai patogesni bei prieinamesni. Tačiau tokie prietaisai vis dar nėra tokie patrauklūs, kokie galėtų būti. Juose matuojama daug EEG kanalų, be to, dažnai reikalinga ir kitų signalų informacija. Tai reiškia, jog pacientas turi būti apmokytas kaip užsidėti kiekvieną EEG elektrodą, o tai nėra labai paprasta.

Vienas iš galimų sprendimų – mažinti EEG matuojamų kanalų kiekį. Tai ne tik padarytų diagnostiką prieinamesnę ir patogesnę, tačiau ir palengvintų tokių prietaisų tobulinimą. Taip pat, yra poreikis tokius duomenis apdoroti automatizuotai (arba bent jau dalį analizės atlikti automatizuotai). Kadangi viena pirminių užduočių, kurią atlieka medikai yra būtent miego fazių nustatymas, verta panagrinėti algoritmus, kurie klasifikuoja miego fazes būtent vieno kanalo EEG duomenyse.

Vienas didžiausių automatinio miego prognozavimo trūkumų yra tai, jog taikant įvairius klasifikavimo algoritmus, yra labai mažai atsižvelgiama į pacientų amžių. Konkrečiai, dėmesys yra daugiausiai skiriamas arba klasifikuojant vaikų miego fazes, arba suaugusiųjų, tačiau suaugusiųjų duomenys nėra papildomai skaidomi į skirtingas amžiaus grupes. Tai šiek tiek stebina, nes pastebima, jog medicininėje literatūroje fiksuojami miego skirtumai ir skirtingose suaugusiųjų amžiaus kategorijose. Dar daugiau, randama, jog tam tikri miego sutrikimai bei miego kokybės suprastėjimai dažniau kamuoja vyresnio amžiaus asmenis [16], todėl jiems yra itin aktuali prieinama diagnostika.

Apskritai, pastaruoju metu sudėtingų signalų analizei taikomi konvoliuciniai neuroniniai tinklai, su jais galima atlikti miego fazių klasifikavimą net naudojant vieno kanalo duomenis. Papildomai, daugėja tyrimų, kuomet signalas yra transformuojamas į skalogramą ir toks paveiksliukas naudojamas kaip konvoliucinių neuroninių tinklų įvestis. Skalogramoms sudaryti naudojama vilnelės transformacija, o ją atliekant galima naudoti įvairias vilneles. Deja, atlikus tyrimų analizę, pasimatė, jog retai lyginama klasifikacija naudojant skirtingas vilnelės funkcijas. Todėl verta išsamiau panagrinėti, ar vilnelės parinkimas yra svarbus prognozuojant miego fazes konvoliuciniais neuroniniais tinklais. Taip pat, reikalingas ir konvoliucinių neuroninių tinklų architektūrų palyginimas, nes kai kada randami prieštaringi rezultatai.

Klasifikavimo rezultatams gerinti autorės akį pratraukė kontrastinio mokymo metodikos. Tai – specifinis neuroninių tinklų mokymo būdas, kuomet yra siekiama modelį mokinti dvejomis stadijomis, iš kurių viena yra neprižiūrimo mokymo stadija. Kitaip tariant, prieš mokinant klasifikuoti miego fazes, modelis yra mokomas išgauti reikšmingas reprezentacijas ir tik tada mokinama klasifikavimo užduočiai. Viena iš tokių mokymo schemų, SimCLR, ir buvo naudojama šiame darbe. Kertinė SimCLR idėja – duomenų augmentacijų naudojimas. Apskritai, duomenų augmentacijos kaip duomenų diversifikavimo/rinkinio praplėtimo strategija naudojama daug kur, SimCLR karkasas integruoja jas būtent neprižiūrimo mokymo stadijoje. Nors kontrastinio mokymo metodai taikyti klasifikuojant miego fazes [24, 30], autorė nerado, jog šis metodas būtų taikytas su VT transformuotais EEG duomenimis. Apskritai, kontrastinio mokymo naudojimas aktualus ir dėl to, jog neprižiūrimo mokymo metu nereikia turėti sužymėtų duomenų, todėl ateityje taikant bei tobulinant šią strategiją, užtektų surinkti daugiau duomenų ir sužymėti tik mažą jų dalį.

Kritikuojama giliojo mokymosi metodų vieta yra modelių interpretuojamumas, tas taikoma ir konvoliuciniams neuroniniams tinklams. Konkrečiai, nėra visada iki galo suprantama, kodėl neuroninis tinklas priima galutinę prognozę. Tam išspręsti sukurtas GradCAM metodas, kurio pagalba galima vizualiai paaiškinti, kurios paveikslėlio vietos labiausiai prisideda prie sprendimo. Kadangi šiame darbe naudojamos skalogramos, kurios reprezentuoja signalą dažnių–laiko srityje, GradCAM metodo pagalba galima išsiaiškinti, kuris dažnių diapazonas yra reikšmingas klasifikacijai bei kuris, priešingai, netgi lemia neigiamą įtaką. Šios įžvalgos gali būti naudingos ne tik informatikams, tačiau ir medicinos ekspertams.

Taigi, galima labai apibendrinti šio darbo **tyrimo tikslus** – šiuo magistro darbu siekta patikrinti, kaip miego fazes klasifikuoja konvoliuciniai neuroniniai tinklai, apmokyti su skirtingomis EEG skalogramomis bei skirtingų amžiaus grupių duomenimis. Kadangi siekiama išbandyti ir galimus KNT tobulinimus, atrinkta geriausia vilnelės bei modelio kombinacija tobulinta naudojant SimCLR mokymo metodą bei taikant duomenų augmentacijas. Darbe naudojama Sleep-EDFx duomenų bazė, kuri suskaidyta į keturis duomenų poaibius pagal pacientų amžių (išsamus poaibių aprašymas pateiktas 5.2.1), apsiribota tik EEG Fpz-Cz kanalo duomenimis.

Keliami tokie tyrimo uždaviniai:

- 1. Įvertinti žemiau taikomų metodų poveikį klasifikuojant skirtingų amžiaus grupių suaugusių asmenų miego fazes.
- 2. Palyginti miego fazių klasifikavimą taikant ResNet–18 bei SqueezeNet architektūras.
- 3. Miego fazių EEG transformuojant į skalogramas, jas sudaryti naudojant: Morse, kompleksinę Morlet, kompleksinę Gauso bei Mexican Hat vilneles.
- 4. Išsirinkus vieną vilnelės ir modelio kombinaciją, palyginti miego fazių klasifikaciją taikant duomenų augmentacijas bei SimCLR kontrastinio mokymosi karkasą.
- GradCAM metodo pagalba išskirti aktyvacijų žemėlapius ir vizualiai įvertinti, ar yra pastebimų tendencijų.

Bendriems klasifikavimo rezultatams vertinti paskaičiuotos makro preciziškumo, jautrumo, F1 metrikos bei Koheno kapa, atskirų miego fazių klasifikavimui – F1 metrika. Taip pat, eksperimentų paklaidai įvertinti naudota 5–grupių kryžminė validacija, rezultatai laikomi statistiškai reikšmingais pagal 1 standartinio nuokrypio taisyklę. Trumpai apibendrinus rezultatus, pastebima, jog geriausiai prognozuojami jauniausių asmenų duomenys, bendros klasifikacijos metrikos didėjant amžiui prastėja, o itin prastai prognozuojami vyriausių pacientų duomenys. Taip pat rasta, jog kai kuriais atvejais pranašiausios buvo dvi vilnelės – Morse (kuri tyrimuose taikoma dažniausiai) bei kompleksinė Morlet (kuri priešingai, tyrimuose taikoma daug rečiau). Nors duomenų augmentacijų bei SimCLR kontrastinio mokymo karkaso taikymas negerina bendrų klasifikavimo metrikų (išskyrus makro jautrumą), tačiau pagerina REM miego fazės klasifikavimą beveik visoms amžiaus

grupėms. Rasta svarbi išimtis – patys vyriausi pacientai, kuriems taikytos strategijos ne tik nepagerino REM klasifikacijos, bet netgi pablogino atsibudimo miego fazės F1, ką galima sieti su parinktų augmentacijų korektiškumu būtent šiai amžiaus grupei (pablogėjimas stebimas ir taikant tik augmentacijas be SimCLR metodo). Taip pat, pastebimas ir I miego fazės F1 įverčio pagerėjimas taikant SimCLR su papildomomis augmentacijomis 66–74m. amžiaus asmenims. Šie rezultatai tik paryškina, jog itin svarbu kreipti dėmesį į pacientų amžiaus statistikas.

Skaitytojui gali būti paranku žinoti bendrą šio rašto darbo strukūrą: 1 skyriuje atliekama susijusių darbų apžvalga, 2 skyriuje aprašoma su miego fazių EEG signalais susijusi specifika. 3 skyrius skirtas signalų reprezentacijai laiko – dažnių srityje, visas 4 skyrius išsamiai aptaria neuroninių tinklų metodikas bei jų taikymą miego fazių klasifikavimo kontekste. 5 bei 6 skyriai apima praktinę dalį (5 aprašomas eksperimentų įgyvendinimas, o 6 – gauti rezultatai).

Kadangi šis magistro darbas yra tęstinis, dalis skyrių paimti iš mokslo tiriamojo darbo (toliau – MTD [26]) bei patobulinti. Tiksliau, iš MTD paimti bei patobulinti įvado,1, 2 bei 3 skyriai, taip pat dalis poskyrių iš 4 yra papildyti, tačiau dalis – visiškai nauji. Papildomai, kadangi praktinė dalis pradėta vykdyti MTD metu, dalis 5 skyriaus poskyrių taip pat papildyti ar patobulinti. Kai kurios iliustracijos (priedas A bei 8 pav.) taip pat paimtos iš MTD.

### 1. Susijusių darbų apžvalga

Miego fazių klasifikavimui vis dažniau naudojami konvoliuciniai neuroniniai tinklai (toliau – KNT). Dar daugiau, dažnai tyrimuose naudojamas ne pats signalas, o jo skalograma. Tai – paveiksliukas, sudarytas taikant vilnelės transformaciją bei pavaizdavus gautų koeficientų absoliučias vertes laiko–dažnių srityje. Vertinama, jog toks duomenų paruošimo būdas išsaugo specifines signalo savybes tiek laiko, tiek dažnių atžvilgiu nestacionariuose signaluose [1]. Dar daugiau, [17] randa, jog toks apdorojimo metodas yra pranašiausias, kuomet miego fazių klasifikacijai yra naudojama KNT bei apsiribojama tik vienu kanalu.

Nėra labai išsamiai nagrinėjama, ar specifinės vilnelės pasirinkimas sudarant skalogramas turi įtakos miego fazių klasifikavimo užduotyse. Šio tyrimo autorei pavyko rasti tik vieną tyrimą [23], kuriame tikrinama, ar skirtingos vilnelės keičia KNT metrikas (tikrintos Morlet, Morse bei Bump vilnelės) klasifikuojant miego fazes. Rasta, jog mažiausias makro jautrumas buvo taikant Morlet vilnelę (0.72), taip pat su šia vilnele stebima ir mažiausia I miego fazės jautrumo statistika (0.16). Tuo tarpu kitos vilnelės turėjo aukštesnius tiek makro, tiek I miego fazės jautrumo įverčius. Kitaip tariant, gali būti, jog vilnelės skiriasi savo pranašumu.

Trumpai apibendrinus kitus tyrimus, kuriuose naudojami neuroniniai tinklai bei būtent tokio tipo transformacijos siekiant suklasifikuoti miego fazes, pastebėtina, jog dažniausiai taikoma Morse vilnelė [12, 17, 27, 49]. Apskritai, šiuose tyrimuose taikoma ganėtinai daug įvairių strategijų (įvairių augmentacijų naudojimas, iš anksto apmokyti neuroniniai tinklai, skirtingos bazinės architektūros), todėl itin sunku įvertinti, kiek gerus rezultatus lemia vilnelės parinkimas.

[13] tyrime naudota kompleksinė Morlet vilnelė ir įdomu tai, jog pasiektas gana aukštas 0.64 jautrumas pirmai miego fazei klasifikuoti. Deja, vėl yra gana sunku vertinti, kiek įtakos turi pasirinkta būtent tokia vilnelė, nes tyrime taikyta keli metodai: taikytas iš anksto apmokytas AlexNet architektūros neuroninis tinklas, taip pat taikytas dažnių filtravimas. Verta paminėti ir [9] tyrimą, kuriame nebuvo klasifikuojamos specifinės miego fazės, tačiau buvo siekta patikrinti, ar skirtingos vilnelių transformacijos turi įtaką atskiriant ar žmogus miega. Rasta, jog vis tik kompleksinė Morlet bei kompleksinė Gauso vilnelės buvo pranašesnės už kitas.

Atsižvelgus į šiuos tyrimus šiame tyrime taikant vilnelių transformaciją pasirinkta naudoti tokias vilneles:

- (a.) Morse vilnelę (taikyta [12, 17, 23, 27, 49]). Taikoma dažniausiai.
- (b.) kompleksinę Morlet vilnelę (taikyta [9, 13])
- (c.) Kompleksinę Gauso vilnelę (taikyta [9])
- (d.) Mexican Hat vilnelę (nėra taikyta)

Klasifikuojant miego fazes įvairiais metodais, visai perspektyviai atrodo kontrastinio mokymo strategijų bei duomenų augmentacijų naudojimas. Pavyzdžiui, [27] taikė paprastas duomenų augmentacijas (pasukimą, apvertimą, apkarpymą, atspindį). Randamas labai nežymus pagerėjimas visoms metrikoms. Tačiau duomenų augmentacijas galima integruoti ne tik į įprastinio mokymo procesą, tačiau ir naudojant kontrastinio mokymo metodikas. Kontrastinis mokymas yra naudotas miego fazių prognozavimo užduotyje, tačiau autorė aptiko tik tyrimus, kurie naudoja neapdoroto signalo duomenis. Pastebima, jog naudojant kontrastinį mokymą pasiekiami neblogi rezultatai. Pavyzdžiui, [30] naudojo kontrastinį mokymą ir pasiekė 0.79 makro F1 Sleep–EDFx duomenų bazei. Dar daugiau, randama jog pagerėja probleminių (tai yra, I bei REM miego fazės) miego fazių klasifikacija lyginant su prižiūrimo mokymo metodais. Kita vertus, [24] taip pat randa geresnius rezultatus taikant SimCLR tipo kontrastinio mokymo karkasą su neapdorotais EEG signalais, tačiau tik taikant ant visos Sleep–EDFx duomenų bazės, bet su jos poaibiu (Sleep–EDF–20) randamas makro F1 pablogėjimas. Kitaip tariant, rezultatai taikant įvarius kontrastinio mokymo algoritmus nevienareikšmiški, tačiau galimai perspektyvūs.

Vis tik, autorė nerado, jog kontrastinio mokymo strategijos būtų taikytos klasifikuojant miego fazių skalogramas. Kadangi taikant prižiūrimo mokymo strategijas klasifikuojant miego fazių skalogramas gaunami pakankamai pakankamai neblogi rezultatai, kyla klausimas, ar kontrastinio mokymo naudojimas jų nepagerintų dar labiau. Būtent dėl to šiame tyrime taip pat siekiama patikrinti, ar kontrastinio mokymo naudojimas (taikant SimCLR mokymo karkasą [5]) naudingas būtent su tokiais duomenimis. Šiam karkasui būtina pasirinkti fundamentinį (angl. *backbone*) neuroninį tinklą, todėl pirma nuspręsta palyginti dvi tyrimuose naudotas bazines KNT architektūras:

- SqueezeNet (taikyta [23, 27]) randami prieštaringi rezultatai lyginant šiuos du tyrimus. Kuomet naudojama ta pati vilnelės transformacija (Morse), [23] tyrime su šia architektūra pasiekia 0.75 makro F1, o štai [27] randa 0.69 makro F1.
- ResNet–18 (taikyta [27]) randama, jog ši architektūra pasiekia geriausius arba panašius F1 įverčius lyginant su didesnėmis ResNet architektūromis. Kitaip tariant, nėra reikalo mokyti didesnę ResNet architektūrą.

Kol kas dar neaptarta tai, jog apskritai miego fazių klasifikavimo tyrimuose dažnas modelių šališkumas, tai yra, modeliai mokinami bei vertinami arba ant pakankamai jaunų asmenų (tarkime, [27]) arba ant vidutinio amžiaus asmenų (pvz., ISRUC-Sleep bazė, naudota [49]). [3] irgi sutinka su teiginiu, jog miego fazių klasifikavimo tyrimai yra šališki vidurinio amžiaus žmonėms ir pastebi tai, jog kuomet modeliai apmokomi su kitokio amžiaus duomenimis nei tiksliniai duomenys, miego fazių klasifikavimas pablogėja. Tas ypač galioja kuomet klasifikuojamos vaikų miego fazės, tačiau pablogėjimas stebimas ir su garbaus amžiaus žmonių duomenimis [3]. Taigi, yra poreikis nustatyti, ar skiriasi klasifikacija skirtingoms suaugusiųjų amžiaus grupėms, o ypač svarbu atkreipti dėmesį į vyresnių pacientų miego fazių klasifikaciją.

Kadangi šis tyrimas taip pat siekia apsiriboti vieno signalo EEG duomenimis, svarbu dar kartą paminėti [23] tyrimą, kuriame miego fazių klasifikacija atlikta su keliais kanalais. Šio tyrimo autoriai pastebėjo, jog didesnis klasifikavimo tikslumas pasiekiamas su Fpz-Cz kanalo duomenimis. Šis kanalas apskritai yra dažniausiai naudojamas tyrimuose (kuomet naudojamasi Sleep-EDFx duomenų baze), jis naudotas ir [49]. Todėl ir šiame darbe apsiribojama būtent jo duomenimis.

Neretai medicininiuose tyrimuose siekiama pateikti papildomus pastebėjimus ar analizes, kurios paprastai parodytų, kaip buvo priimamas/priimami konkretūs neuroninio tinklo sprendimai. Ne išimtis – ir miego fazių klasifikavimo tyrimai. Vaizdų analizės srityje yra sukurti algoritmai, kurie padeda netgi vizualiai pabrėžti, kurios paveiksliuko dalys teigiamai bei neigiamai prisideda prie modelio sprendimo priėmimo (tarkime, klasifikuojant vieną ar kitą klasę). Vienas iš tokių metodų, GradCAM [45], taikytas ir dirbant su miego fazių skalogramomis [23]. Vis tik, pačio GradCAM kūrėjai išskiria, jog galima naudoti ši metodą ne tik siekiant paaiškinti, kurie regionai prisideda teigiamai, tačiau kurie netgi pablogina klasifikaciją [45], o šio darbo autorė nerado, jog tokia strategija atliekama tyrimuose. Dar daugiau, kadangi skalogramos yra fiksuoto dydžio paveiksliukai, galima pateikti ir apibendrintą reikšmingų regionų svarbą, kaip tai darė [15] su radiologinėmis nuotraukomis, ir galimai akcentuoti, kurie dažnių regionai aktualūs klasifikuojant miego fazes, o ne tik vieną konkretų miego įrašo fragmentą.

### 2. Elektroencefalogramos ir miego fazių specifika

### 2.1. Polisomnografijos: duomenų rinkimas bei žymėjimas

Polisomnografija – klinikinis įrašo tipas, kuriame pateikiama laboratorijoje užfiksuota įvairi fiziologinė informacija. Tiksliau, žmogui miegant laboratorinėje aplinkoje, fiksuojama elektroencefalograma, elektrookulograma (EOG), smakro elektromiografija (EMG) bei elektrokardiograma (EKG) [22]. Papildomai, rekomenduojama rinkti ir kitokius fiziologinius duomenis (tokius kaip: gulėjimo pozicija, kvėpavimo ritmas) [22], tačiau tai yra tik rekomendacija, o taip pat - specifinė miego fazė nustatoma naudojantis pagrinde EEG, EOG, EMG bei EKG duomenimis [22].

Apskritai, vertintojai, žymėdami kuri miego fazė yra kuri, susiduria su daug sunkumu ir griežtų taisyklių (labai apibendrintai kiekvienos miego fazės vertinimo kriterijai aprašyti 2.2). Medikai turi sužymėti labai ilgus polisomnografinius įrašus, vertinti kelių tipų duomenis.

Gali atsitikti netgi taip, jog viename intervale atsitinka miego fazių pokytis. Tokiu atveju, rekomenduojama žymėti pagal dažniausiai matomą aktyvumą. Tai kelia tam tikrą duomenų problematiką: miego fazių duomenys dėl tokio žymėjimo standarto savaime yra "nešvarūs". Dar daugiau, net jeigu fazė intervale yra viena, yra ganėtinai daug kriterijų, kaip ją konkrečiai identifikuoti: reikia žiūrėti, koks aktyvumas yra, kokio nėra ir taip pat atsižvelgti į prieš tai buvusias fazes [22]. Kitaip tariant, tai itin daug žmogiškųjų resursų reikalaujantis darbas.

Svarbu ir tai, jog patys vertintojai ne visada tarpusavyje sutinka dėl tinkamo vertinimo. Didžiausi nesutarimai dėl miego fazių iš vertintojų pusės kyla būtent I miego fazėje [8]. Tačiau pastebima, jog apskritai miego fazių identifikacijai turi įtaką daug faktorių: [8] išskiria pacientų amžių bei žmogiškuosius veiksnius. Apibendrinant, miego fazių identifikavimas nėra triviali užduotis net ir kvalifikuotam medicinos personalui.

### 2.2. Miego fazės

Pasak Amerikos miego medicinos akademijos (toliau – angl. trump. AASM), miego fazes galima skirstyti pagal tai, koks smegenų aktyvumas dominuoja konkrečiu laiko momentu [22]. Išskiriamos trys miego fazių grupės: budrumo fazė (jos metu žmogus nemiega), trys NREM miego fazės (toliau autorės vadinama I, II bei III miego fazės) bei viena REM miego fazė. 1 lentelėje išskirtos svarbiausios miego fazių savybės, tačiau verta aptarti kiekvieną miego fazę išsamiau.

Miego fazė	Aktyvumas	Dažnis	Pastaba
Atsibudimo	α	8-13 Hz	greiti akių judesiai
I fazė	$\alpha, \theta$	4-13 Hz	
II fazė	K kompleksas, miego verpstės	11-16 Hz	
III fazė	δ	0.5-2 Hz	
REM fazė	$\delta,  heta$	2-6 Hz	greiti akių judesiai

1 lentelė. Miego fazių EEG aktyvumo specifika. Apibendrinanti lentelė sudaryta pagal [22] šaltinį.

Atsibudimo fazė – tai fazė, kurios metu žmogus nemiega. Apskritai, ši fazė pasižymi  $\alpha$  smegenų aktyvumu (tai yra, 8-13 Hz), greitais akių judesiais, tačiau kartais aptinkamas ir  $\beta$  aktyvumas (13-15 Hz). Po šios fazės miego dažniausiai seka I miego fazė. Jos metu taip pat pastebimas įvairus EEG aktyvumas ( $\alpha$ ,  $\theta$  bangos, 4-13 Hz diapazonas), tačiau kiek mažesnė EEG amplitudė. Taip pat

šioje fazėje gali (bet neprivalo) būti ir staigių, vertekso formos bangų. Šioje fazėje akys juda lėtai [22].

II miego fazės skiriamieji bruožai – K kompleksas bei miego verpstės (angl. *sleep spindles*). Miego verpstės – staigus EEG aktyvumo padažnėjimas, trunkantis maždaug 0.5 sekundės (11-16 Hz), o K kompleksai – itin staigus EEG aktyvumo pokytis (pagal [22] - nebūtinas kriterijus). Po šios fazės gali būti atsibudimo, III ar REM miego fazės.

Gilaus miego, III fazė, pasižymi  $\delta$  ritmu, tai yra, ypač žemais (0.5-2 Hz) didelės amplitudės dažniais, tačiau vertinant ar miegas gilus, užtenka, jog žemų dažnių būtų 20%. Autorė taip pat nori atkreipti dėmesį į tai, jog toks gilaus miego priskyrimas galioja nuo 2007 metų, kuomet AASM paskelbė naujus miego fazių žymėjimo kriterijus. Iki tol buvo naudojamas kiek kitoks gilaus miego apibrėžimas, kuomet išskiriamos 2 gilaus miego fazės (vadinamasis R&K žymėjimas, [43]). Esminis šių klasifikacijų skirtumas – kiek  $\delta$  ritmo turi būti, jeigu tokio aktyvumo daugiau, skiriama, jog tai yra itin gilus miegas (IV fazė). Todėl dažnai senesniuose duomenyse galima pamatyti ne 3, o 4 NREM miego fazes, tačiau dabar rekomenduojama laikytis AASM standarto. Kartais šioje miego fazėje taip pat matomos II miego fazėje aptinkamos miego verpstės

Dar neaptarta REM miego fazė pasižymi tuo, jog, kaip ir I miego fazė, turi mišrų EEG aktyvumą ( $\delta$ ,  $\theta$  bangos, 2-6 Hz diapazonas). Papildomai, išskiriama ir specifinė signalo forma – [22] pažymi, jog šios miego fazės metu EEG signalas primena pjūklo formą. Šios miego fazės metu taip pat pastebimi greiti akių judesiai

Kaip atrodo realūs EEG duomenys skirtingų miego fazių metu galima pamatyti priede A.

#### 2.3. Miegas ir amžius

Kaip pastebėta 2.1, svarbus demografinis faktorius ekspertiniam miego fazių žymėjimui yra miegančiojo amžius. Žinoma, jog vaikų miegui vertinimui yra net taikomi skirtingi kriterijai [22], tačiau kadangi šiame darbe orientuojamasi į suaugusių asmenų miego fazių klasifikaciją, verta trumpai aptarti, kokie miego pokyčiai atsiranda žmogui senstant.

Visų pirma, [16] įvardina, jog vyresni žmonės miega daug neramiau nei jauni. Neramus miegas pasireiškia keliais rodikliais: dažnesniais atsibudimais bei padidėjusiu negilių miego fazių (I bei II) skaičiumi. Automatizuotai miego fazių klasifikacijai tai turi bent dvi implikacijas. Visų pirma, pakitusi miego fazių proporcija reiškia, jog kuomet medikai vertina polisomnografinius įrašus, dažnėja mišrių duomenų kiekis (tai yra, duomenų, kuomet pažymėta tik viena miego fazė, tačiau yra dvi), nes miegas yra mažiau stabilus. Pakitusi miego fazių proporcija papildomai sukelia sunkumų jau iš algoritmų pusės, nes tai lemia labiau nesubalansuotus duomenis (miego fazių proporcijų pavyzdį iš Sleep–EDFx duomenų bazės galima pamatyti 5.2 poskyryje, 7 pav.).

Dar daugiau, pastebima ir tai, jog vyresnių asmenų miego fazės kokybiškai taip pat kitokios, o konkrečiai, pastebimi pokyčiai būtent gilaus miego (III) fazėje [16]. Taip pat, randama, jog amžius taip pat turi įtaką ir ekspertiniam vertinimui – randama, jog ekspertai mažiau sutaria dėl II bei III miego fazių [8].

### 3. Laiko – dažnio srities signalų reprezentacija

#### 3.1. Vilnelės transformacija

Kaip aprašyta 2.2 poskyryje, miego fazės iš esmės viena nuo kitos skiriasi dominuojančiais dažniais bei tam tikromis signale esančiomis struktūromis. Kita vertus, atrodo jog tik dažnių srities

analizė ar transformacija yra vis tik nepakankama analizuojant medicininius signalus – ši informacija kinta laike. Kaip vieną iš metodų, kurio pagalba galima išlaikyti dažninę signalo informaciją laike galima įvardinti trumpalaikę Furjė transformaciją, kuomet signalo fragmentas yra suskaidomas į langelius ir tuose intervaluose atliekama Furjė transformacija. Deja, toks dažninės informacijos laike išskyrimas nėra be trūkumų – gerai tiek laiko, tiek dažnių lokalizacijai reikia optimalaus langelio pasirinkimo [1]. Šia problemą sprendžia vilnelės transformacija (angl. *Wavelet transform*, toliau - VT), kuomet laiko langelis yra prisitaikantis. Ši savybė labai naudinga analizuojant nestacionarius signalus.

Apskritai, vilnelė yra bangą primenanti funkcija, kuri gali kisti tiek laiko, tiek skalės atžvilgiu. Kitaip tariant, galima lyginti signalą su daug įvairių ir perstumdytų vilnelės versijų. Vilnelės mastelis iš esmės nusako, kiek originali vilnelės funkcija (angl. *Mother wavelet*) yra suspausta ar ištempta. Skalė yra susijusi su vilnelės poslinkiu: vilnelės skalei esant didelei, vilnelė labai ištempiama, ji reprezentuoja žemus dažnius, tačiau to pasekoje gali nedaug slinktis per signalą. Galioja ir atvirkštinis teiginys – kai skalė maža, vilnelė suspaudžiama, geriau reprezentuojami aukšti dažniai ir galima daryti daugiau poslinkių per signalą, tačiau prasčiau reprezentuojama dažninė informacija. Yra daug įvairių vilnelių funkcijų ir, priklausomai nuo jų parinkimo, galima išryškinti skirtingas signale esančias savybes.

#### 3.2. Signalo skalograma

Palyginus signalą su vilnele skirtingose skalės ir laiko vietose, gaunama vilnelės transformacijos koeficientų matrica. Pavaizdavus šių koeficientų absoliučias reikšmes naudojant spalvų gamą gaunama signalo skalograma. Tokia vizualizacija leidžia aiškiai matyti dažninės informacijos kitimą laike. Apskritai, tokia signalo reprezentacija gali būti naudinga dėl kelių priežasčių. Visų pirma, kadangi signalas pakankamai gerai reprezentuojamas laiko–dažnių srityje, tai supaprastina dalies algoritmų taikymą. Pavyzdžiui, tokia signalo transformacija leidžia taikyti vaizdų analizės algoritmus. Papildomai, kadangi skalograma yra sudaroma taikant VT, ji gali būti pranašesnė už spektrogramas (kurios sudaromos taikant trumpalaikę Furjė transformaciją).

Konstruojant skalogramą, svarbu ne tik VT metu parenkami parametrai (tarkime, motininės vilnelės parinkimas), tačiau ir parinkta spalvų gama bei paveiksliuko mastelis. Nors spalvų gamos parinkimas nėra labai esminis aspektas (reikalinga norint labiau palyginti rezultatus tarp skirtingų tyrimų), gali reikėti į jį reikia atsižvelgti tolimesniuose algoritmų žingsniuose (tarkime, jeigu tai-komos duomenų augmentacijos, o ypač – spalvinės). Taip pat, dažnai medicininiuose tyrimuose naudojamas logaritminis mastelis (tiek su spektrogramomis, tiek su skalogramomis), nes jis geriau atvaizduoja reikšmingus dažnius (beveik visi 1 skyriuje nagrinėti tyrimai naudoja šį mastelį).

### 4. Neuroninių tinklų taikymas miego fazių klasifikacijai

#### 4.1. Neuroniniai tinklai

Miego fazių klasifikavimo užduotyse vis labiau taikomi neuroniniais tinklais grįsti metodai. Nors tai nėra vienintelė tokių duomenų apdorojimo strategija (galima taikyti ir paprastesnius mašininio mokymo algoritmus [31]), esminis jos privalumas - galima daryti ganėtinai mažai prielaidų apie specifines EEG kanalo ar miego fazių specifikas.

Bendrai, neuroninių tinklų pagalba galima aproksimuoti bet kokią funkciją. Tai ypač naudinga savybė analizuojant nestruktūruotus duomenis (tarkime, klasifikuojant tekstą, vaizdus) arba, kaip

yra ir šiame magistro darbe, nestacionarius signalus, reprezentuojamus laiko-dažnių srityje.

Trumpai apibendrinus visiems neuroniniams tinklams būdingas struktūras, galima išskirti, jog bazinis jų komponentas – neuronas, sudarytas iš trijų svarbiausių dalių: svorių bei poslinkio (tai – neuroninio tinklo parametrai) bei aktyvacijos funkcijos. Sudauginus įvestį su svoriais ir susumavus su poslinkiu atliekama tiesinė transformacija, o aktyvacijos funkcijos leidžia atlikti netiesinę transformaciją [32].

Yra begalė neuroninių tinklų architektūrų, kurios tinkamos klasifikuoti miego fazes. Vis tik, šiame darbe apsiribota būtent konvoliuciniais neuroniniais tinklais. Dėl to, verta pasigilinti į tai, kas yra esminės KNT komponentės bei trumpai aptarti šiame darbe naudojamas architektūras.

#### 4.1.1. Konvoliuciniai neuroniniai tinklai

Konvoliuciniai neuroniniai tinklai, lyginant su pilnai sujungtais neuroniniais tinklais, apskritai pasižymi keliomis naudingomis savybėmis [32]: mažesnių parametrų kiekiu, lokaliais ryšiais bei dimensiškumo mažinimu. Šios charakteristikos iš esmės pagerina modelio konvergavimą bei sumažina parametrų kiekį, kas turi reguliarizacinį efektą (tai yra, KNT rečiau persimoko).



1 pav. 2D konvoliucijos schema su papildymu bei sutelkimu, kuomet naudojama maksimali reikšmė. Sukurta autorės, remantis [32].

[32] taip pat išskiria, jog tipiškai KNT turi 4 sudedamąsias komponentes (jų sąveikos schema pavaizduota 1 pav.):

1. Konvoliucijos (filtro) branduolys (angl. *convolution kernel*) – išskiriami požymių žemėlapiai, kai konvoliucinis filtras slenka per įvestį. Žemiau pateikiamas 2D konvoliucijos apibrėžimas:

$$Y_{i,j} = (I * K)_{i,j} = \sum_{m=1}^{M} \sum_{n=1}^{N} I_{i-m+a,j-n+b} K_{m,n}$$
(4.1)

Čia laikoma, jog  $Y_{i,j}$  – pozicijoje (i, j) gaunamas požymių žemėlapis, I – įvestis, K – konvoliucinis filtras (jo dydis:  $M \times N$ ).

- 2. Papildymo transformacija (angl. *padding*) padidinama įvestis tam, jog kraštuose esanti informacija nebūtų prarandama.
- Žingsnio parametrą (angl. *stride*) kontroliuoja požymių žemėlapių tankumą (kuo didesnis šis parametras, tuo požymių žemėlapiai mažesni).
- 4. Sutelkimo sluoksnis (angl *pooling*) pasiekiamas dimensiškumo mažinimas. Būna sutelkimas naudojant vidutinę arba maksimalią reikšmes.

Apskritai, KNT yra gana daug ir įvairių, jos skiriasi ne tik savo architektūra, tačiau ir parametrų kiekiu. Šiame darbe naudojamos bazinės modelių architektūros SqueezeNet bei ResNet–18 modeliai yra gana kompaktiškos (atitinkamai turi 1.2 bei 11.7 milijonų parametrų), tačiau rasta, jog miego fazių klasifikavimo užduotyse jos gali būti neprastesnės už didesnes KNT architektūras [23, 27].

#### 4.1.2. ResNet tipo architektūros

Liekanų neuroninio tinklo (angl. *Residual Neural Network*, trump. ResNet, toliau taip ir vadinama) [19] architektūra – plačiai naudojama įvairiuose taikymuose. Esminis šios architektūros principas – liekanų blokai, kuomet konvoliucinio bloko įvestis sujungiama su išvestimi (pavaizduota 2). Ši architektūra buvo sukurta siekiant pagerinti būtent gilių neuroninių tinklų mokymą: aptikta, jog didinat sluoksnių skaičių neuroniniuose tinkluose, modeliai nebūtinai mokosi geriau dėl nykstančių/sprogstančių gradientų problemos. Dar daugiau, [19] teigia, jog net ir užtikrinus, jog ši problema neatsitinka (naudojant, tarkime, kitokias modelių svorių inicializacijos strategijas), liekanų blokai vis tiek lemia geresnį mokymąsi. Papildomai, autoriai naudoja rinkinio normalizaciją, kas taip pat skatina modelio konvergavimą [19]. Įvairaus gylio bei konfigūracijos ResNet architektūros yra naudotos miego fazių klasifikavimo užduotyje [17, 27].



2 pav. Liekanų bloko iliustracija. Parengta vadovaujantis [19].

#### 4.1.3. SqueezeNet architektūra

SqueezeNet architektūros pagrindinis privalumas – itin mažas parametrų kiekis bei modelio dydis, tačiau palyginti geras tikslumas (vertinant ant *ImageNet* duomenų rinkinio, tačiau ši architektūra plačiai naudojama ir su kitais duomenimis) [21]. Esminė šios architektūros dedamoji komponentė – taip vadinamas *Fire* blokas (pavaizduotas 3 pav.), sudarytas iš:

- Suspaudimo (angl. squeeze) sluoksnio pasižymi itin mažais konvoliucijų branduoliais (1x1). Tokie filtrai parinkti būtent dėl parametrų kiekio mažinimo, nes taip sumažinamas tolimesnės įvesties į išplėtimo sluoksnį kanalų kiekis.
- 2. Išplėtimo (angl. *expand*) sluoksnio jame naudojama ir 3x3 dydžio konvoliuciniai branduoliai, tačiau stengiamasi riboti jų kiekį.

Taip pat šioje architektūroje naudojami sutelkimo sluoksniai (tiek imant maksimalią, tiek vidutinę reikšmę) tarp kai kurių *Fire* blokų [21].



3 pav. Fire bloko iliustracija. Parengta pagal [21].

#### 4.2. Neuroninių tinklų mokymas

Bendrąja prasme, neuroninis tinklas mokosi pamažu koreguodamas savo parametrus (svorius ir poslinkius) taip, jog būtų minimizuota paklaida tarp neuroninio tinklo prognozių ir tikrųjų verčių. Tiesioginio sklidimo (angl. *feedforward*) metu įvesties duomenys pereina per visus neuroninio tinklo sluoksnius ir taip gaunamos išvestys. Paklaidoms skaičiuoti naudojama nuostolių funkcija, kuri naudojama atgalinės sklaidos algoritme (angl. *backpropagation*) paskaičiuoti gradientus pagal neuroninio tinklo parametrus. Dažniausiai neuroninio tinklo parametrai optimizacijos algoritmu koreguojami taip, kad nuostolių funkcija būtų kuo mažesnė. Šis procesas vyksta iteratyviai, modeliai mokomi duomenų rinkiniais (angl. *batches*) tol, kol idealiu atveju pasiekiama optimali neuroninio tinklo parametrų konfigūracija. Deja, realybėje ją rasti yra sunku, dažnai susiduriama su persimokymo bei nepakankamo išmokimo reiškiniais (žr. 4.2.1 poskyrį),

Nuostolių funkcija glaudžiai susijusi su tai, kokią užduotį atlieka neuroninis tinklas. Klasifikuojant miego fazes (kadangi tai yra daugelio klasių prognozavimo užduotis), tinka naudoti kryžminės entropijos nuostolių funkciją:

$$\mathcal{L}_{C\mathcal{E}}(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log(\hat{y}_{i,c})$$
(4.2)

Čia laikoma, jog N — pavyzdžių kiekis, c — konkreti klasė, C – bendras klasių kiekis,  $y_{i,c}$  – tikroji klasė (1, jei pavyzdys *i* priklauso klasei c, ir 0 kitu atveju), o  $\hat{y}_{i,c}$  – modelio prognozuota tikimybė, kad pavyzdys *i* priklauso klasei c.

Kadangi ši funkcija reikalauja tikimybės, jog atvejis priklauso konkrečiai klasei, dažnai neuroninių tinklų išvestyje taikoma vadinamoji *softmax* aktyvacijos funkcija:

$$\hat{y}_{i,c} = \frac{e^{z_{i,c}}}{\sum_{j=1}^{C} e^{z_{i,j}}}$$
(4.3)

kur laikoma, jog:  $\hat{y}_{i,c}$  netekties funkcijoje naudojama klasės tikimybė,  $z_{i,c}$  – paskutiniame neuroninio tinklo sluoksnyje gaunamos išvestys, vadinamieji logitai, klasei *c*, atvejui *i*.

Vis tik verta trumpai pažymėti, jog nuostolių funkcijos parinkimas būtent šioje užduotyje nėra vienareikšmiškas dalykas, tyrimuose aptinkamos skirtingos nuostolių funkcijos.

#### 4.2.1. Persimokymas ir nepakankamas mokymasis

Mokant neuroninius tinklus, visada susiduriama su bent viena iš dviejų problemų: modelių persimokymu (angl. *overfitting*) arba nepakankamu išmokimu (angl. *underfitting*). Bendrąja prasme, modelis nepakankamai apsimokina, kuomet nesugeba gerai išmokti mokymo duomenų. Modelis persimokina, kuomet nesugeba generalizuoti, tai yra, kai su mokymo duomenimis prognozė yra gera, tačiau su validacijos duomenimis – nelabai [50]. Spręsti nepakankamą išmokimą bei persimokymą [50] siūloma:

- 1. Ankstyvas sustojimas (angl. *early stopping*) mokymas stabdomas, jei modelio metrikos negerėja po pasirinktų epochų kiekio. Taip siekiama išvengti persimokymo.
- Mokymo duomenų aibės padidinimas surinkus daugiau duomenų (tas galioja ypač jeigu turima mažai duomenų) modelis galimai geriau mokysis (sprendžiamas nepakankamas išmokimas). Papildomai, duomenų aibę galima dirbtinai padidinti taikant duomenų augmentacijas (plačiau aprašoma 4.3) ir taip išvengti persimokymo [2].
- 3. Modelį reguliarizuoti sprendžiama persimokymo problema. Pasak [50], tai daroma arba atrenkant reikšmingus požymius, arba sumažinant neinformatyvių požymių įtaką.

Verta trumpai aptarti modelio reguliarizaciją, nes tai – labai platus strategijų rinkinys. Vienas iš būdų reguliarizuoti modelius – limituoti parametrų kiekį. Dažnai naudojamas atsitiktinio praretinimo transformacijos parametras (angl. *dropout*), kuomet mokymo metu atsitiktinai pasirenkama, kurie modelio vienetai (pvz., neuronai) turės įtaką sprendimui. Taip siekiama sumažinti modelio priklausomybę nuo konkrečių vienetų ir pagerinti jo gebėjimą generalizuoti. Po mokymo, vertinant modelį, visi vienetai yra naudojami sprendimui priimti, o atsitiktinis praretinimas nebetaikomas [50].

Papildomai, dažnai sutinkama ir L1 bei L2 reguliarizacijos, kurių tikslas – kontroliuoti modelių svorius pridedant baudą prie nuostolių funkcijos. L1 reguliarizacija siekiama, jog išliktų tik informatyviausi neuroninio tinklo vienetai. Tai pasiekiama pridedant baudos termą prie nuostolių funkcijos, kuris yra proporcingas svorių absoliučių reikšmių sumai:

$$\mathcal{L} = \mathcal{L} + \lambda \sum_{i=1}^{N} |\theta_i|$$
(4.4)

L2 reguliarizacijos atveju, svoriai baudžiami proporcingai jų kvadratų sumai, ko pasekoje mažesni svoriai tampa stabilesni, o modelis – labiau generalizuojantis:

$$\mathcal{L} = \mathcal{L} + \lambda \sum_{i=1}^{N} \theta_i^2$$
(4.5)

4.4 bei 4.5 laikoma, jog  $\mathcal{L}$  – nuostolių funkcija, N — pavyzdžių kiekis,  $\lambda$  – reguliarizacijos koeficientas,  $\theta$  – neuroninio tinklo parametrai. Abu šie būdai padeda išvengti persimokymo, tačiau L1 reguliarizacija dažniau naudojama požymių atrankai, o L2 – bendram modelio paprastumui ir stabilumui užtikrinti [50].

Taip pat verta paminėti ir papildomus iššūkius, su kuriais susiduriama, kai duomenys yra nebalansuoti, tai yra, kuomet vienų klasių yra ženkliai daugiau nei kitų. Pastebima, jog naudojant įprastą kryžminės entropijos netekties funkciją, modeliai dažnai išmoksta stipriai reprezentuotas klases, tačiau mažiau reprezentuotos yra labiau ignoruojamos mokymo metu [44]. Nors nebalansuotų duomenų prognozei gerinti siūloma daug įvairių metodų, vienas iš paprasčiausių – naudojant kryžminės entropijos nuostolių funkciją, kiekvienai klasei priskirti svorius ir taip padidinti mažiau reprezentuotų klasių svarbą.

#### 4.2.2. Iš anksto apmokyti modeliai bei kontrastinis mokymas

Dažnai mokant neuroninius tinklus susiduriama su duomenų kiekio problema – neretai klasifikacijos užduotyse (o ypač, kai sužymėtų duomenų nėra daug), jų yra nepakankamai jeigu norima kokybiškai apmokyti klasifikatorių. Dar daugiau, tam tikrose užduotyse neuroninius tinklus mokinant tradicinėmis strategijomis, ne visada galima pagerinti klasifikavimo rezultatą elementariai dėl klasių panašumo. Šios dvi problemos itin aktualios prognozuojant miego fazes bei naudojant vieno kanalo duomenis. Visų pirma, tam tikrų miego fazių duomenų yra mažai, be to, tam tikros miego fazės yra tarpusavyje panašios tiek iš medicininės pusės (tai yra, pastebimas panašus smegenų aktyvumas), tiek žymint duomenis juose gali būti ne viena miego fazė.

Vaizdų apdorojimo užduotyse dėl duomenų trūkumo populiaru naudoti iš anksto apmokytus modelius, nes jie leidžia perkelti žinias iš vienos srities į kitą. Toks mokymas paprastai yra vykdomas dvejomis stadijomis: išankstinio apmokymo (angl. *pretaining*) bei adaptacijos (angl. *fine-tuning*) stadijomis. Bendrąja prasme, pirmoje stadijoje modelis apmokomas su dideliu duomenų rinkiniu, o antroje – pritaikomas skirtingai užduočiai ar tiesiog kitiems duomenims. Išankstinio mokymo metu galima mokinti ir neprižiūrimo mokymo užduočiai – tokiu atveju bendra mokymo schema vadinama save prižiūrinčiu mokymu (angl. *self–supervised learning*).

Vienas iš save prižiūrinčio mokymo metodų – kontrastinis mokymas, kuris remiasi idėja, kad modeliai gali išmokti reikšmingas reprezentacijas, tiesiogiai nenaudojant sužymėtų duomenų. Šis metodas neprižiūrimo mokymo metu naudoja dviejų skirtingų įvesčių poras [28]:

- Teigiama pora siekiama maksimizuoti panašumą tarp šių porų reprezentacijų.
- Neigiama pora panašumas tarp šių porų reprezentacijų turi būti minimizuotas.

Porų reprezentacijos dažniausiai išgaunamos naudojant pasirinktos architektūros neuroninį tinklą (šiame darbe tam naudojamasi KNT) bei naudojant papildomą projekcinį sluoksnį. Iš esmės, tokiu mokymu prieš adaptavimo fazę siekiamą gauti tokią neuroninio tinklo konfigūraciją, kuri palengvintų tolimesnės užduoties atlikimą. Pastebima, jog taip mokinant modelius ne tik pasiekiami geresni rezultatai nei naudojant prižiūrimą mokymą, tačiau adaptacijos fazę galima vykdyti su mažiau sužymėtų duomenų [5]. Dar daugiau, tokios tendencijos randamos ir taikant kontrastinį mokymą miego fazių klasifikacijos užduotyse [30].

#### 4.2.3. SimCLR karkasas

SimCLR – vaizdų užduotyse dažnai naudojamas karkasas, kurio esmė – išankstinio apmokymo metu modelį mokinti ne prižiūrimo, o kontrastinio mokymo užduočiai. Žemiau (4 pav.) pavaizduota SimCLR schema išankstinio apmokymo stadijoje. Jos metu:

- A. Atliekamos duomenų augmentacijos, kurių pagalba sukuriamos dvi to paties paveiksliuko pradinės reprezentacijos.
- B. Ištraukiami informatyvūs požymiai tiek [5], tiek šiame tyrime tam pasinaudota KNT.
- C. Projekcinis sluoksnis informatyvūs požymiai suprojektuojami į mažesnę erdvę, taip sumažinamas dimensijų kiekis. Reikalinga gerinti mokymo procesui [5].



4 pav. SimCLR schema, išankstinio apmokymo stadija. Pavaizduota: augmentacijų taikymas (A), informatyvių požymių ištraukimas (B), projekcinis sluoksnis (C), panašumas tarp vektorių (žalia rodyklė). Parengta pagal [5].

SimCLR kūrėjai šioje mokymo stadijoje siūlo naudoti specifinę, NT–Xent nuostolių funkciją (angl. *Normalized Temperature Scaled Cross Entropy*):

$$\mathcal{L}_{i,j} = -\log \frac{\exp(\operatorname{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\operatorname{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)}$$
(4.6)

Laikoma, jog  $\tau$  – temperatūros parametras,  $z_i$  ir  $z_j$  – teigiami pavyzdžiai,  $z_i$  ir  $z_k$  – neigiami pavyzdžiai, N – duomenų rinkinio dydis, bei:

$$\sin(u, v) = \frac{u^{\top} \cdot v}{\|u\| \|v\|}$$
(4.7)

Čia *u* bei *v* – reprezentacijų vektoriai.

Kaip galima matyti iš 4 pav., neigiamos poros nėra tiesiogiai konstruojamos, o tiesiog parenkamos iš duomenų rinkinio. Kitaip tariant, traktuojama, jog dvi augmentuotos to paties paveiksliuko versijos yra teigiama pora. Analogiškai, dviejų skirtingų paveiksliukų augmentacijos (net ta pati augmentacija) laikoma neigiama pora.

#### 4.3. Duomenų augmentacijos

#### 4.3.1. Dažniausiai taikomos duomenų augmentacijos

Duomenų augmentacijos – dažnai taikomas duomenų rinkinio praplėtimo ar diversifikavimo metodas. Dar daugiau, jos kai kuriais atvejais yra neatsiejama modelio dalis. Tarkime, šiame darbe naudojama kontrastinio mokymo SimCLR schema stirpriai priklausoma nuo skirtingų to paties vaizdo augmentacijų. Kalbant apie tradicines, dažniausiai taikomas vaizdų augmentacijas, [2] išskiria tokias grupes:

1. Geometrinės transformacijos – pasukimas (o taip pat ir lokalus pasukimas), apvertimas, apkirpimas, poslinkis.

- 2. Fotometrinės šviesos bei spalvų pakeitimas.
- 3. Triukšmo atsitiktinis ištrynimas, triukšmo idėjimas.
- 4. Filtro naudojami konvoliuciniai branduoliai su fiksuotais parametrais.

Laikoma, jog kiekviena iš šių augmentacijų grupių yra savaip naudinga vaizdų analizės užduotims. Pavyzdžiui, fotometrinėmis transformacijomis galima atkartoti natūraliai atsirandančius apšvietimo pokyčius, geometrinėmis transformacijos galima praplėsti duomenų rinkinį, filtro transformacijos gali padėti aptikti paveiksliukuose esančias kraštines [2]. Be to, duomenų augmentacijos, padidindamos duomenų įvairovę, sumažina persimokymo tikimybę. Net ir naudojant paprastas duomenų augmentacijas, modelis mokomas atpažinti esminius duomenų bruožus, nepaisant galimų variacijų.

#### 4.3.2. Skalogramų augmentacijų problematika

Tyrimuose aptikta, jog augmentuojant skalogramas taikomos ir įprastos duomenų augmentacijos (tarkime, taikyta [27]). Vis tik, verta pažymėti, jog ne visada toks augmentavimas yra korektiškas. Visų pirma, pirmiausiai į akis krenta tai, jog dažnai naudojama apvertimo/pasukimo augmentacija. Tačiau galima diskutuoti, ar ši transformacija yra tinkama duomenims, kuriuose paveiksliuko pozicija yra svarbi (skalogramos yra tokie duomenys) [2, 11]. Taip pat, ne visos spalvinės transformacijos atrodo pagrįstos tam tikrais atvejais, pavyzdžiui, vaizdo pavertimas į juodai baltą skalogramų atveju galimai iškreiptų reprezentuojamų koeficientų dydį (priklausomai nuo skalogramai parinktos spalvų gamos). Deja, yra gana sunku pasakyti, kurios iš tradicinių augmentacijų yra tinkamos naudoti su skalogramomis, nes jos gali būti korektiškos, tačiau turėti kitokį efektą nei su įprastais vaizdais.

Tačiau yra ir bazinių augmentacijų, kurios sukurtos būtent laiko–dažnių sritį reprezentuojantiems vaizdams. [38] išskiria tris augmentacijas, kurias siūlo naudoti šnekamosios kalbos atpažinimo užduotyse būtent ant Mel spektrogramų: laiko iškraipymą, laiko bei dažnių maskavimą (pastarosios dvi augmentacijos panašios į atsitiktinio ištrynimo techniką, kuri dažnai taikoma tradiciniuose vaizduose). [11] pasinaudojo šiomis technikomis bei taip pat pritaikė druskos ir pipirų, *MixUp* (kuomet sujungiamos dvi skalogramos) bei triukšmo augmentacijas ir taip pagerino pradinį KNT modelį.

Papildomai, specializuotuose taikymuose naudojamas ne tik nepriklausomas augmentacijų taikymas. Įprastai, kuomet taikomos duomenų augmentacijos, jos taikomos nekontroliuojant jų sąveikos bei nustatant tik tikimybę, jog specifinė augmentacija bus naudojama (kitaip tariant, atliekamos nepriklausomai). Tai reiškia, jog vienas ir tas pats vaizdas gali būti transformuojamas daugybę kartų. Tai yra tikrai tinkamas augmentavimo metodas, kuomet augmentacijomis siekiama atkartoti natūraliai gamtoje esančius dėsningumus (tarkime, apšvietimo bei pozicijos pokyčius), nes tokios augmentacijos iš esmės pernelyg neiškreipia duomenų (taigi – ir jų kombinacija duomenų taip pat neiškreipia). Tačiau toks metodas gali būti ir netinkamas, kuomet augmentacijomis nėra siekiama vien praplėsti duomenų rinkinį arba kuomet duomenys yra itin jautrūs.

Net ir paprastas vaizdų augmentacijas galima taikyti priklausomai. Tarkime, anksčiau aptartame SimCLR karkase duomenų augmentacijos taikomos sekvenciškai (kuomet dirbama su įprastais vaizdais), nes tokio karkaso tikslas – sukurti kelias vieno paveiksliuko versijas, tačiau jos turi būti tarpusavyje palyginamos. Tačiau augmentuojant skalogramas bei taikant augmentacijas nepriklausomai ar sekvenciškai gali kilti pavojus, jos pernelyg iškreips pradinius duomenis ir dėl to neturės prasmės arba bus perteklinės. Tarkime, taikant dažnių bei laiko filtravimą, gali būti nekorektiška kartu taikyti apkirpimą bei druskos ir pipirų augmentacijas, nes jos irgi atlieka tam tikrą maskuojančią funkciją. Paprasčiausias sprendimas tokiu atveju dirbant su skalogramomis – tiesiog taikyti augmentacijas priklausomai, tarkime, vienu metu naudoti tik vieną augmentaciją arba kelias kartu prasmingas kombinacijas (pavyzdžiui, kombinuoti dažnių ir laiko maskavimą).

### 4.4. Miego fazių klasifikavimo ypatumai

#### 4.4.1. Kryžminės validacijos poreikis

Neuroninių tinklų mokymo proceso metu reikia įvertinti, ar neįvyksta persimokymas/nepakankamas išmokimas bei kaip gerai neuroninis tinklas generalizuzoja ant nematyto duomenų poaibio. Tam mažesnės apimties eksperimentuose duomenys skaidomi į mokymo/validavimo bei testavimo poaibius. Mokymo poaibio paskirtis aiški (neuroninis tinklas mokosi iš šių duomenų), o mokymo progresas stebimas ant validacijos duomenų poaibio. Kadangi mokymo/modelio atrinkimo proceso metu dažnai atliekama įvairių parametrų paieška, testavimo poaibis yra itin reikalingas – jeigu duomenys skaidomi tik į mokymo ir validavimo (nepaliekant testavimo poaibio), iškyla persimokymo ant validacijos duomenų grėsmė.

Vis dėlto tokia trijų poaibių (mokymo, validavimo ir testavimo) duomenų skaidymo strategija nėra visuomet optimalus pasirinkimas, ypač dirbant su medicininiais duomenimis. Medicininiai duomenys dažnai pasižymi didele įvairove, dėl ko kyla rizika, jog vertinant modelį ant testavimo duomenų galima gauti pernelyg optimistiškas prognozes. Atsižvelgus į tai, jog klasifikuojami medicininiai duomenys, šiame darbe bei šios krypties tyrimuose, naudojamas k–grupių kryžminės validacijos metodas. Papildomai, kryžminė validacija aktuali ir dėl to, jog MTD [26] metu rasta, jog taikant skirtingas vilnelių transformacijas, modelių įverčiai buvo gana arti, todėl svarbu įvertinti, ar stebimi skirtumai statistiškai reikšmingi.



5 pav. Kryžminės validacijos schema kuomet k = 5. Sukurta autorės.

Konkrečiai, šiame darbe atliekama 5 grupių kryžminė validacija (bendra schema pavaizduota 5 paveiksliuke). Tokio vertinimo metodo esmė yra paprasta: duomenys suskirstomi į mokymo ir

testavimo duomenų grupes. Modeliai apmokomi su vis kitu duomenų poaibiu iš bendros mokymo duomenų aibės, mokymo procesas stebimas bei koreguojamas naudojant validacijos duomenis (kiekvieno modelio atveju validacijos duomenų aibė nesutampa). Tokiu būdu apmokyti KNT vertinami ant niekur nenaudoto testavimo duomenų poaibio ir taip gaunamos reikalingos metrikos, kurios suvidurkinamos.

#### 4.4.2. Klasifikavimo vertinimo metrikos

Dažnai klasifikavimo užduotyse modeliams vertinti sutinkamos įvairios metrikos: tikslumas (angl. *accuracy*), jautrumas (angl. *recall*), preciziškumas (angl. *precision*), F1 metrika (harmoninis jautrumo ir preciziškumo vidurkis). Kadangi miego fazių klasifikavimas iš esmės yra daugelio klasių prognozavimo užduotis su nebalansuotu duomenų rinkiniu, verta trupmai aptarti įvairių metrikų korektiškumą. Prieš aptariant metrikas, verta paminėti toliau naudojamus žymėjimus:

- *i* konkreti klasė.
- *C* visų klasių kiekis.
- N visų klasifikuojamų atvejų kiekis
- TP<sub>i</sub> teisingai suklasifikuoti *i* klasės atvejai (angl. *True Positives*).
- FN<sub>i</sub> klaidingai suklasifikuota, jog tai ne *i* klasė (angl. *False Negatives*).
- FP<sub>i</sub> klaidingai suklasifikuota kaip *i* klasė (angl. *False Positives*).

Bendrąja prasme, dažnai sutinkama tikslumo metrika (teisingai suklasifikuotų pavyzdžių proporcija) neblogai tinka vertinti būtent binarinę klasifikaciją. Jeigu yra daugiau dvi klasės (o taip pat, jeigu duomenys stipriai nebalansuoti), ši metrika gali būti stipriai iškreipta [37]. Kaip alternatyvą tikslumo metrikai, vertinant bendrą klasifikatoriaus tikslumą siūloma naudoti Koheno kapą (žym.  $\kappa$ , angl. *Cohen's Kappa*), kurios skaičiavime atsižvelgiama į galimą atsitiktinį sutikimą [6]. Jos formulė pateikiama žemiau:

$$\kappa = \frac{P_o - P_e}{1 - P_e} \tag{4.8}$$

Čia laikoma, jog:

•  $P_o$  – teisingai suklasifikuotų atvejų proporcija (arba įprasta tikslumo metrika)

$$P_o = \sum_{i=1}^C \left(\frac{\mathrm{TP}_i}{N}\right) \tag{4.9}$$

•  $P_e$  – atsitiktinis dviejų vertintojų (šiuo atveju – medikų ir naudojamo KNT) sutikimas:

$$P_e = \sum_{i=1}^{C} \left( \frac{\mathrm{TP}_i + \mathrm{FP}_i}{N} \right) \cdot \left( \frac{\mathrm{TP}_i + \mathrm{FN}_i}{N} \right)$$
(4.10)

Kadangi miego fazių klasifikavimo užduotyse yra daugiau nei 2 klasės, svarbu patikrinti, kokios yra jautrumo (kitaip dar vadinama atkūrimo statistika), preciziškumo bei F1 metrikos. Žemiau pateiktos šių metrikų formulės

$$Jautrumas_i = \frac{TP_i}{TP_i + FN_i}$$
(4.11)

$$\operatorname{Preciziškumas}_{i} = \frac{\operatorname{TP}_{i}}{\operatorname{TP}_{i} + \operatorname{FP}_{i}}$$
(4.12)

$$F1_{i} = 2 \times \frac{\text{Preciziškumas}_{i} \times \text{Jautrumas}_{i}}{\text{Preciziškumas}_{i} + \text{Jautrumas}_{i}}$$
(4.13)

Norint įvertinti bendrą klasifikatoriaus gerumą, taip pat galima naudoti šias statistikas, tačiau autorė atkreipia dėmesį, jog itin svarbu, kaip yra skaičiuojamas tas įvertis. Viena iš esminių problemų, su kuriomis susiduriama klasifikuojant miego fazių duomenis yra tai, kad jie nebalansuoti. Mažiausiai reprezentuojama miego fazė yra būtent I miego fazė (ta pati, kurią sunkiausia vertinti net ir medikams), tačiau konkretus fazių disbalansas priklauso nuo pasirinkto duomenų rinkinio. Kuomet duomenys nėra subalansuoti, geriau naudoti makro vidurkinimą, kuomet iš pradžių paskaičiuojami metrikų įverčiai kiekvienai klasei ir tik tada rezultatai suvidurkinami. Taip kiekviena klasė laikoma lygiaverte. Žemiau pateiktas makro F1 skaičiavimas, tokia pati logika galioja ir jautrumo bei preciziškumo metrikoms.

Makro F1 = 
$$\frac{1}{N} \sum_{i=1}^{N} F1_i$$
 (4.14)

#### 4.4.3. Modelių paaiškinamumas GradCAM metodu

Viena iš giliojo mokymosi metodų taikymo problemų – jų paaiškinamumas. Įprastai laikoma, jog neuroninis tinklas yra kaip "juodoji dėžė", o taip ypač dažnai buvo daroma vaizdų analizės užduotyse. Šiai problemai spręsti buvo pasiūlytas GradCAM metodas (nuo angl. *Gradient-Weighted Class Activation Mapping*) [45], kurio esminis tikslas – vizualiai pabrėžti KNT sprendimus. Kitaip tariant, dažniausias taikymas yra paryškinti, kurios būtent paveiksliuko dalys labiausiai nulemia neuroninio tinklo sprendimą. GradCAM algoritmas naudojamas su jau apmokytais KNT, todėl gali teikti atgalinį ryšį programuotojams apie galimus algoritmų tobulinimus.

6 pav. pavaizduota GradCAM schema, tačiau verta panagrinėti kiekvieną algoritmo žingsnį. Paprastai aprašant šį algoritmą galima išskirti, jog jis iš esmės susideda iš trijų dalių:

- Tiesioginis sklidimas: jo metu išskiriami požymių žemėlapiai (žym. A<sup>k</sup>) bei klasės įvertis (žym. y<sup>c</sup>, kur c – klasifikuojamos klasės indeksas).
- 2. Atgalinis sklidimas: jo metu skaičiuojami gradientai, kurie suvidurkinami naudojant globalų sutelkimą.
- 1 bei 2 žingsnyje gauti rezultatai sudauginami bei sumuojami. Pritaikoma ReLu aktyvacijos funkcija (siekiant palikti tik tas reikšmes, kurios daro teigiamą įtaką) bei taip gaunamas vadinamas aktyvacijų žemėlapis.



6 pav. GradCAM schema, parengta remiantis [45]. Raudonos spalvos rodyklėmis pavaizduotas tiesioginis sklidimas, mėlynos – atgalinis sklidimas.

Taip galima surasti, kurios neuroninio tinklo dalys daugiausiai lemia sprendimą, tai yra, kurios aktyvacijos svarbiausios. Kadangi formaliai ši metodą galima taikyti bet kuriam KNT sluoksniui, galima pamatyti netgi tarpinius aktyvacijų žemėlapius. Vis tik, praktiniuose taikymuose (ir tarkime, teikiant atgalinį ryšį specialistams), siekiant žmogui suprantamo vizualinio interpretuojamumo, naudojamas paskutinis požymių žemėlapis (tai yra, paskutinis konvoliucinis sluoksnis).

GradCAM kūrėjai pastebi, jog priklausomai nuo tikslo, galima naudoti tiek teigiamus, tiek neigiamus gradientus bei toliau vykdyti skaičiavimus ta pačia seka. Kuomet aktyvacijų žemėlapiams gauti naudojami neigiami gradientai, gaunami kontrafaktiniai aktyvacijų žemėlapiai (angl. *counterfactual explanations*), kuriuose iš esmės pavaizduota, kurie regionai neigiamai prisideda prie modelio sprendimo prognozuojant klasę, tai yra, kuriuos regionus pašalinus, sprendimas galimai pagerėtų. Kadangi šiame darbe klasifikuojamos signalų skalogramos, GradCAM metodu paryškintos vietos gali padėti atkreipti dėmesį kokie dažniai bei kuris laiko momentas galimai prisidėjo prie neuroninio tinklo sprendimo arba kurie dažniai/laikas signale kaip tik buvo neigiamai veikiantys prognozę.

### 5. Eksperimentų įgyvendinimas

#### 5.1. Naudotos bibliotekos bei programavimo aplinka

Šis darbas buvo atliekamas Python programavimo kalba (*Jupyter* aplinkoje), *Google Colab* platformoje bei asmeniniame kompiuteryje. Asmeniniame kompiuteryje buvo atliekamas EEG duomenų karpymas bei signalų pavertimas į skalogramas (žr. 5.2 skyrių), *Google Colab* platformoje buvo atliekamas modelių mokymas bei vertinimas. Asmeninio kompiuterio specifikacijos: Linux Mint 21.2 Cinnamon operacinė sistema (5.15.0-125-generic), Intel i7-10510U procesorius, Nvidia GeForce GTX 1650 Max-Q grafinė plokštė, 16 GB operatyviosios atminties (RAM). *Google Colab* aplinkos specifikacijos: Linux Ubuntu 22.04.3 LTS operacinė sistema (6.1.85+), Intel Xeon

2.2 GHz 12 branduolių procesorius, Nvidia L4 grafinė plokštė, 53 GB operatyviosios atminties (RAM).

Naudotos šios Python kalbos bibliotekos:

- Pandas [34]: darbui su csv bei xls tipo failais.
- PyEDFlib [36]: naudojama EDF failų (tai yra, polisomnografinių įrašų) bei jų anotacijų nuskaitymui.
- PyWavelets [29] bei ssqueezepy [35]: VT skirtos bibliotekos, PyWavelets pasižymi labai dideliu vilnelių pasirinkimu, tačiau neturi Morse šeimos vilnelių, dėl ko naudota ir ssqueezepy biblioteka.
- OpenCV [4], PIL [46]: naudota darbui su vaizdais.
- Matplotlib [20] ir Seaborn [48]: naudotos rezultatų vizualizacijai.
- Scikit–learn [40]: naudota duomenų grupavimui atlikti bei klasių svoriams gauti (svoriai naudoti kryžminės entropijos netekties funkcijoje, žr. 5.4.1)
- NumPy [39], SciPy [47], PyTorch [18], Torchvision [33] darbui su skaičių masyvais bei KNT implementacijoms ir mokymui.
- TorchMetrics [10] naudota metrikų skaičiavimams.
- tqdm [7] mokymo progresui sekti.

### 5.2. Duomenys ir jų paruošimas

Šiame tyrime naudojami duomenys iš Sleep–EDFx duomenų bazės [14, 25]. Ši duomenų bazė pasirinkta dėl to, jog yra dažnai naudojama tyrimuose, o taip pat joje yra įvairaus amžiaus pacientų duomenys. Išsamesnė joje esančių pacientų informacija aptariama 5.2.1 poskyryje.

Naudojami visi sveikų, sutrikimų neturinčių asmenų duomenys bei pasirenkamas tik vienas EEG kanalas. Prieš sudarant skalogramas, iš duomenų bazės nuskaityti tik EEG Fpz–Cz kanalo duomenys, jie sukarpyti į 30 sekundžių trukmės intervalus. Prafiltruotos nesužymėtos miego fazės ir judėjimo periodai, o taip pat duomenyse esančios 3 ir 4 miego fazės sujungtos į vieną (remiantis 2.2 skyriuje aprašytu rekomenduojamu žymėjimu). Papildomai, kiekvieno įrašo pradžioje bei pabaigoje esantys ilgi atsibudimo periodai sutrumpinti iki 15 minučių.

Kiekvienam miego fazės signalui apskaičiuota vidutinė amplitudė – pagal tai dalis duomenų taip pat atmesti (jeigu buvo už tarpkvartilinio intervalo ribų, skaičiuota grupuojant pagal miego fazes).

#### 5.2.1. Duomenų skaidymas pagal amžiaus grupes

Eksperimentams įgyvendinti, pacientų duomenys pirma suskaidyti pagal amžiaus grupes – taip gauti keturi skirtingi duomenų poaibiai (2 lentelė). Galima atkreipti dėmesį, jog paskutinis duomenų poaibis (vadinamas Sleep–EDF–100) turi ne tik mažiausiai pacientų, tačiau pasižymi ir ženkliai didesniu amžiaus standartiniu nuokrypiu. Vis tik, pasirinkta šio poaibio neskaidyti į papildomas amžiaus grupes, nes tokiu atveju labai sumažėja duomenų kiekis.

Pavadinimas	Amžiaus ribos (imtinai)	Vid. amžius (std.)	Pacientų kiekis
Sleep–EDF–20	25 - 34	28.65 (2.94)	20
Sleep-EDF-60	50 - 60	53.77 (2.83)	22
Sleep-EDF-80	66 - 74	69.05 (2.82)	20
Sleep-EDF-100	85 - 101	90.56 (4.53)	16

2 lentelė. Sudaryti Sleep-EDFx duomenų poaibiai.

**Pastaba**: Sleep–EDF–60, Sleep–EDF–80 bei Sleep–EDF–100 poaibių pavadinimai sugalvoti pagal aukščiausią galimą amžiaus ribą. Kadangi Sleep–EDF–20 poaibis naudojamas kai kuriuose tyrimuose, jo pavadinimas nepakeistas.

Pažymėtina, jog skirtingose amžiaus grupėse yra skirtingas miego fazių disbalansas (matoma 7 pav.). Matomos tokios tendencijos – kuo asmenys vyresni, tuo miegas neramesnis. Tai yra, daugiau atsibudimo miego fazės, mažiau gilaus III miego fazės. Autorė taip pat nori atkreipti dėmesį, jog skirtinguose šios duomenų bazės poaibiuose besiskiriantis atsibudimo miego fazės kiekis parodo to poaibio bendrą miego nestabilumą – kaip aptarta 5.2, visų pacientų duomenys buvo apkarpyti taip, jog paliktas vienodas 15 minučių kiekis miego įrašo gale bei pradžioje iki pirmos/paskutinės užfiksuotos miego fazės. Taigi, atsibudimo fazės kiekio padidėjimas nurodo, jog pacientų grupė buvo labiau linkusi pabusti miego viduryje.



7 pav. Miego fazių kiekis skirtinguose duomenų poaibiuose.

#### 5.2.2. Skalogramų sudarymas

Sukarpytiems ir pravalytiems duomenims skalogramos sudaromos tokia eiga:

- 1. Kiekvienas intervalas normalizuojamas iš reikšmių atimant vidurkį bei padalinant iš standartinio nuokrypio (remiantis [49]).
- Atliekama vilnelės transformacija (naudojantis PyWavelets ir ssqueezepy bibliotekomis), gaunama VT koeficientų matrica. Laikomos tik absoliučios koeficientų reikšmės. Taip pat gaunamas su kiekvienu skalės parametru susietas dažnis.
- 3. Koeficientai paverčiami į skalogramą: pasirenkama naudoti logaritminį mastelį dažnių atžvilgiu bei paveiksliuką sumažinti iki norimo dydžio (šiame tyrime jis yra 224x224).

Visoms skalogramoms konstruoti pasirinktas Jet spalvų žemėlapis (jis ne tik yra dažniausiai naudojamas tyrimuose, tačiau ir turi geresnę skiriamąją gebą, pvz., itin aukšti koeficientai vaizduojami raudonai). 8 pav. matoma, kaip atrodo visos sudarytos skalogramos vienam EEG signalo fragmentui. Kaip galima matyti paveiksliuke, vienas ir tas pat signalas, priklausomai nuo vilnelės, gali būti reprezentuojamas gana skirtingai dažnių bei tam tikrų struktūrų prasme (tai – motininės vilnelės funkcijos pasirinkimo įtaka).



8 pav. Originalus signalo fragmentas (viršuje): abscisių ašyje nurodytas laikas, ordinačių - įtampa ( $\mu$ V). Atliktos vilnelių transformacijos (apačioje): iš kairės dešinę pavaizduota: Morse, kompleksinė Morlet, kompleksinė Gauso bei Mexican Hat vilnelės. Šiuose paveiksliukuose ordinačių ašyje pavaizduota signale esantys dažniai, kuo labiau raudona spalva, tuo didesnės VT gautos absoliučios koeficientų reikšmės.

#### 5.2.3. 5 grupių kryžminės validacijos taikymas

Siekiant įvertinti modelių paklaidas, buvo atliekama 5 grupių kryžminė validacija. Svarbu atkreipti dėmesį į tai, jog miego fazių klasifikavimo užduotyse vienam pacientui galimi keli duomenys, todėl reikia užtikrinti, jog apmokant neuroninius tinklus neįvyksta duomenų nutekėjimas (angl. *data leakage*). Duomenims nutekėjus (vieno paciento duomenys matomi tiek mokymo, tiek validavimo metu), neįmanoma įvertinti, kurioje mokymosi kreivės vietoje modelis yra, nes negalima pasitikėti validacijos metrikomis. Paprasčiausias būdas užtikrinti, jog ši problema neatsitiktų – į mokymo, validavimo bei testavimo duomenų aibes skaidyti pagal unikalų paciento identifikacijos kodą.

Taip buvo daroma ir šiame darbe. Po duomenų pavertimo į skalogramas bei suskaidymo į amžiaus grupes (aprašyta ankstesniuose 5.2.1 ir 5.2.2 poskyriuose), kiekvienam amžiaus poaibiui atliekamas toks skaidymas:

1. Testavimo poaibio atskyrimas: testavimo duomenų poaibiams pasirinkta naudoti 4 pacientų duomenis (Sleep–EDF–60 poaibio atveju– 5 pacientų), pacientų kodai parenkami atsitiktinai

(naudotasi Scikit-learn bibliotekos train\_test\_split funkcija). Jų duomenys padedami į šalį (naudojami tik modelių vertinimui, tai yra modeliai su šiais duomenimis mokomi nebuvo).

2. Mokymo duomenų skaidymas: likę duomenys suskaidomi į 5 grupes (taip pat pagal pacientų kodus, naudojantis Scikit–learn bibliotekos KFold klase). Kiekvieną kartą mokinant modelį, vis kita grupė naudojama kaip validacijos duomenys (o likę duomenys naudojami mokymui).

#### 5.3. Duomenų augmentacijos

Buvo parinktos kontrasto, apkirpimo bei dažnių ir laiko maskavimo augmentacijos (vizualizaciją galima matyti 9 pav.). Visos augmentacijos taikytos su vienoda tikimybe. Augmentacijos taikytos priklausomai, o konkrečiai, vienu metu buvo taikoma tik viena duomenų augmentacija, o ne jų kombinacija (išskyrus laiko bei dažnių maskavimą, jie atliekami visada kartu):

I. Kontrastas: atsitiktinai parenkamas kontrasto faktorius (iš rėžio [0.4, 1.5], parenkant tolygiai). Kiekviena skalogramos reikšmė gaunama:

$$S'_{i,j} = (S_{i,j} - I) \cdot c + I \tag{5.1}$$

Čia laikoma, jog  $S_{i,j}$  – originalus skalogramos pikselis pozicijoje *i*, *j*,  $S'_{i,j}$  – naujas pikselis, *c* – kontrasto faktorius, o *I* – vidutinis RGB kanalų intensyvumas.

- II. Apkirpimas: atsitiktinai parenkamas apkirpimo faktorius (iš rėžio [0.6, 1.0], parenkant tolygiai). Sumažinama atsitiktinai parinkus tokių matmenų kvadratinį langelį. Po apkirpimo paveiksliuko matmenys grąžinami į originalų 224x224 dydį.
- III. Dažnių ir laiko maskavimas: šios augmentacijos taikomos kartu, o taip pat nepriklausomai. Maskuojant tiek dažnių, tiek laiko dimensijas tam tikri paveiksliuko plotai buvo pakeičiami į vidutinę įmanomą Jet spalvų žemėlapio reikšmę.
  - a. Laiko maskavimas: maskuojamas m pikselių pločio vertikalus langas. Pradinis maskavimo indeksas parenkamas atsitiktinai bei tolygiai (iš rėžio [0, W-m], parinktas m = 20, čia W – paveiksliuko plotis).
  - b. Dažnių maskavimas: pasirinkta maskuoti tik tam tikrą dažnių diapazoną (maždaug 0.3Hz). Kiekviena paveiksliuko pikselių eilutė susiejama su dažniu, tolygiai parenkamas pradinis maskavimo dažnis. Randamas viršutinis maskavimo dažnis, maskuojamos tos eilutės, kurių indeksų dažniai patenka į intervalą tarp pradinio ir viršutinio dažnio. Jeigu paveiksliuko pikselių eilutė reprezentuoja daugiau nei šį diapazoną, tuomet maskuojama tik ji.

Įdomu, jog pirminiuose eksperimentuose vietoj kontrasto bandyta naudoti druskos bei pipirų triukšmo augmentaciją, kuri yra naudota su skalogramų duomenimis ir vertinama kaip tinkama [11], tačiau buvo pastebėtas atsibudimo miego fazės pablogėjimas (vienai iš kryžminės validacijos grupių, Sleep–EDF–20 duomenims). Dėl to eksperimentai su šia augmentacija nebuvo tęsiami.



9 pav. Originalios skalogramos bei taikytų augmentacijų pavyzdys.

### 5.4. Konvoliuciniai modeliai ir apmokymo procedūros

Kadangi šio tyrimo tikslas yra ne tik ištirti skirtingų vilnelių įtaką, tačiau ir patikrinti, ar kontrastinio mokymo bei augmentacijų taikymas pageriną miego fazių klasifikaciją, buvo suformuota tokia eksperimentų eiga:

- 1. Atliekamas duomenų paruošimas (aprašyta 5.2). Nepriklausomai nuo to, ar taikomos augmentacijos ar ne, atliekama z-įverčio normalizacija (tai yra, neuroninių tinklų įvestis visiems RGB kanalams kinta intervale [-1, 1]).
- 2. Mokomi baziniai modeliai (SqueezeNet ir ResNet–18, mokymas aprašytas 5.4.1). Eksperimentai atliekami su kiekviena amžiaus grupe.
- 3. Skaičiuojamos makro jautrumo, preciziškumo, F1 bei Koheno kapos metrikos bei F1 metrikos kiekvienai miego fazei (pasinaudota funkcijomis iš TorchMetrics bibliotekos). Išvedamos metrikų statistikos (vidurkiai, standartiniai nuokrypiai ir kt.).
- 4. Įvertinus bazinius modelius, atrenkama geriausia vilnelės bei architektūros kombinacija. Tai nėra labai lengva užduotis, nes, kaip skaitytojas gali pamatyti 6.1.1 poskyryje, modelių rezultatai buvo gana arti, o taip pat, jie skyrėsi priklausomai nuo amžiaus grupės. Vis tik, pasirinkta ResNet–18 architektūra bei kompleksinė Morlet vilnelė.
- 5. Parinkus architektūros bei vilnelės kombinaciją, buvo atliekami eksperimentai su duomenų augmentacijomis ir kontrastiniu mokymu (naudota SimCLR mokymo metodika, mokymas aprašytas 5.4.2). Eksperimentai atliekami su kiekviena amžiaus grupe.
- 6. Kaip ir 3. punkte, skaičiuojamos tos pačios metrikos.
- 7. Pritaikomas Grad-CAM metodas, gaunami vidutiniai aktyvacijų žemėlapiai (žr. 5.4.3 skyrių).

Kaip matoma aukščiau, 5 eksperimentų dalyje lyginama tiek augmentacijų naudojimo, tiek kontrastinio mokymo įtaka, todėl suformuoti bei lyginti keturi mokymo scenarijai:

- I. Bazinis ResNet–18 modelis:
  - a. Nenaudojant augmentacijų (toliau: ResNet<sub>1</sub>, tai yra, vienas iš jau apmokytų bazinių modelių).
  - b. Naudojant augmentacijas (toliau: ResNet<sub>2</sub>). Tikimybė, jog duomenys bus augmentuojami 0.5, vienu metu taikoma tik viena augmentacija.

- II. SimCLR karkaso modelis (ResNet-18 architektūros pagrindu):
  - a. Nenaudojant augmentacijų prižiūrimo mokymo metu (toliau: SimCLR<sub>1</sub>). Visi duomenys augmentuojami neprižiūrimo mokymo stadijoje.
  - b. Naudojant augmentacijas prižiūrimo mokymo metu (toliau: SimCLR<sub>2</sub>). Neprižiūrimo mokymos stadijoje visi duomenys augmentuojami, klasifikavimo metu duomenys augmentuojami kaip ir ResNet<sub>2</sub> scenarijaus atveju.

#### 5.4.1. Baziniai modeliai

Baziniams modeliams pasrinktos SqueezeNet bei ResNet–18 architektūros. Pasirinkta naudoti kiek didesnį nei numatytą atsitiktinio praretinimo transformacijos parametrą (taip siekta labiau reguliarizuoti modelius, nes pirminiuose bei MTD [26] eksperimentuose pastebėtas persimokymas). Parinktas atsitiktinio praretinimo transformacijos parametras 0.65, taikytas tik paskutiniame sluoksnyje (abiejose architektūrose). Taip pat visuose modeliuose pasirinktas fiksuotas duomenų rinkinio dydis – 64. Naudotas AdamW optimizatorius bei kryžminės entropijos nuostolių funkcija (žr. 4.2 skyrių). Skaitytojui taip pat gali būti aktualu žinoti, jog naudojant PyTorch kryžminės entropijos funkciją, nėra taikoma *softmax* aktyvacija paskutiniame sluoksnyje (naudojama CrossEntropyLoss nuostolių funkcijos klasė kaip įvestį priima logitus).

Siekiant kiek sukontroliuoti nebalansuotų duomenų įtaką persimokymui, kryžminės entropijos nuostolių funkcijoje buvo naudoti svorių parametrai, kurie turėtų kompensuoti klasių disproporciją, suteikiant didesnį svorį retesnėms klasėms ir mažesnį – dažniau pasitaikančioms [44]. Klasių svoriai buvo nustatyti atvirkščiai proporcingai klasių dažniams kiekvienoje kryžminės validacijos grupėje (pasinaudojama Scikit–learn bibliotekoje esančia compute\_class\_weight funkcija).

Modeliai buvo mokomi 100 epochų, tačiau kaip papildoma reguliarizacijos strategija buvo taikomas ankstyvas sustojimas (konkrečiai, mokymas stabdomas, jeigu modelis negerėja per 15 epochų, vertinama makro F1 statistika ant validacijos poaibio). Saugomas bei vertinamas paskutinis geriausias modelis. Papildomai, taikoma L2 reguliarizacija. Hiperparametrų aprašas pateiktas 3 lentelėje.

Modelis	Mokymo žingsnis	Atsitiktinis praretinimas	L2 reguliarizacija	Epochų kiekis	Ankstyvas sustojimas
SqueezeNet (visos vilnelės)	1e-05	0.65	0.01	100	15
ResNet–18 (visos vilnelės)	1e-06	0.65	0.01	100	15

3 lentelė. Rezultatuose lyginamų bazinių modelių mokymo hiperparametrai.

#### 5.4.2. Iš anksto apmokyti modeliai naudojant SimCLR

Parinkus geresnę modelio ir vilnelės kombinaciją, modelis buvo iš anksto apmokinamas naudojant SimCLR [5] kontrastinio mokymo schemą. Mokinama 2 stadijomis:

I. Neprižiūrimo mokymo stadija: pasirenkamas duomenų rinkinys, su kuriuo bus vykdomas apmokymas, taip pat modelio architektūra. Paskutinis modelio sluoksnis (klasifikacijos sluoksnis) pakeičiamas į projekcinį sluoksnį (dviejų sluoksnių perceptronas). Pirmojo sluoksnio įvesties dydis yra 512, o išvesties dydis sutampa su antrojo sluoksnio įvestimi. Galutinio projekcinio sluoksnio išvesties dydis yra 128, kaip ir originalioje SimCLR implementacijoje [5]. Tarp sluoksnių naudota aktyvacijos funkcija ReLU.

II. Prižiūrimo mokymo stadija: apmokoma klasifikacijos užduočiai. Projekcinis sluoksnis pakeičiamas į klasifikacijos sluoksnį, naudojama tipinė klasifikacijos netekties funkcija.

Taip pat, svarbu išskirti esmines SimCLR karkaso korekcijas:

- Parinktos augmentacijos: 5.3 skyriuje aptarta, jog parinkta būtent skalogramoms tinkamesnės duomenų augmentacijos, jos naudotos ir neprižiūrimo mokymo stadijoje.
- Papildoma reguliarizacija: išankstinio apmokymo stadijoje tarp perceptrono sluoksnių naudotas atsitiktinis praretinimas (0.2). Taip pat naudotas išankstinis sustojimas (vertinama validacijos nuostolių funkcija). Nors retai, tačiau persimokymas gali įvykti ir išankstinio apmokymo stadijoje [42].

Dažnai galima netgi pasinaudoti jau iš anksto apmokintais modeliais (pavyzdžiui, ant *ImageNet* duomenų rinkinio). Deja, miego fazių skalogramos nėra tipiniai duomenys, todėl buvo pasirinkta išankstinį mokymą atlikti nuo pradžių. Taip pat pažymima, jog išankstinis mokymas buvo atliekamas su tais pačiais duomenimis, su kuriais buvo atliekama klasifikacija. Kitaip tariant, apmokoma su duomenimis iš vienos validacijos grupės, svoriai adaptuojami klasifikacijai. Taip daroma visoms 5 validacijos grupėms.

I bei II mokymo stadijose naudojamos skirtingos netekties funkcijos. Konkrečiai, neprižiūrimo mokymo stadijos metu naudota NT-Xent nuostolių funkcija, skaitytojui žinoma iš 4.2.3 poskyrio (parinktas temperatūros parametras  $\tau = 0.5$ ). Nauodatas AdamW optimizatorius, taikyta L2 reguliarizacija. Duomenų rinkinio dydis – 64.

Klasifikacijos užduočiai mokymo metu naudota 4.2 skyriaus pradžioje pateikta skaitytojui jau žinoma kryžminės entropijos nuostolių funkcija (kaip ir apmokant bazinius modelius, papildomai buvo naudojami svoriai kiekvienai miego fazei). Naudoti tie patys hiperparametrai, kaip ir apmokant bazinį ResNet–18 modelį.

Modelis	Mokymo	Atsitiktinis	L2	Epochų	Ankstyvas
(mokymo stadija)	žingsnis	praretinimas	reguliarizacija	kiekis	sustojimas
SimCLR	1e-06	0.2	0.01	150	15
(išankstinis apmokymas)	10 00	0.2	0.01	150	10
SimCLR	1e 06	0.65	0.01	100	15
(klasifikacija)	10-00	0.05	0.01	100	15

4 lentelė. Rezultatuose lyginamų SimCLR scenarijų hiperparametrai kiekvienai mokymo stadijai.

#### 5.4.3. GradCAM tyrimas

Buvo įgyvendintas GradCAM algoritmas (aprašytas 4.4.3 poskyryje). Algoritmo implementacijoje galimi ir kontrafaktiniai požymių žemėlapiai. Kadangi GradCAM interpretacija, o ypač su skalogramų duomenimis, yra stipriai kokybinė, siekta gauti apibendrinančius aktyvacijų žemėlapius:

I. Skaičiuojami paprasti aktyvacijų žemėlapiai teisingai suklasifikuotiems atvejams (grupuojant pagal miego fazę).

- II. Skaičiuojami kontrafaktiniai aktyvacijų žemėlapiai (naudojant neigiamą gradientą) neisingai suklasifikuotiems atvejams (grupuojant pagal miego fazę).
- III. Gauti aktyvacijų žemėlapiai suvirdurkinami (grupuojant pagal miego fazę).

Kiekvienam kryžminės validacijos modeliui šie žemėlapiai skaičiuojami atskirai, tai yra, nebuvo suvidurkinami skirtingų grupių modelių aktyvacijų žemėlapiai. Vidurkinimo strategija taikyta [15] siekiant palyginti modelių aptiktus regionus su ekspertų išskirtais galimais diagnozei aktualiais regionais. Šiuo atveju, buvo siekta pamatyti, ar požymių žemėlapiai turi didžiausias aktyvacijos vertes panašiuose dažnių diapazonuose kaip teoriškai išskiriami miego fazių dažniai (žr. į 1 lentelę) o taip pat, vizualiai įvertinti, kurie skalogramų regionai gali trukdyti priimti teisingą sprendimą.

### 6. Rezultatai

Žemiau pateiktuose poskyriuose vaizduojami apmokytų modelių rezultatai (vertinama ant testavimo duomenų poaibio). Vertinamos makro jautrumo, preciziškumo, F1 metrikos bei Koheno kapa. Atskirų miego fazių klasifikavimui vertinti naudojama F1 statistika. Pateiktose lentelėse matomi vidutiniai įverčiai bei standartiniai nuokrypiai, tačiau atsižvelgus į tai, jog k-grupių kryžminė validacija buvo atliekama tik 5 kartus (taigi – galima didelė išskirčių įtaka), prieduose pateikiamos ir metrikų medianos bei absoliutūs medianos nuokrypiai (naud. angl. trump. MAD).

### 6.1. Bazinių modelių palyginimas



10 pav. Bazinių modelių mokymosi kreivių pavyzdžiai. Pilka vertikali brūkšninė linija žymi, kurios epochos modelis buvo išsaugotas.

Prieš pradedant nagrinėti rezultatus, verta trumpai paminėti mokymo procesą. Kaip išskirta 5.4 skyriuje, esminė problema, su kuria susidurta pirminiuose eksperimentuose buvo modelių persimokymo reiškinys, todėl siekta reguliarizuoti modelius (naudojant gana didelį atsitiktinį praretinimą bei ankstyvą sustojimą). 10 paveiksliuke galima matyti, kaip su tokiais parametrais mokėsi vienas iš ResNet–18 bei SqueezeNet bazinių modelių. Kaip galima matyti, persimokymo buvo išvengta – pastebimas mažas atstumas tarp mokymo ir validavimo duomenų kreivių tiems modeliams, kurie ir buvo vertinami.

Svarbu pažymėti, jog praktiškai visų modelių rezultatai buvo gana arti, tačiau lentelėse pasirinkta paryškinti tuos rezultatus, kurie buvo statistiškai reikšmingi taikant 1 standartinio nuokrypio taisyklę (taigi, jeigu vidurkis nuo kito vidurkio buvo per jų standartinių nuokrypių sumą). Siekiant pabrėžti vienareikšmišką vilnelių pranašumą buvo arba žymima tik viena vilnelė, arba tos dvi vilnelės, kurios pranašesnės už likusias (tačiau tarpusavyje reikšmingai nesiskiria).

Papildomai, vertinant bazinius modelius, pabraukiami tie rezultatai, kurie yra statistiškai reikšmingai geresni naudojant tą pačią vilnelės transformaciją bei modelio architektūrą, kitaip tariant, naudojant kitą architektūrą, rezultatas statistiškai reikšmingai pablogėjo.

#### 6.1.1. Rezultatai su Sleep–EDF–20 duomenimis

Žemiau pavaizduotose (5 bei 6) lentelėse matyti SqueezeNet ir ResNet–18 architektūrų makro jautrumo, preciziškumo, F1 ir Koheno kapos įverčiai Sleep–EDF–20 duomenų poaibiui. Pastebima jog taikant SqueezeNet modelio architektūrą, nerasta, jog kuri nors vilnelė(-s) statistiškai reikšmingai pranoko kitas, todėl negalima daryti išvadų, jog taikant šią architektūrą, viena ar kelios iš šių vilnelių yra geresnės.

5 lentelė. SqueezeNet architektūros makro statistikų (jautrumo, preciziškumo bei F1) ir Koheno kapos įverčiai Sleep–EDF–20 duomenims.

Vilnelė	Jautrumas	Preciziškumas	F1	К
	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)
Morse	0.706 (0.019)	0.679 (0.013)	0.671 (0.018)	0.666 (0.018)
K. Morlet	0.715 (0.029)	0.696 (0.022)	0.686 (0.029)	0.691 (0.029)
K. Gauso	0.684 (0.028)	0.668 (0.017)	0.663 (0.024)	0.657 (0.03)
Mex. Hat	0.697 (0.022)	0.675 (0.014)	0.666 (0.022)	0.658 (0.026)

6 lentelė. ResNet–18 architektūros makro statistikų (jautrumo, preciziškumo bei F1) ir Koheno kapos įverčiai Sleep–EDF–20 duomenims.

Vilnelė	Jautrumas	Preciziškumas	<b>F1</b>	К
	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)
Morse	0.697 (0.008)	0.672 (0.008)	0.666 (0.007)	0.664 (0.012)
K. Morlet	<b>0.714</b> (0.009)	<b>0.694</b> (0.01)	<b>0.685</b> (0.012)	0.686 (0.015)
K. Gauso	0.674 (0.008)	0.657 (0.007)	0.651 (0.008)	0.646 (0.013)
Mex. Hat	0.679 (0.018)	0.662 (0.011)	0.655 (0.015)	0.652 (0.018)

Kita vertus, taikant ResNet–18 architektūrą, kai kurie skirtumai yra statistiškai reikšmingi. Konkrečiai, stebimi didžiausi makro jautrumo ( $0.714\pm0.009$ ), preciziškumo ( $0.694\pm0.01$ ) bei F1

(0.685±0.012) įverčiai su kompleksine Morlet vilnele, todėl naudojant šią architektūrą su Sleep– EDF–20 duomenimis laikoma, jog ši vilnelė pranašiausia.

Taip pat pastebėtina, jog taikant vieną ir tą pačią vilnelę, nebuvo pastebėti statistiškai reikšmingi bendrų metrikų skirtumai taikant skirtingą architektūrą (tai yra, lyginant tos pačios vilnelės rezultatus 5 ir 6 lentelėse). Esminis skirtumas tarp architektūrų pasimato vertinant standartinius nuokrypius. Taikant SqueezeNet architektūrą, galima pastebėti, jog didžiausi metrikų standartiniai nuokrypiai pastebimi taikant kompleksinę Morlet vilnelę (nuo 0.022 iki 0.029), mažiausi – taikant Morse (nuo 0.013 iki 0.019). Ši tendencija neišsilaiko miego fazes klasifikuojant su ResNet–18 architektūra – didžiausi standartiniai visų metrikų nuokrypiai stebimi taikant Mexican Hat vilnelę (nuo 0.011 iki 0.018). Autorė taip pat nori atkreipti dėmesį, jog lyginant šias dvi architektūras, nepriklausomai nuo vilnelės transformacijos, miego fazes klasifikuojant su ResNet–18 architektūros modeliu, sumažėjo visų įverčių standartiniai nuokrypiai.

Vertinant bendrus F1 įverčius kiekvienai miego fazei (matoma 7 bei 8 lentelėse), galima pastebėti tokias bendras tendencijas: geriausiai buvo klasifikuojamos II bei III miego fazė, blogiausiai – I fazė. Tokie rezultatai nestebina, turint omenyje, jog II miego fazė Sleep–EDF–20 duomenyse yra labiausiai reprezentuota, o III miego fazė, priešingai nei kitos, yra ganėtinai lengvai atskiriama (turi žemų dažnių). I miego fazė, kita vertus yra mažiausiai reprezentuota, dėl jos nesutaria ir ekspertai.

Vilnelė	Atsibudimas	I fazė	II fazė	III fazė	REM fazė
	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)
Morse	0.77 (0.02)	0.26 (0.031)	0.81 (0.013)	0.806 (0.022)	0.707 (0.021)
K. Morlet	0.775 (0.056)	0.25 (0.056)	<b>0.84</b> (0.015)	0.831 (0.021)	0.735 (0.021)
K. Gauso	0.769 (0.05)	0.253 (0.042)	0.798 (0.023)	0.813 (0.02)	0.683 (0.022)
Mex. Hat	0.746 (0.067)	0.257 (0.029)	0.805 (0.019)	0.815 (0.027)	0.706 (0.02)

7 lentelė. SqueezeNet architektūros F1 įverčiai kiekvienai miego fazei, Sleep–EDF–20 duomenims.

8 lentelė. ResNet–18 architektūros F1 įverčiai kiekvienai miego fazei, Sleep–EDF–20 duomenims.

Vilnelė	Atsibudimas	I fazė	II fazė	III fazė	REM fazė
,	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)
Morse	0.788 (0.028)	0.249 (0.01)	0.8 (0.002)	0.796 (0.021)	0.695 (0.012)
K. Morlet	0.78 (0.029)	0.27 (0.026)	<b>0.831</b> (0.011)	0.836 (0.011)	0.711 (0.011)
K. Gauso	0.774 (0.023)	0.22 (0.03)	0.789 (0.005)	0.813 (0.004)	0.659 (0.011)
Mex. Hat	0.769 (0.031)	0.224 (0.029)	0.798 (0.007)	0.823 (0.007)	0.66 (0.026)

Rasta ir statistiškai reikšmingų skirtumų – tiek SqueezeNet, tiek ResNet–18 modelyje klasifikuojant II miego fazę, geriausia buvo kompleksinė Morlet vilnelė (F1 atitinkamai 0.84±0.015 bei 0.831±0.011). Kitoms miego fazėms nerasta viena ar dvi vienareikšmiškai pranašiausios vilnelės. Kaip ir vertinant bendrus metrikų įverčius, pastebima tendencija, jog taikant ResNet–18 modelį praktiškai visoms vilnelėms sumažėjo arba nepakito miego fazių F1 standartiniai nuokrypiai. Išimtis – standartinio nuokrypio padidėjimas atsibudimo fazei taikant Morse vilnelę (+0.008).

Taip pat, vidutiniai F1 įverčiai kiekvienai fazei statistiškai reikšmingai nesiskiria naudojant

kitokią architektūrą (išskyrus Mexican Hat vilnelei, kuomet taikant SqueezeNet modelį stebimas 0.046 F1 padidėjimas). Kitaip tariant, dažnu atveju architektūros parinkimas nedaro įtakos.

#### 6.1.2. Rezultatai su Sleep-EDF-60 duomenimis

9 bei 10 lentelėse matyti SqueezeNet ir ResNet–18 architektūrų makro jautrumo, preciziškumo, F1 bei Koheno kapos įverčiai Sleep–EDF–60 duomenų poaibiui. Taikant ResNet–18 rasta, jog Morse bei kompleksinė Morlet pasižymėjo aukščiausiais jautrumo įverčiais (atitinkamai  $0.689\pm0.008$  bei  $0.696\pm0.012$ ), tačiau vertinant kitas metrikas šios vilnelės nebuvo statistiškai reikšmingai pranašesnės. Taikant SqueezeNet architektūrą, pranašumai tarp vilnelių nepastebimi.

Kaip ir su Sleep–EDF–20 duomenimis, pastebimos dvi tendencijos. Pirma, matoma, jog nei ResNet–18, nei SqueezeNet architektūra nebuvo pranašesnė – lyginant tos pačios vilnelės bendrus metrikų įverčius 9 bei 10 lentelėse, nestebimi statistiškai reikšmingi makro jautrumo, preciziškumo, F1 bei Koheno kapos metrikų skirtumai. Išimtis – kompleksinė Gauso vilnelė, kuriai SqueezeNet architektūra lėmė didesnį makro preciziškumo įvertį (+0.021), tačiau kitos metrikos statistiškai reikšmingai nesiskyrė.

Taip pat, taikant ResNet–18 modelį aptinkami mažesni bendrų metrikų standartiniai nuokrypiai. Išimtys – taikant kompleksinę Morlet vilnelę pakilo jautrumo (+0.003) bei precizišumo (+0.004), o taikant kompleksinę Gauso vilnelę – jautrumo (+0.006) bei F1 (+0.005) metrikų standartiniai nuokrypiai.

Vilnelė	Jautrumas	Preciziškumas	<b>F1</b>	К
, miere	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)
Morse	0.703 (0.015)	0.61 (0.017)	0.618 (0.015)	0.574 (0.016)
K. Morlet	0.692 (0.009)	0.608 (0.01)	0.614 (0.017)	0.573 (0.024)
K. Gauso	<u>0.676</u> (0.003)	0.59 (0.01)	0.596 (0.007)	0.55 (0.009)
Mex. Hat	0.67 (0.024)	0.583 (0.025)	0.591 (0.021)	0.545 (0.022)

9 lentelė. SqueezeNet architektūros makro statistikų (jautrumo, preciziškumo bei F1) ir Koheno kapos įverčiai Sleep–EDF–60 duomenims.

10 lentelė. ResNet–18 architektūros makro statistikų (jautrumo, preciziškumo bei F1) ir Koheno kapos įverčiai Sleep–EDF–60 duomenims.

Vilnelė	Jautrumas	Preciziškumas	<b>F</b> 1	К
,	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)
Morse	<b>0.689</b> (0.008)	0.593 (0.007)	0.605 (0.01)	0.57 (0.011)
K. Morlet	<b>0.696</b> (0.012)	0.598 (0.014)	0.609 (0.011)	0.57 (0.01)
K. Gauso	0.655 (0.009)	0.575 (0.007)	0.587 (0.012)	0.543 (0.017)
Mex. Hat	0.662 (0.011)	0.577 (0.01)	0.588 (0.014)	0.545 (0.019)

11 bei 12 lentelėse matoma F1 įverčiai kekvienai miego fazei ir, trumpai apibendrinti juos galima taip: nepriklausomai nuo parinktos architektūros ar vilnelės transformacijos, geriausiai klasifikuojama atsibudimo, blogiausiai – I miego fazė. Įdomu, jog lyginant jaunesnių pacientų duomenis (tai yra, Sleep–EDF–20 duomenų klasifikaciją), ženkliai prasčiau klasifikuojama II bei III miego fazės, o atsibudimo fazė klasifikuojama geriau. Atsibudimo ir III miego fazių klasifikavimo pokyčiai nestebina, turint omenyje, jog šiame (Sleep–EDF–60) duomenų poaibyje atsibudimo fazės duomenų yra žymiai daugiau, o III – mažiau (žr. 7 pav.). Tačiau II miego fazės klasifikacijos pablogėjimas (lyginant su jaunesnių pacientų duomenimis) kiek stebina, nes šios fazės duomenų netrūksta.

Vilnelė	Atsibudimas	I fazė	II fazė	III fazė	REM fazė
,	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)
Morse	0.83 (0.009)	0.315 (0.019)	0.742 (0.023)	0.568 (0.041)	0.634 (0.026)
K. Morlet	0.827 (0.015)	0.318 (0.011)	0.754 (0.032)	0.572 (0.053)	0.6 (0.021)
K. Gauso	0.84 (0.08)	<u>0.3</u> (0.01)	0.719 (0.014)	0.559 (0.04)	0.562 (0.028)
Mex. Hat	0.831 (0.013)	0.294 (0.026)	0.721 (0.018)	0.565 (0.025)	0.542 (0.05)

11 lentelė. SqueezeNet architektūros F1 įverčiai kiekvienai miego fazei, Sleep–EDF–60 duomenims.

12 lentelė. ResNet–18 architektūros F1 įverčiai kiekvienai miego fazei, Sleep–EDF–60 duomenims.

Vilnelė	Atsibudimas	I fazė	II fazė	III fazė	REM fazė
	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)
Morse	0.828 (0.007)	<b>0.297</b> (0.005)	0.745 (0.011)	0.552 (0.035)	<b>0.606</b> (0.006)
K. Morlet	0.841 (0.006)	<b>0.312</b> (0.024)	0.733 (0.016)	0.562 (0.018)	<b>0.598</b> (0.025)
K. Gauso	0.838 (0.006)	0.274 (0.01)	0.73 (0.015)	0.594 (0.024)	0.496 (0.021)
Mex. Hat	0.833 (0.007)	0.271 (0.014)	0.729 (0.02)	0.585 (0.024)	0.523 (0.028)

11 bei 12 lentelėse taip pat matoma, jog vilnelės parinkimas taikant SqueezeNet architektūrą neturi įtakos atskirų miego fazių F1 įverčiams, tai yra, nebuvo rasta, jog yra pranašių vilnelių. Taikant ResNet–18 architektūrą, aptikta, jog Morse bei kompleksinė Morlet vilnelės pranašios klasifikuojant I miego fazę (F1 Morse vilnelei  $0.297\pm0.005$  bei kompleksinei Morlet vilnelei  $0.312\pm0.024$ ) bei REM fazę (F1 Morse vilnelei  $0.606\pm0.006$  bei kompleksinei Morlet vilnelei  $0.598\pm0.025$ ).

Taikant vieną ir tą pačia vilnelę, modelio architektūros parinkimas dažniausiai nėra svarbus. Išimtis – kompleksinė Gauso vilnelė. Ją taikant, I miego fazę SqueezeNet modelis klasifikuoja statistiškai reikšmingai geriau (F1+0.026) nei taikant ResNet–18 architektūrą. Taip pat, su šia vilnele bei SqueezeNet architektūra statistiškai reikšmingai geriau klasifikuojama ir REM fazė (F1+0.066).

Vėl stebimi sumažėję F1 metrikų standartiniai nuokrypiai taikant ResNet–18 architektūrą. Išimtis – su kompleksine Morlet vilnele padidėję I (+0.013) bei REM (+0.004) miego fazės, o su kompleksine Gauso bei Mexican Hat — padidėję II miego fazės (atitinkamai + 0.002 bei +0.001) F1 įverčių standartiniai nuokrypiai.

#### 6.1.3. Rezultatai su Sleep–EDF–80 duomenimis

13 bei 14 lentelėse matomi SqueezeNet ir ResNet–18 architektūrų makro jautrumo, preciziškumo, F1 bei Koheno kapos įverčiai Sleep–EDF–80 duomenų poaibiui. Pastebima, jog taikant SqueezeNet modelio architektūrą, rasta, jog Morse vilnelė yra pranašiausia – randami geriausi makro jautrumo  $(0.668\pm0.01)$  bei preciziškumo  $(0.622\pm0.014)$  įverčiai. Naudojant ResNet–18 modelio architektūrą, pastebima, jog geriausios vidutinės bendros metrikos matomos naudojant Morse bei kompleksinę Morlet vilneles. Jos pasižymi aukščiausiais jautrumo visų metrikų įverčiais.

Šiame duomenų poaibyje pastebima, jog modelio parinkimas vis tik turi įtakos bendriems makro metrikų įverčiams. Konkrečiai, SqueezeNet architektūros parinkimas naudinagas Morse bei Mexican Hat vilnelių metrikoms. Morse vilnelei taikant ResNet–18 architektūrą, pablogėjo makro jautrumas (-0.028), preciziškumas (-0.032), F1 (-0.028) bei  $\kappa$  (-0.029), o Mexican Hat vilnelei – makro jautrumo (-0.027) bei preciziškumo (-0.017) įverčiai.

Kaip ir anksčiau, naudojant ResNet–18 mažėja ir bendrų metrikų standartiniai nuokrypiai. Išimtis – Morse vilnelė, kurią taikant padidėjo jautrumo (+0.005) bei preciziškumo (+0.001) standartiniai nuokrypiai, o F1 metrikos nuokrypis – nepakito.

Vilnelė	Jautrumas	Preciziškumas	<b>F1</b>	К
( IIIIoio	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)
Morse	<b>0.668</b> (0.01)	<b>0.622</b> (0.014)	0.629 (0.013)	0.571 (0.014)
K. Morlet	0.636 (0.021)	0.58 (0.021)	0.594 (0.029)	0.533 (0.028)
K. Gauso	0.623 (0.024)	0.573 (0.016)	0.58 (0.025)	0.496 (0.042)
Mex. Hat	<u>0.635</u> (0.013)	<u>0.587</u> (0.009)	0.597 (0.011)	0.519 (0.016)

13 lentelė. SqueezeNet architektūros makro statistikų (jautrumo, preciziškumo bei F1) ir Koheno kapos įverčiai Sleep–EDF–80 duomenims.

14 lentelė. ResNet–18 architektūros makro statistikų (jautrumo, preciziškumo bei F1) ir Koheno kapos įverčiai Sleep–EDF–80 duomenims.

Vilnelė	Jautrumas	Preciziškumas	<b>F</b> 1	К
,	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)
Morse	<b>0.64</b> (0.015)	<b>0.59</b> (0.015)	<b>0.601</b> (0.013)	<b>0.542</b> (0.012)
K. Morlet	<b>0.65</b> (0.012)	<b>0.6</b> (0.01)	<b>0.614</b> (0.01)	<b>0.558</b> (0.013)
K. Gauso	0.6 (0.013)	0.568 (0.006)	0.576 (0.008)	0.49 (0.011)
Mex. Hat	0.608 (0.013)	0.57 (0.006)	0.581 (0.007)	0.499 (0.011)

Atskirų miego fazių rezultatai matomi 15 bei 16 lentelėse ir trumpai juos apibendrinus galima pasakyti, jog nepriklausomai nuo vilnelės ar modelio parinkimo, geriausiai klasifikuojama atsibudimo bei II, o blogiausiai – I miego fazės. Geriausiai klasifikuojamų miego fazių šiame duomenų poaibyje yra daugiausiai, o tendencija I miego fazę klasifikuoti prasčiausiai stebima ir su jaunesnių pacientų (Sleep–EDF–20 bei Sleep–EDF-60) duomenimis.

Taip pat rasta, jog Morse vilnelė statistiškai reikšmingai geriau klasifikuoja I miego fazę nei visos kitos vilnelės taikant SqueezeNet modelio architektūrą ( $F1 = 0.341 \pm 0.018$ ). Taikant ResNet–18 architektūrą, randama, jog Morse bei kompleksinė Morlet pranašios klasifikuojant I bei II miego fazes. Taikant Morse vilnelę, modelio parinkimas buvo svarbus, nes su ResNet–18 architektūra stebimas I miego fazės F1 pablogėjimas (-0.042)

Taikant ResNet–18 architektūrą, kaip ir anksčiau, pastebimi mažesni standartiniai nuokrypiai, tačiau šį kartą ši tendencija nėra tokia išreikšta. Pastebimos išimtys – Morse, kompleksinė Gauso ir Mexican Hat vilnelės. Taikant kompleksinę Gauso vilnelę nežymiai padidėjo III miego fazės

(+0.001), o taikant Mexican Hat vilnelę – atsibudimo fazės (+0.006) F1 standartiniai nuokrypiai. Taikant Morse vilnelę su ResNet–18 modeliu padidėjo visų (išskyrus III) miego fazių standartiniai nuokrypiai.

15 lentelė. SqueezeNet architektūros F1 įverčiai kiekvienai miego fazei, Sleep–EDF–80 duomenims.

Vilnelė	Atsibudimas	I fazė	II fazė	III fazė	REM fazė
	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)
Morse	0.79 (0.01)	<b>0.341</b> (0.018)	0.768 (0.006)	0.607 (0.013)	0.639 (0.041)
K. Morlet	0.762 (0.037)	0.26 (0.027)	0.745 (0.024)	0.62 (0.013)	0.583 (0.085)
K. Gauso	0.758 (0.046)	0.249 (0.027)	0.698 (0.033)	0.63 (0.016)	0.567 (0.05)
Mex. Hat	0.762 (0.028)	0.267 (0.025)	0.723 (0.007)	0.635 (0.01)	0.597 (0.03)

16 lentelė. ResNet–18 architektūros F1 įverčiai kiekvienai miego fazei, Sleep–EDF–80 duomenims.

Vilnelė	Atsibudimas	I fazė	II fazė	III fazė	REM fazė
	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)
Morse	0.77 (0.024)	<b>0.299</b> (0.02)	<b>0.757</b> (0.011)	0.597 (0.008)	0.582 (0.043)
K. Morlet	0.776 (0.024)	<b>0.286</b> (0.02)	<b>0.765</b> (0.009)	0.619 (0.007)	0.626 (0.024)
K. Gauso	0.746 (0.027)	0.249 (0.014)	0.72 (0.008)	0.616 (0.017)	0.547 (0.008)
Mex. Hat	0.753 (0.034)	0.25 (0.014)	0.72 (0.007)	0.623 (0.007)	0.556 (0.02)

#### 6.1.4. Rezultatai su Sleep-EDF-100 duomenimis

17 bei 18 lentelėse matomi makro jautrumo, preciziškumo, F1 bei Koheno kapos įverčiai Sleep– EDF–100 duomenų poaibiui. Nepastebima, jog kažkuri vilnelė pranašiausia taikant tiek Squeeze-Net, tiek ResNet–18 architektūras. Taip pat randama, jog ResNet–18 architektūros taikymas naudingas Morse bei Mexican Hat vilnelėms. Taikant šią architektūrą statistiškai reikšmingai padidėjo Morse vilnelės preciziškumo (+0.052) bei Koheno kapos (+0.061 įverčiai), o taip pat Mexican Hat vilnelės Koheno kapos (+0.055) įvertis.

17 lentelė. SqueezeNet architektūros makro statistikų (jautrumo, preciziškumo bei F1) ir Koheno kapos įverčiai Sleep–EDF–100 duomenims.

Vilnelė	Jautrumas	Preciziškumas	<b>F</b> 1	К
,	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)
Morse	0.522 (0.028)	0.475 (0.027)	0.465 (0.033)	0.455 (0.031)
K. Morlet	0.505 (0.059)	0.483 (0.044)	0.47 (0.058)	0.464 (0.077)
K. Gauso	0.515 (0.015)	0.508 (0.029)	0.495 (0.019)	0.497 (0.013)
Mex. Hat	0.524 (0.03)	0.489 (0.019)	0.486 (0.018)	0.485 (0.028)

Vilnelė	Jautrumas Preciziškumas		<b>F1</b>	К
( IIIIOIO	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)
Morse	0.546 (0.012)	0.527 (0.021)	0.515 (0.022)	0.516 (0.023)
K. Morlet	0.554 (0.011)	0.534 (0.017)	0.527 (0.008)	0.537 (0.018)
K. Gauso	0.510 (0.023)	0.522 (0.018)	0.507 (0.016)	0.503 (0.024)
Mex. Hat	0.537 (0.015)	0.519 (0.013)	0.514 (0.012)	<u>0.54</u> (0.015)

18 lentelė. ResNet–18 architektūros makro statistikų (jautrumo, preciziškumo bei F1) ir Koheno kapos įverčiai Sleep–EDF–100 duomenims.

Vėl matoma tendencija, jog taikant ResNet–18 modelį sumažėja beveik visų metrikų standartiniai nuokrypiai, išskyrus kuomet taikoma kompleksinė Gauso vilnelė, tada stebimas makro jautrumo bei Koheno kapos metrikos standartinių nuokrypių padidėjimai (atitinkamai +0.008 bei +0.009). Taip pat, taikant SqueezeNet architektūrą šiems duomenims stebimi pakankamai dideli standartiniai nuokrypiai (lyginant su kitais duomenų poaibiais), kas rodo, jog SqueezeNet modelis šiam duomenų poaibiui yra labai nestabilus.

Analizuojant atskirų miego fazių klasifikaciją (matoma 19 bei 20 lentelėse), trumpai apibendrinti galima taip: naudojant tiek SqueezeNet, tiek ResNet–18 modelį, nepriklausomai nuo vilnelės parinkimo, geriausiai klasifikuojama atsibudimo, blogiausiai – REM bei I miego fazės. Atsibudimo fazės Sleep–EDF–100 duomenų poaibyje yra daugiausiai, todėl gera šios fazės klasifikacija nestebina, kaip ir nestebina tai, jog viena iš blogiausiai klasifikuojamų fazių yra I miego fazė. Kas keista, tai labai žemi REM fazės įverčiai – nors ši fazė nėra labai reprezentuota, kituose duomenų poaibiuose jos taip pat nėra daug, tačiau ji klasifikuojama daug geriau.

19 lentelė. SqueezeNet architektūros F1 įverčiai kiekvienai miego fazei, Sleep–EDF–100 duomenims.

Vilnelė	Atsibudimas	I fazė	II fazė	III fazė	REM fazė
	Vid. (Std.)				
Morse	0.836 (0.039)	0.201 (0.019)	0.491 (0.071)	0.588 (0.048)	0.21 (0.082)
K. Morlet	0.862 (0.051)	0.203 (0.031)	0.491 (0.09)	0.606 (0.065)	0.189 (0.079)
K. Gauso	0.87 (0.008)	0.193 (0.018)	0.559 (0.062)	0.627 (0.045)	0.226 (0.07)
Mex. Hat	0.854 (0.021)	0.203 (0.024)	0.56 (0.066)	0.629 (0.029)	0.184 (0.041)

20 lentelė. ResNet–18 architektūros F1 įverčiai kiekvienai miego fazei, Sleep–EDF–100 duomenims.

Vilnelė	Atsibudimas	I fazė	II fazė	III fazė	REM fazė
	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)
Morse	0.866 (0.007)	0.236 (0.009)	0.571 (0.043)	0.619 (0.029)	0.281 (0.038)
K. Morlet	0.883 (0.013)	0.228 (0.013)	<u>0.626</u> (0.039)	<b>0.66</b> (0.005)	0.239 (0.076)
K. Gauso	0.872 (0.021)	0.208 (0.01)	0.588 (0.027)	0.611 (0.035)	0.255 (0.05)
Mex. Hat	0.881 (0.015)	0.222 (0.012)	<u>0.65</u> (0.021)	0.615 (0.044)	0.202 (0.031)

Aptikta ir pranaši vilnelė taikant ResNet–18 modelio architektūrą – III miego fazė buvo klasifikuojama geriausiai, taikant kompleksinę Morlet vilnelę (F1 =  $0.66\pm0.005$ ). Kita vertus, kitoms miego fazėms (o tai pat ir taikant SqueezeNet architektūrą), nebuvo pastebėta, jog viena kažkuri vilnelė statistiškai reikšmingai geresnė už kitas.

Papildomai, matoma, jog naudojant ResNet–18 architektūrą stebimi net keli statistiškai reikšmingi pagerėjimai. Visų pirma pastebimas I miego fazės F1 įverčio pagerėjimas Morse duomenims (+0.035). Papildomai, matoma, jog su šia architektūra pagerėja ir II miego fazės klasifikavimas tiek kompleksine Morlet (+0.035), tiek Mexicat Hat (+0.09) vilnelėmis transformuotiems duomenims.

Kaip ir analizuojant prieš tai buvusių miego duomenų poaibių rezultatus, matoma, jog naudojant ResNet–18 modelio architektūrą, sumažėja beveik visų vilnelių F1 įverčių standartiniai nuokrypiai. Išimtys – taikant kompleksinę Gauso vilnelę pastebėtas atsibudimo fazės (+0.013), o taikant Mexican Hat vilnelę – III miego fazės (+0.015) F1 įverčių standartinių nuokrypių padidėjimas.

#### 6.2. SimCLR bei augmentacijų taikymo rezultatai

Žemiau pateiktuose poskyriuose nagrinėjama išankstinio mokymo bei augmentacijų taikymo įtaką tiek bendroms metrikoms (6.2.1 poskyris), tiek atskirų miego fazių klasifikavimo F1 įverčiams (6.2.2 poskyris). Autorė primena, jog visi lyginami modeliai buvo ResNet–18 architektūros pagrindu, o duomenys buvo transformuoti kompleksine Morlet vilnele, nes laikoma, kad tokia kombinacija yra verta tobulinimo.

Visose lentelėse juoda spalva išskirti tie modelių rezultatai, kurie statistiškai reikšmingai skiriasi nuo bazinio modelio (ResNet<sub>1</sub>), kuris apmokytas be išankstinio mokymo bei netaikant duomenų augmentacijų. Jeigu paryškinti ResNet<sub>1</sub> modelio rezultatai, reiškia, kad buvo pastebėtas statistiškai reikšmingas pablogėjimas.

#### 6.2.1. Bendri įverčiai

Bendrų metrikų įverčiai matomi 21, 22, 23 bei 24 lentelėse. Kaip matoma 21 lentelėje, makro F1 įverčiai taikant skirtingą apmokymo strategiją ar taukant augmentacijas statistiškai reikšmingai nesiskyrė nei vienam duomenų poaibiui . Tą patį galima pasakyti ir apie makro preciziškumo bei Koheno kapos įverčius (22 bei 24 lentelės).

Statistiškai reikšmingi skirtumai (lyginant su baziniu modeliu) matomi vertinant makro jautrumo įverčius (23 lentelė). Konkrečiai, stebimi pagerėjimai naudojant SimCLR<sub>2</sub> scenarijų Sleep– EDF–20 poaibiui (+0.024), o taikant SimCLR<sub>2</sub>, matomas pagerėjimas Sleep–EDF–60 (+0.022) bei Sleep–EDF–80 duomenims (+0.027).

21 lentelė. Suformuotų scenarijų (taikant SimCLR bei augmentacijas) makro F1 įverčiai kiekvienam duomenų poaibiui.

Scenariius	Sleep–EDF–20	Sleep-EDF-60	Sleep-EDF-80	Sleep-EDF-100
	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)
ResNet <sub>1</sub>	0.685 (0.012)	0.609 (0.011)	0.614 (0.01)	0.527 (0.008)
ResNet <sub>2</sub>	0.682 (0.02)	0.61 (0.012)	0.625 (0.008)	0.528 (0.029)
SimCLR <sub>1</sub>	0.701 (0.014)	0.627 (0.013)	0.629 (0.016)	0.53 (0.021)
SimCLR <sub>2</sub>	0.7 (0.018)	0.626 (0.012)	0.632 (0.014)	0.527 (0.035)

Scenariius	Sleep-EDF-20	Sleep-EDF-60	Sleep-EDF-80	Sleep-EDF-100
	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)
ResNet <sub>1</sub>	0.694 (0.01)	0.598 (0.014)	0.6 (0.01)	0.534 (0.017)
ResNet <sub>2</sub>	0.689 (0.015)	0.598 (0.011)	0.607 (0.009)	0.539 (0.027)
SimCLR <sub>1</sub>	0.705 (0.009)	0.62 (0.011)	0.614 (0.015)	0.547 (0.027)
SimCLR <sub>2</sub>	0.705 (0.012)	0.618 (0.011)	0.615 (0.011)	0.54 (0.037)

22 lentelė. Suformuotų scenarijų (taikant SimCLR bei augmentacijas) makro preciziškumo įverčiai kiekvienam duomenų poaibiui.

23 lentelė. Suformuotų scenarijų (taikant SimCLR bei augmentacijas) makro jautrumo įverčiai kiekvienam duomenų poaibiui.

Scenariius	Sleep-EDF-20	Sleep-EDF-60	Sleep-EDF-80	Sleep-EDF-100
	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)
ResNet <sub>1</sub>	0.714 (0.009)	0.696 (0.012)	0.65 (0.012)	0.554 (0.011)
ResNet <sub>2</sub>	0.711 (0.018)	0.704 (0.01)	0.666 (0.009)	0.559 (0.02)
SimCLR <sub>1</sub>	<b>0.738</b> (0.011)	0.716 (0.01)	0.669 (0.015)	0.565 (0.019)
SimCLR <sub>2</sub>	0.732 (0.015)	<b>0.718</b> (0.009)	<b>0.677</b> (0.01)	0.562 (0.038)

24 lentelė. Suformuotų scenarijų (taikant SimCLR bei augmentacijas) Koheno kapos įverčiai kiekvienam duomenų poaibiui.

Scenariius	arijus Sleep–EDF–20 Sleep–EDF–60		Sleep-EDF-80	Sleep-EDF-100
Jan Jan	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)
ResNet <sub>1</sub>	0.686 (0.015)	0.57 (0.01)	0.558 (0.013)	0.537 (0.018)
ResNet <sub>2</sub>	0.687 (0.023)	0.572 (0.014)	0.569 (0.009)	0.507 (0.02)
SimCLR <sub>1</sub>	0.698 (0.02)	0.585 (0.019)	0.574 (0.021)	0.512 (0.02)
SimCLR <sub>2</sub>	0.698 (0.023)	0.586 (0.014)	0.576 (0.023)	0.499 (0.051)

#### 6.2.2. Įverčiai kiekvienai miego fazei

25 lentelė. Suformuotų scenarijų (taikant SimCLR bei augmentacijas) F1 įverčiai kiekvienam duomenų poaibiui, atsibudimo fazė.

Scenariius	rijus <u>Sleep–EDF–20</u> <u>Sleep–ED</u>		Sleep-EDF-80	Sleep-EDF-100
	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)
ResNet <sub>1</sub>	0.78 (0.029)	0.841 (0.006)	0.776 (0.024)	<b>0.883</b> (0.013)
ResNet <sub>2</sub>	0.79 (0.04)	0.845 (0.011)	0.788 (0.017)	0.842 (0.016)
SimCLR <sub>1</sub>	0.801 (0.042)	0.849 (0.009)	0.791 (0.021)	0.841 (0.018)
SimCLR <sub>2</sub>	0.797 (0.047)	0.85 (0.009)	0.799 (0.025)	0.837 (0.043)

25 lentelėje pateikiami modelių F1 įverčiai atsibudimo fazei. Visiems (išskyrus Sleep–EDF– 100) duomenų poaibiams tiek augmentacijų, tiek išankstinio mokymo taikymas nei pagerino, nei pablogino šios fazės F1 įverčio. Įdomu tai, jog vyriausių asmenų poaibiui (Sleep-EDF-100) stebimas ryškus šios fazės F1 įverčių pablogėjimas taikant  $\text{ResNet}_2$  (-0.041) bei  $\text{SimCLR}_1$  (-0.042) strategijas. Taikant  $\text{SimCLR}_2$  scenarijų vidutinis F1 įvertis statistiškai reikšmingai nepablogėjo, tačiau labai pakilo jo standartinis nuokrypis (+0.03). Kitaip tariant, galima vertinti, jog bazinis modelis vyriausiems asmenims buvo pranašiausias klasifikuojant atsibudimo miego fazę.

26 lentelėje matoma, jog taikant išankstinį apmokymą ir augmentacijas (SimCLR<sub>2</sub> scenarijus), matomas statistiškai reikšmingas F1 įverčio pagerėjimas I miego fazei su Sleep-EDF-80 duomenimis (+0.036). Kitiem duomenų poaibiams toks pagerėjimas nėra pastebimas, tačiau nepastebimas ir pablogėjimas. 27 bei 28 lentelėse matoma, jog II bei III miego fazės F1 įverčiai nesiskyrė taikant duomenų augmentacijas ar išankstinį apmokymą.

Scenariius	Sleep–EDF–20	Sleep-EDF-60	Sleep-EDF-80	Sleep-EDF-100
	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)
ResNet <sub>1</sub>	0.27 (0.026)	0.312 (0.024)	0.286 (0.02)	0.228 (0.013)
ResNet <sub>2</sub>	0.255 (0.033)	0.326 (0.012)	0.309 (0.018)	0.214 (0.018)
SimCLR <sub>1</sub>	0.300 (0.022)	0.331 (0.016)	0.308 (0.035)	0.231 (0.013)
SimCLR <sub>2</sub>	0.281 (0.027)	0.337 (0.012)	<b>0.32</b> (0.014)	0.229 (0.02)

26 lentelė. Suformuotų scenarijų (taikant SimCLR bei augmentacijas) F1 įverčiai kiekvienam duomenų poaibiui, I fazė.

27 lentelė. Suformuotų scenarijų (taikant SimCLR bei augmentacijas) F1 įverčiai kiekvienam duomenų poaibiui, II fazė.

Scenariius	Sleep-EDF-20	Sleep-EDF-60	Sleep-EDF-80	Sleep-EDF-100
~ · · · · · · · J ····	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)
ResNet <sub>1</sub>	0.831 (0.011)	0.733 (0.016)	0.765 (0.01)	0.626 (0.039)
ResNet <sub>2</sub>	0.824 (0.008)	0.729 (0.009)	0.758 (0.006)	0.61 (0.027)
SimCLR <sub>1</sub>	0.825 (0.006)	0.74 (0.028)	0.76 (0.018)	0.628 (0.05)
SimCLR <sub>2</sub>	0.83 (0.007)	0.737 (0.019)	0.751 (0.019)	0.584 (0.068)

28 lentelė. Suformuotų scenarijų (taikant SimCLR bei augmentacijas) F1 įverčiai kiekvienam duomenų poaibiui, III fazė.

Scenariius	Sleep-EDF-20	Sleep-EDF-60	Sleep-EDF-80	Sleep-EDF-100
~	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)
ResNet <sub>1</sub>	0.836 (0.011)	0.562 (0.018)	0.619 (0.007)	0.66 (0.005)
ResNet <sub>2</sub>	0.818 (0.011)	0.552 (0.01)	0.622 (0.007)	0.662 (0.021)
SimCLR <sub>1</sub>	0.832 (0.004)	0.558 (0.027)	0.613 (0.008)	0.667 (0.036)
SimCLR <sub>2</sub>	0.842 (0.01)	0.555 (0.023)	0.617 (0.008)	0.662 (0.03)

Scenariius	Sleep-EDF-20	Sleep-EDF-60	Sleep-EDF-80	Sleep-EDF-100
~~~ <b>.</b>	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)	Vid. (Std.)
ResNet <sub>1</sub>	0.711 (0.011)	0.598 (0.026)	0.626 (0.024)	0.239 (0.076)
ResNet <sub>2</sub>	0.723 (0.019)	0.599 (0.024)	0.661 (0.043)	0.31 (0.094)
SimCLR <sub>1</sub>	<b>0.749</b> (0.014)	<b>0.657</b> (0.018)	0.671 (0.024)	0.282 (0.109)
SimCLR <sub>2</sub>	<b>0.749</b> (0.015)	<b>0.653</b> (0.018)	<b>0.674</b> (0.025)	0.321 (0.053)

29 lentelė. Suformuotų scenarijų (taikant SimCLR bei augmentacijas) F1 įverčiai kiekvienam duomenų poaibiui, REM fazė.

Ko gero didžiausias pokytis matomas vertinant įvairių strategijų naudojimą REM fazės klasifikavimui (29 lentelė). Tik su vyriausių asmenų duomenimis bazinis modelis nesiskyrė nuo kitų, tačiau su visais kitais duomenų poaibiais (Sleep–EDF–20, Sleep–EDF–60 bei Sleep–EDF–80) išankstinio apmokymo ir augmentacijų taikymas (SimCLR<sub>2</sub> scenarijus) darė teigiamą įtaką REM fazės F1 įverčiui. Dar daugiau, Sleep–EDF–20 bei Sleep–EDF–60 duomenims statistiškai reikšmingas F1 padidėjimas stebimas taikant tik išankstinį apmokymą (tai yra, nenaudojant papildomų duomenų augmentacijų, SimCLR<sub>1</sub> scenarijus).

#### 6.3. GradCAM taikymas

Žemiau pateikiami įdomesni ir aktualesni GradCAM radiniai. Autorė primena, jog žemiau pavaizduoti akyvacijų žemėlapiai yra visų testinių duomenų svarbių regionų žemėlapių suvidurkiniamas vienam konkrečiam k-grupės modeliui. Taip pat, svarbu nepamiršti, jog požymių žemėlapiai nereiškia, jog tame regione buvo pastebėtas tam tikras dažnių aktyvumas, tačiau vaizduoja, kad regionas yra svarbus (tai yra, gali būti, jog kaip tik jokio aktyvumo netgi nebuvo). Raudona spalva žymima svarbūs regionai, mėlyna – nesvarbūs. Priede D pateikta daugiau aktyvacijų žemėlapių pavyzdžių.



11 pav. Vidutiniai GradCAM aktyvacijų žemėlapiai visoms miego fazėms, Morse vilnelė (Sleep– EDF–20 duomenys, teisingai suklasifikuoti atvejai). Viršutinėje eilutėje matomi SqueezeNet modelio aktyvacijų žemėlapiai, apatinėje – ResNet–18.

11 pav. matoma, kurie regionai svarbūs teisingai klasifikuojant miego fazes Morse vilnele (Sleep–EDF–20 duomenimis). Įdomu, jog čia stebima, jog su labai žemais dažniais siejami re-

gionai (0.2–2.6 Hz) nebuvo itin svarbūs visoms miego fazėms, netgi III miego fazei (matomas tik nelabai ryškus pažaliavimas aplink žemų dažnių zoną).



12 pav. Vidutiniai GradCAM aktyvacijų žemėlapiai visoms miego fazėms, kompleskinė Morlet vilnelė (Sleep–EDF–20 duomenys, teisingai suklasifikuoti atvejai). Viršutinėje eilutėje matomas SqueezeNet modelio aktyvacijų žemėlapis, apatinėje – ResNet–18.

12 pav. matoma, kurios skalogramų vietos reikšmingos klasifikuojant miego fazes naudojant kompleksinę Morlet vilnelę. Matoma, jog esminis skirtumas tarp modelių toks – klasifikuojant ResNet–18 modeliu, daug svarbesnis paveiksliuko vidurys, o SqeezeNet modelis koncentruojasi į itin aukštus ar žemus dažnius. Apibendrinant abu modelius, galima pamatyti, jog nepriklausomai nuo modelio klasifikuojant miego fazes atsibudimo bei II fazei svarbus aukštesnių dažnių diapazonas, III – žemų, o REM – vidutinis.



13 pav. Vidutiniai GradCAM kontrafaktiniai aktyvacijų žemėlapiai neteisingai suklasifikuotiems REM fazės pavyzdžiams (Sleep–EDF–100 duomenys). Viršutinėje eilutėje matomi SqueezeNet modelio aktyvacijų žemėlapiai, apatinėje – ResNet–18.

Kadangi ypatingai blogai klasifikuojami pačių vyriausių asmenų duomenys (Sleep–EDF–100 poaibis), verta panagrinėti šį poaibį atskirai. 13 pav. matyti, kurios skalogramų vietos neigiamai prisideda prie modelio prognozės REM miego fazei, kuri pasižymėjo itin bloga klasifikacija (čia lyginama tik neteisingai suklasifikuoti atvejai). Matoma, jog nepriklausomai nuo pasirinkto modelio ar vilnelės kombinacijos, labiausiai neigiamai prisideda apačioje esančios paveiksliuko vietos (išskyrus su Mexican Hat vilnele taikant SqueezeNet architektūrą, tuomet nebuvo prognozei trukdančių regionų). Kitaip tariant, regionai, siejami su žemais dažniais, gali daryti įtaką netinkamai modelio prognozei klasifikuojant REM miego fazę.

Taip pat, įdomu patikrinti, kas atsitinka naudojant kontrastinį mokymą bei augmentacijas. Kadangi buvo pagerintas REM fazės klasifikacijos rezultatas, verta patikrinti, ar matomi reikšmingų vietų pokyčiai. 14 pav. matyti, jog naudojant tik duomenų augmentacijas (ResNet<sub>2</sub>) svarbūs regionai šiek tiek pakito (padidėjo mažesnių nei 5Hz dažnių svarba), o taikant SimCLR scenarijus, paveiksliuko kraštai tapo neprasmingi.



14 pav. Vidutiniai GradCAM aktyvacijų žemėlapiai teisingai suklasifikuotiems REM fazės pavyzdžiams (Sleep–EDF–20 duomenys) taikant kontrastinio mokymo/augmentacijų scenarijus.

### Išvados ir rekomendacijos

Šiame darbe analizuota, kaip miego fazes klasifikuoja KNT, kuomet naudojami tik vieno EEG kanalo duomenys. Vertinama vilnelės ir KNT architektūros įtaka, taikytas kontrastinio mokymo SimCLR karkasas bei duomenų augmentacijos. Kadangi palyginami 4 skirtingų amžiaus grupių rezultatai, šio darbo autorė nori priminti, jog tikslūs duomenų poaibių kodai bei amžiaus statistikos pateiktos 2 lentelėje.

Apibendrinant bazinių modelių rezultatus, galima teigti, jog geriausiai klasifikuojama jauniausių (25–34m. amžiaus grupė, vidutiniai makro F1 įverčiai kinta nuo 65.1 % iki 68.6 % nepriklausomai nuo parinktos vilnelės ar modelio architektūros), o blogiausiai – seniausių pacientų duomenys (85–101m. amžiaus grupė, nepriklausomai nuo parinktos vilnelės ar modelio architektūros vidutiniai makro F1 įverčiai kinta nuo 46.5% iki 52.7%). Taip pat, pastebima tendencija yra tai, jog dažnai geriausiai klasifikuojamos miego fazės yra tos, kurios yra geriausiai reprezentuotos duomenų po-aibyje (priklausomai nuo amžiaus grupės – atsibudimo bei II miego fazė). Visur prasčiausiai prognozuojama I miego fazė ir to buvo tikėtasi, nes ši tendencijai gerai žinoma iš miego fazių klasifikavimo tyrimų. Aptikti ir netikėti rezultatai: labai prasta REM miego fazės klasifikacija su vyriausių pacientų duomenimis (85–101m. amžiaus grupė). Tokių rezultatų nebuvo tikėtasi dėl to, jog ši amžiaus grupė nėra tiriama, todėl nebuvo žinoma, kad šios amžiaus grupės asmenims REM fazę sunku prognozuoti, taigi šis pastebėjimas – naujas.

Kadangi šiame darbe buvo lyginama tiek KNT architektūros parinkimo, tiek vilnelės įtaka miego fazių prognozei, atlikus rezultatų analizę, galima pateikti tokias išvadas:

- Dažnai sunku išskirti pranašiausią vilnelę, nes metrikos (tiek bendros, tiek atskirų miego fazių klasifikavimo F1) statistiškai reikšmingai nesiskyrė. Kuomet aptinkama pranaši vilnelė, tai būdavo kompleksinė Morlet ir/arba Morse vilnelės:
  - 1.1 Morse vilnelės pranašumas stebimas naudojant SqueezeNet architektūrą klasifikuojant 66–74m. asmenų duomenis. Lyginant su kitomis vilnelėmis, stebimas didžiausias makro jautrumas bei preciziškumas bei didžiausias I miego fazės F1 (13 bei 15 lentelės).
  - 1.2 Kompleksinės Morlet vilnelės pranašumas aptiktas 25–34m. asmenų grupei taikant ResNet–18 architektūrą (vertinant makro metrikas bei II miego fazės F1, 6 bei 8 lentelės). Papildomai, su šia vilnele geriausiai klasifikuojama šios amžiaus grupės pacientų II miego fazė taikant SqueezeNet architektūrą (7 lentelė) bei klasifikuojant 85–101m. amžiaus pacientų III miego fazę su ResNet–18 architektūra (20 lentelė).
  - 1.3 Rasta, jog Morse bei kompleksinė Morlet vienodai geros, kuomet naudojama ResNet-18 architektūra: 50–60m. amžiaus pacientams vertinant makro jautrumo metrikas bei I ir REM fazės F1 įverčius (10 bei 12 lentelės), taip pat 66–74m. asmenims vertinant visas bendras metrikas bei I ir II miego fazių F1 (14 bei 16 lentelės).
- 2. Architektūros parinkimo svarbą galima apibendrinti taip:
  - 2.1 SqueezeNet modelis naudingas klasifikuojant 66–74m. amžiaus grupės miego fazes taikant Morse vilnelę (lyginant su ResNet–18, geresnės visos bendros metrikos bei I miego fazės F1, 13 bei 15 lentelės). Taip pat šis modelis naudingas ir kai kurioms kitoms vilnelėms, tačiau rezultatai vis tiek nėra geresni, negu taikant kitas vilneles ar ResNet–18 architektūrą (žr. į 7, 9, 11 bei 13 lenteles).

- 2.2 ResNet–18 modelis pagerina Morse (didesnis makro preciziškumas bei Koheno kapa, taip pat I miego fazės F1), kompleksinės Morlet (didesnis II fazės F1) bei Mexican Hat (didesnė Koheno kapa bei II fazės F1) rezultatus 85–101m. pacientams (18 bei 20 lentelės).
- 2.3 Kitais atvejais, architektūros rezultatais nesiskyrė, taigi net su itin mažu SqueezeNet modeliu galima pakankamai gerai klasifikuoti miego fazes. Kita vertus, su ResNet–18 architektūra dažnai stebimi mažesni metrikų standartiniai nuokrypiai, kas reiškia jog ši architektūra stabilesnė.

Pritaikius duomenų augmentavimo bei kontrastinio mokymo scenarijus (pilnas aprašymas pateikiamas 5.4), stebimi tokie rezultatai:

- Nors makro F1, preciziškumo bei Koheno kapos įverčiai taikant SimCLR ir/arba papildomas augmentacijas nepakito, beveik visiems pacientams pagerintas makro jautrumo įvertis. 25–34m. pacientų grupei taikant tik SimCLR, vidutinis makro jautrumas pagerintas 2.4%. 50–60m. bei 66–74m. amžiaus pacientams SimCLR bei augmentacijų taikymas padidino jautrumą per atitinkamai 2.2% bei 2.7%.
- Beveik visiems pacientams SimCLR taikymas pagerino REM miego fazės F1 įverčius. 25– 34m. amžiaus grupei stebimas 3.8% pagerėjimas taikant tiek SimCLR, tiek papildomomas augmentacijas, 50–60m. amžiaus grupei matomas 5.9% pagerėjimas taikant tik SimCLR, o taikant SimCLR su papildomomis augmentacijomis, F1 padidėja per 5.5%. 66–74m. grupei SimCLR su papildomis augmentacijomis pagerina REM fazės F1 per 4.8%.
- 3. Dar daugiau, 66–74m. amžiaus pacientams pastebimas ir I miego fazės F1 pagerėjimas taikant SimCLR ir papildomas augmentacijas (+3.4%).
- 4. Priešingai nei kitoms amžiaus grupėms, 85–101m. amžiaus pacientams, REM fazės klasifikavimas nepagerėjo taikant SimCLR ir/arba duomenų augmentacijas. Dar daugiau, šių strategijų taikymas pablogina atsibudimo fazės F1. Konkrečiai, taikant tik duomenų augmentacijas, atsibudimo fazės F1 pablogėja per 4.1%. Taikant tik SimCLR, F1 pablogėja per 4.2%. Taikant abi strategijas F1 įvertis statistiškai reikšmingai nepablogėja, tačiau padidėja šios metrikos standartinis nuokrypis (+3%). Kitaip tariant, net ir šis scenarijus sukelia nenorimą modelio nestabilumo efektą. Autorės nuomone, tokius rezultatus galima sieti su nekorektiškomis duomenų augmentacijomis būtent šios amžiaus grupės pacientams.

Papildomai, šiame darbe buvo pritaikytas GradCAM metodas, kuriuo siekta pateikti vizualinius modelių sprendimo priėmimo paaiškinimus. Nors taikant šį metodą nebuvo siekiama pateikti kategoriškų išvadų, randama, jog GradCAM taikymas, o ypač apibendrintų aktyvacijos žemėlapių, gali būti naudingas duomenų analizės įrankis. Tarkime, šiame darbe aptikta, jog kuomet 85–101m. amžiaus asmenims neteisingai suklasifikuojami REM fazės atvejai, neigiamai prie tinkamos prognozės gali prisidėti žemesnių dažnių diapazonas.

Apibendrinus, galima suformuluoti tokias rekomendacijas – vilnelės parinkimas gali būti svarbus, tačiau dažniausiai naudojama Morse vilnelė yra viena iš geresnių taikomų transformacijų. Taip pat rasta, jog kompleksinė Morlet vilnelė (kuri tyrimuose taikoma labai retai) yra ne ką prastesnė ir tam tikrais atvejais netgi pranašesnė. Taip pat, darbe naudotas SimCLR bei augmentacijų taikymas gali būti naudingas gerinant miego fazių klasifikaciją, tačiau tobulinant šį metodą, labai svarbu atsižvelgti į pacientų amžių.

# Ateities tyrimų planas

Atsižvelgus į gautus rezultatus, ateities tyrimuose siūloma:

- Atlikti išsamesnę analizę su vyresnio amžiaus asmenimis. Konkrečiai, kadangi šiame darbe rasta, jog prasčiausia klasifikacija buvo su 85–101m. amžiaus pacientų duomenimis, o taip pat naudotos kontrastinio mokymo/augmentacijų taikymo metodikos netgi pablogino kai kurių miego fazių klasifikavimą, reikalinga korektiškų augmentacijų paieška. Taip pat, šiai amžiaus grupei galimai reikia surinkti daugiau duomenų.
- Taip pat, reikalingas išsamesnis augmentacijų palyginimo tyrimas. Didžiausias šio darbo trūkumas – nebuvo palyginta, ar geriau augmentuoti skalogramą, ar signalą. Ateities darbuose siūloma atlikti tokį tyrimą.
- 3. Augmentuojant skalogramas šiame tyrime naudojamos gana paprastos augmentacijos, tačiau visai perspektyvi sritis generatyviniais neuroniniais tinklais kuriamos augmentacijos. Būtų aktualu išbandyti ir šią strategiją.
- 4. Papildomai, vienas iš šio darbo trūkumų 5 grupių kryžminės validacijos naudojimas. 5 grupių kryžminė validacija buvo atliekama siekiant taupyti laiką ir skaičiavimo resursus, tačiau tai reiškia, jog lyginant klasifikavimo rezultatus, galima sutikti daug išskirčių, kurios iškreipia palyginimą. Ateityje siūloma daryti tokią validaciją, kurios validavimo duomenimis yra tik vienas pacientas.
- 5. Tyrime taikyti tik KNT architektūros neuroniniai tinklai, tačiau vaizdų analizėje taikomi ir kitokie metodai. Gali būti, kad kitokių neuroninių tinklų architektūrų taikymas gali pagerinti rezultatą. Tarkime, miego fazių skalogramoms klasifikuoti kuriami hibridiniai KNT (kuriuose yra, tarkime, dėmesio sutelkimo sluoksniai arba rekurentiniai vienetai).
- 6. Atrodo, jog kontrastinio mokymo metodikos yra perspektyvios gerinti miego fazių klasifikavimą. Vis tik šiame darbe su SimCLR karkasu buvo gerinama tik viena architektūros ir vilnelės kombinacija. Ateityje naudojant kontrastinį mokymą, galima išbandyti kitas kombinacijas, o taip pat netgi naudoti kitokį kontrastinio mokymosi karkasą.
- 7. Tyrime klasifikacijos mokymo metu naudota kryžminės entropijos funkcija (su klasių svoriais) galimai nėra optimaliausia nuostolių funkcija. Siekiant geresnio mokymosi, ateityje galima išbandyti ir kitas netekties funkcijas (pavyzdžiui, židinio nuostolių funkciją).

### Literatūros šaltiniai

- [1] A. Akan and O. K. Cura. Time–frequency signal processing: Today and future. *Digital Signal Processing*, 119:103216, 2021.
- [2] K. Alomar, H. I. Aysel, and X. Cai. Data augmentation in classification and segmentation: A survey and new strategies. *Journal of Imaging*, 9(2), 2023.
- [3] M. Baumert, S. Hartmann, and H. Phan. Automatic sleep staging for the young and the old– evaluating age bias in deep learning. *Sleep Medicine*, 107:18–25, 2023.
- [4] G. Bradski. The OpenCV Library. Dr. Dobb's Journal of Software Tools, 2000.
- [5] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.
- [6] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.
- [7] C. O. da Costa-Luis. tqdm: A fast, extensible progress meter for python and cli. *Journal of Open Source Software*, 4(37):1277, 2019.
- [8] H. Danker-Hopfe, D. Kunz, G. Gruber, G. Klösch, J. L. Lorenzo, S. L. Himanen, B. Kemp, T. Penzel, J. Röschke, H. Dorn, A Schlögl, E. Trenker, and G. Dorffner. Interrater reliability between scorers from eight european sleep laboratories in subjects with different sleep disorders. *Journal of sleep research*, 13(1):63–69, 2004.
- [9] A. Demir, I. Khalil, and B. Kiziltan. Eeg-next: A modernized convnet for the classification of cognitive activity from eeg. *arXiv preprint arXiv:2212.04951*, 2022.
- [10] N. S. Detlefsen, J. Borovec, J. Schock, A. Harsh, T. Koker, L. Di Liello, D. Stancl, C. Quan, M. Grechkin, and W. Falcon. Torchmetrics - measuring reproducibility in pytorch, 2022. Apache-2.0 License.
- [11] J. Eisele, A. Gerlach, M. Maeder, and S. Marburg. Convolutional neural network with data augmentation for object classification in automotive ultrasonic sensing. *The Journal of the Acoustical Society of America*, 153(4):2447–2447, 2023.
- [12] H. ElMoaqet, M. Eid, M. Ryalat, and T. Penzel. A deep transfer learning framework for sleep stage classification with single-channel eeg signals. *Sensors*, 22(22):8826, 2022.
- [13] A. A. Gharbali, S. Najdi, and J. M. Fonseca. Transfer learning of spectrogram image for automatic sleep stage classification. In *Image Analysis and Recognition: 15th International Conference, ICIAR 2018, Póvoa de Varzim, Portugal, June 27–29, 2018, Proceedings 15*, pages 522–528. Springer, 2018.
- [14] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101.

- [15] N. Gozzi, E. Giacomello, M. Sollini, M. Kirienko, A. Ammirabile, P. Lanzi, D. Loiacono, and A. Chiti. Image embeddings extracted from cnns outperform other transfer learning approaches in classification of chest radiographs. *Diagnostics*, 12(9):2084, 2022.
- [16] K. K. Gulia and V. M. Kumar. Sleep disorders in the elderly: a growing challenge. *Psycho-geriatrics*, 18(3):155–165, 2018.
- [17] S. Haghayegh, K. Hu, K. Stone, S. Redline, and E. Schernhammer. Automated sleep stages classification using convolutional neural network from raw and time-frequency electroencephalogram signals: systematic evaluation study. *Journal of Medical Internet Research*, 25:e40211, 2023.
- [18] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. Fernández del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.
- [19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770– 778, 2016.
- [20] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.
- [21] F. N. Iandola. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and < 0.5 mb model size. arXiv preprint arXiv:1602.07360, 2016.
- [22] C. Iber, S. Ancoli-Israel, A.L. Chesson, and S. Quan. The aasm manual for the scoring of sleep and associated events: Rules, terminology and technical specifications. Westchester, IL: American Academy of Sleep Medicine, 01 2007.
- [23] P. Jadhav, G. Rajguru, D. Datta, and S. Mukhopadhyay. Automatic sleep stage classification using time-frequency images of cwt and transfer learning using convolution neural network. *Biocybernetics and Biomedical Engineering*, 40(1):494–504, 2020.
- [24] X. Jiang, J. Zhao, B. Du, and Z. Yuan. Self-supervised contrastive learning for eeg-based sleep staging. In 2021 International Joint Conference on Neural Networks (IJCNN), pages 1–8. IEEE, 2021.
- [25] B. Kemp, A. H. Zwinderman, B. Tuk, H. A. C. Kamphuisen, and J. J. L. Oberye. Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the eeg. *IEEE Transactions on Biomedical Engineering*, 47(9):1185–1194, 2000.
- [26] M. Kisieliūtė. Mokslo tiriamojo darbo projektas, 6 2024.
- [27] C. E. Kuo, G. T. Chen, and P. Y. Liao. An eeg spectrogram-based automatic sleep stage scoring method via data augmentation, ensemble convolution neural network, and expert knowledge. *Biomedical Signal Processing and Control*, 70:102981, 2021.

- [28] P. H. Le-Khac, G. Healy, and A. F. Smeaton. Contrastive representation learning: A framework and review. *Ieee Access*, 8:193907–193934, 2020.
- [29] G. R. Lee, R. Gommers, F. Waselewski, K. Wohlfahrt, and A. OLeary. Pywavelets: A python package for wavelet analysis. *Journal of Open Source Software*, 4(36):1237, 2019.
- [30] S. Lee, Y. Yu, S. Back, H. Seo, and K. Lee. Sleepyco: Automatic sleep scoring with feature pyramid and contrastive learning. *Expert Systems with Applications*, 240:122551, 2024.
- [31] D. Li, Y. Ruan, F. Zheng, Y. Su, and Q. Lin. Fast sleep stage classification using cascaded support vector machines with single-channel eeg signals. *Sensors*, 22(24), 2022.
- [32] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, 33(12):6999–7019, 2021.
- [33] TorchVision maintainers and contributors. Torchvision: Pytorch's computer vision library. https://github.com/pytorch/vision, 2016.
- [34] W. McKinney. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, pages 51 56, 2010.
- [35] J. Muradeli. ssqueezepy. https://github.com/OverLordGoldDragon/ssqueezepy/, 2020.
- [36] H. Nahrstaedt, S. Kern, L. Cerina, S. Appelhoff, D. T.H. Kao, C. Franklin, S. Clarke, J. Zitting, D. Ojeda, C. Boulay, O. Moore, and D. Kosachev. holgern/pyedflib: v0.1.23, November 2021.
- [37] J. Opitz. A closer look at classification evaluation metrics and a critical reflection of common evaluation practice. *Transactions of the Association for Computational Linguistics*, 12:820– 836, 06 2024.
- [38] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv*:1904.08779, 2019.
- [39] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.
- [40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [41] L. Petkevičius. Giliojo mokymosi metodų (deep learning) sąvokų žodynėlis. https://github. com/linas-p/ML-AI-2-LT.
- [42] Z. Rabin, J. Davis, B. Lewis, and M. Scherreik. Overfitting in contrastive learning?, 2024.

- [43] A. Rechtschaffen and A. Kales. A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects. 1968. Accessed: 2024-06-09.
- [44] M. R. Rezaei-Dastjerdehei, A. Mijani, and E. Fatemizadeh. Addressing imbalance in multilabel classification using weighted cross entropy loss function. In 2020 27th national and 5th international iranian conference on biomedical engineering (ICBME), pages 333–338. IEEE, 2020.
- [45] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016.
- [46] P. Umesh. Image processing in python. CSI Communications, 23, 2012.
- [47] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.
- [48] M. L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021.
- [49] Y. Wei, Y. Zhu, Y. Zhou, X. Yu, and Y. Luo. Automatic sleep staging based on contextual scalograms and attention convolution neural network using single-channel eeg. *IEEE Journal of Biomedical and Health Informatics*, 28(2):801–811, 2024.
- [50] X. Ying. An overview of overfitting and its solutions. In *Journal of physics: Conference series*, volume 1168, page 022022. IOP Publishing, 2019.

# Priedai

Dokumentą sudaro 4 priedai: priede A pateikti miego fazių pavyzdžiai (iš naudotų duomenų), priede B ir C pateikti modelių metrikų medianų įverčiai, o priede D pateikti papildomi GradCAM aktyvacijų žemėlapiai.

### A. EEG Fpz-Cz kanalo fragmentai skirtingų miego fazių metu

Skirtingo miego fazių EEG (Fpz-Cz kanalo) signalo fragmentai: abscisių ašyje nurodytas laikas, ordinačių - įtampa ( $\mu$ V). Nuo viršaus į apačią vaizduojama: atsibudimo, I, II, III bei REM fazės.



# B. Bazinių modelių metrikų medianų įverčiai

Vilnelė	Jautrumas	Preciziškumas	<b>F1</b>	К
	Med.(MAD)	Med. (MAD)	Med.(MAD)	Med.(MAD)
Morse	0.711 (0.017)	0.686 (0.008)	0.672 (0.015)	0.656 (0.008)
K. Morlet	0.728 (0.016)	0.705 (0.009)	0.695 (0.016)	0.692 (0.007)
K. Gauso	0.696 (0.018)	0.676 (0.011)	0.673 (0.018)	0.66 (0.021)
Mex. Hat	0.704 (0.008)	0.679 (0.013)	0.676 (0.01)	0.674 (0.002)

30 lentelė. SqueezeNet architektūros makro statistikų (jautrumo, preciziškumo bei F1) ir Koheno kapos medianų įverčiai Sleep–EDF–20 duomenims.

31 lentelė. ResNet–18 architektūros makro statistikų (jautrumo, preciziškumo bei F1) ir Koheno kapos medianų įverčiai Sleep–EDF–20 duomenims.

Vilnelė	Jautrumas	Preciziškumas	F1	К
,	Med.(MAD)	Med. (MAD)	Med.(MAD)	Med.(MAD)
Morse	0.7 (0.005)	0.673 (0.002)	0.666 (0.006)	0.658 (0.005)
K. Morlet	0.718 (0.003)	0.697 (0.005)	0.69 (0.005)	0.686 (0.013)
K. Gauso	0.674 (0.002)	0.657 (0.001)	0.65 (0.003)	0.647 (0.01)
Mex. Hat	0.676 (0.015)	0.66 (0.009)	0.652 (0.011)	0.647 (0.015)

32 lentelė. SqueezeNet architektūros F1 medianų įverčiai kiekvienai miego fazei, Sleep–EDF–20 duomenims.

Vilnelė	Atsibudimas	I fazė	II fazė	III fazė	REM fazė
	Med. (MAD)				
Morse	0.77 (0.017)	0.26 (0.026)	0.811 (0.012)	0.796 (0.015)	0.711 (0.019)
K. Morlet	0.777 (0.009)	0.264 (0.051)	0.842 (0.008)	0.835 (0.004)	0.742 (0.015)
K. Gauso	0.754 (0.052)	0.246 (0.039)	0.8 (0.013)	0.81 (0.013)	0.689 (0.013)
Mex. Hat	0.764 (0.006)	0.258 (0.006)	0.802 (0.021)	0.824 (0.012)	0.702 (0.013)

33 lentelė. ResNet–18 architektūros F1 medianų įverčiai kiekvienai miego fazei, Sleep–EDF–20 duomenims.

Vilnelė	Atsibudimas	I fazė	II fazė	III fazė	REM fazė
	Med. (MAD)	Med. (MAD)	Med. (MAD)	Med. (MAD)	Med. (MAD)
Morse	0.792 (0.025)	0.248 (0.008)	0.801 (0.0003)	0.804 (0.007)	0.696 (0.011)
K. Morlet	0.76 (0.014)	0.271 (0.012)	0.832 (0.007)	0.841 (0.006)	0.714 (0.009)
K. Gauso	0.774 (0.019)	0.223 (0.014)	0.788 (0.003)	0.812 (0.004)	0.658 (0.007)
Mex. Hat	0.761 (0.03)	0.218 (0.005)	0.8 (0.006)	0.823 (0.003)	0.650 (0.022)

Vilnelė	Jautrumas	Preciziškumas	<b>F</b> 1	К
( micic	Med.(MAD)	Med. (MAD)	Med.(MAD)	Med.(MAD)
Morse	0.703 (0.013)	0.61 (0.006)	0.611 (0.01)	0.569 (0.01)
K. Morlet	0.692 (0.004)	0.603 (0.004)	0.609 (0.014)	0.571 (0.01)
K. Gauso	0.676 (0.0003)	0.587 (0.007)	0.595 (0.004)	0.552 (0.003)
Mex. Hat	0.678 (0.018)	0.591 (0.014)	0.597 (0.008)	0.547 (0.002)

34 lentelė. SqueezeNet architektūros makro statistikų (jautrumo, preciziškumo bei F1) ir Koheno kapos medianų įverčiai Sleep–EDF–60 duomenims.

35 lentelė. ResNet–18 architektūros makro statistikų (jautrumo, preciziškumo bei F1) ir Koheno kapos medianų įverčiai Sleep–EDF–60 duomenims.

Vilnelė	Jautrumas	Preciziškumas	<b>F1</b>	К
, millione	Med.(MAD)	Med. (MAD)	Med.(MAD)	Med.(MAD)
Morse	0.69 (0.008)	0.594 (0.006)	0.606 (0.006)	0.566 (0.009)
K. Morlet	0.695 (0.009)	0.598 (0.01)	0.613 (0.007)	0.573 (0.007)
K. Gauso	0.656 (0.008)	0.578 (0.005)	0.59 (0.008)	0.546 (0.015)
Mex. Hat	0.656 (0.002)	0.575 (0.006)	0.585 (0.012)	0.547 (0.016)

36 lentelė. SqueezeNet architektūros F1 medianų įverčiai kiekvienai miego fazei, Sleep–EDF–60 duomenims.

Vilnelė	Atsibudimas	I fazė	II fazė	III fazė	REM fazė
,	Med. (MAD)				
Morse	0.823 (0.002)	0.319 (0.003)	0.746 (0.023)	0.56 (0.034)	0.646 (0.002)
K. Morlet	0.829 (0.014)	0.318 (0.005)	0.747 (0.029)	0.563 (0.051)	0.599 (0.019)
K. Gauso	0.839 (0.002)	0.305 (0.005)	0.723 (0.009)	0.571 (0.032)	0.545 (0.008)
Mex. Hat	0.826 (0.006)	0.296 (0.008)	0.721 (0.013)	0.576 (0.013)	0.547 (0.044)

37 lentelė. ResNet–18 architektūros F1 medianų įverčiai kiekvienai miego fazei, Sleep–EDF–60 duomenims.

Vilnelė	Atsibudimas I fazė		II fazė	III fazė	REM fazė
	Med. (MAD)	Med. (MAD)	Med. (MAD)	Med. (MAD)	Med. (MAD)
Morse	0.827 (0.006)	0.295 (0.003)	0.74 (0.006)	0.549 (0.014)	0.606 (0.005)
K. Morlet	0.84 (0.003)	0.305 (0.018)	0.733 (0.003)	0.565 (0.018)	0.598 (0.022)
K. Gauso	0.837 (0.004)	0.272 (0.006)	0.735 (0.013)	0.6 (0.017)	0.504 (0.012)
Mex. Hat	0.831 (0.004)	0.276 (0.002)	0.739 (0.009)	0.594 (0.011)	0.517 (0.001)

Vilnelė	Jautrumas	Preciziškumas	<b>F1</b>	К
	Med.(MAD)	Med. (MAD)	Med.(MAD)	Med.(MAD)
Morse	0.666 (0.01)	0.62 (0.011)	0.634 (0.008)	0.575 (0.008)
K. Morlet	0.643 (0.014)	0.594 (0.017)	0.607 (0.015)	0.547 (0.016)
K. Gauso	0.631 (0.019)	0.57 (0.016)	0.593 (0.012)	0.506 (0.039)
Mex. Hat	0.634 (0.012)	0.585 (0.006)	0.592 (0.005)	0.517 (0.016)

38 lentelė. SqueezeNet architektūros makro statistikų (jautrumo, preciziškumo bei F1) ir Koheno kapos medianų įverčiai Sleep–EDF–80 duomenims.

39 lentelė. ResNet–18 architektūros makro statistikų (jautrumo, preciziškumo bei F1) ir Koheno kapos medianų įverčiai Sleep–EDF–80 duomenims.

Vilnelė	Jautrumas	Preciziškumas	<b>F1</b>	К
, millione	Med.(MAD)	Med. (MAD)	Med.(MAD)	Med.(MAD)
Morse	0.638 (0.015)	0.591 (0.013)	0.597 (0.01)	0.539 (0.007)
K. Morlet	0.65 (0.011)	0.602 (0.001)	0.612 (0.007)	0.56 (0.013)
K. Gauso	0.604 (0.005)	0.57 (0.005)	0.577 (0.004)	0.494 (0.008)
Mex. Hat	0.612 (0.006)	0.572 (0.003)	0.579 (0.007)	0.502 (0.004)

40 lentelė. SqueezeNet architektūros F1 įverčiai kiekvienai miego fazei, Sleep–EDF–80 duomenims.

Vilnelė	Atsibudimas	I fazė	II fazė	III fazė	REM fazė
,	Med. (MAD)				
Morse	0.79 (0.008)	0.35 (0.005)	0.769 (0.005)	0.605 (0.009)	0.646 (0.034)
K. Morlet	0.757 (0.039)	0.274 (0.008)	0.755 (0.009)	0.619 (0.006)	0.597 (0.017)
K. Gauso	0.768 (0.039)	0.238 (0.018)	0.7 (0.009)	0.621 (0.005)	0.563 (0.041)
Mex. Hat	0.774 (0.016)	0.263 (0.025)	0.728 (0.001)	0.637 (0.001)	0.609 (0.019)

41 lentelė. ResNet–18 architektūros F1 medianų įverčiai kiekvienai miego fazei, Sleep–EDF–80 duomenims.

Vilnelė	Atsibudimas	I fazė	II fazė	III fazė	REM fazė
	Med. (MAD)				
Morse	0.775 (0.008)	0.289 (0.009)	0.762 (0.001)	0.593 (0.003)	0.561 (0.015)
K. Morlet	0.785 (0.015)	0.291 (0.014)	0.767 (0.004)	0.618 (0.003)	0.63 (0.013)
K. Gauso	0.76 (0.008)	0.253 (0.011)	0.722 (0.008)	0.612 (0.009)	0.548 (0.007)
Mex. Hat	0.764 (0.023)	0.248 (0.013)	0.72 (0.006)	0.627 (0.003)	0.566 (0.007)

Vilnelė	Jautrumas	Preciziškumas	<b>F1</b>	К
	Med.(MAD)	Med. (MAD)	Med.(MAD)	Med.(MAD)
Morse	0.528 (0.027)	0.47 (0.016)	0.459 (0.018)	0.465 (0.021)
K. Morlet	0.526 (0.043)	0.496 (0.035)	0.476 (0.055)	0.505 (0.04)
K. Gauso	0.521 (0.01)	0.5 (0.024)	0.496 (0.01)	0.5 (0.001)
Mex. Hat	0.517 (0.025)	0.487 (0.014)	0.487 (0.014)	0.484 (0.018)

42 lentelė. SqueezeNet architektūros makro statistikų (jautrumo, preciziškumo bei F1) ir Koheno kapos medianų įverčiai Sleep–EDF–100 duomenims.

43 lentelė. ResNet–18 architektūros makro statistikų (jautrumo, preciziškumo bei F1) ir Koheno kapos medianų įverčiai Sleep–EDF–100 duomenims.

Vilnelė	Jautrumas	Preciziškumas	F1	К
, miere	Med.(MAD)	Med. (MAD)	Med.(MAD)	Med.(MAD)
Morse	0.549 (0.009)	0.525 (0.015)	0.521 (0.016)	0.525 (0.017)
K. Morlet	0.551 (0.002)	0.530 (0.011)	0.531 (0.004)	0.529 (0.006)
K. Gauso	0.516 (0.007)	0.520 (0.009)	0.514 (0.005)	0.507 (0.008)
Mex. Hat	0.536 (0.011)	0.524 (0.002)	0.518 (0.009)	0.54 (0.005)

44 lentelė. SqueezeNet architektūros F1 medianų įverčiai kiekvienai miego fazei, Sleep–EDF–100 duomenims.

Vilnelė	Atsibudimas	I fazė	II fazė	III fazė	REM fazė
,	Med. (MAD)				
Morse	0.813 (0.01)	0.202 (0.002)	0.466 (0.028)	0.59 (0.042)	0.209 (0.08)
K. Morlet	0.866 (0.04)	0.206 (0.018)	0.488 (0.084)	0.588 (0.069)	0.174 (0.065)
K. Gauso	0.871 (0.009)	0.193 (0.004)	0.559 (0.05)	0.61 (0.04)	0.262 (0.021)
Mex. Hat	0.851 (0.021)	0.205 (0.025)	0.568 (0.025)	0.638 (0.031)	0.169 (0.024)

45 lentelė. ResNet–18 architektūros F1 medianų įverčiai kiekvienai miego fazei, Sleep–EDF–100 duomenims.

Vilnelė	Atsibudimas	ibudimas I fazė		III fazė	REM fazė
,	Med. (MAD)	Med. (MAD)	Med. (MAD)	Med. (MAD)	Med. (MAD)
Morse	0.867 (0.005)	0.239 (0.006)	0.5s66 (0.04)	0.605 (0.017)	0.283 (0.015)
K. Morlet	0.886 (0.003)	0.222 (0.008)	0.609 (0.02)	0.66 (0.003)	0.253 (0.025)
K. Gauso	0.875 (0.011)	0.202 (0.003)	0.581 (0.009)	0.625 (0.023)	0.28 (0.026)
Mex. Hat	0.876 (0.003)	0.222 (0.008)	0.658 (0.017)	0.637 (0.002)	0.196 (0.027)

# C. SimCLR bei augmentacijų scenarijų metrikų medianų įverčiai

Scenariius	Sleep-EDF-20	Sleep-EDF-60	Sleep-EDF-80	Sleep-EDF-100
~ · · · · · · · J · · ·	Med. (MAD)	Med. (MAD)	Med. (MAD)	Med. (MAD)
ResNet <sub>1</sub>	0.69 (0.005)	0.613 (0.007)	0.612 (0.007)	0.531 (0.004)
ResNet <sub>2</sub>	0.689 (0.015)	0.609 (0.011)	0.624 (0.002)	0.537 (0.005)
SimCLR <sub>1</sub>	0.695 (0.003)	0.63 (0.005)	0.634 (0.009)	0.523 (0.015)
SimCLR <sub>2</sub>	0.694 (0.004)	0.629 (0.009)	0.634 (0.005)	0.54 (0.017)

46 lentelė. Suformuotų scenarijų (taikant SimCLR bei augmentacijas) makro F1 medianų įverčiai kiekvienam duomenų poaibiui.

47 lentelė. Suformuotų scenarijų (taikant SimCLR bei augmentacijas) makro preciziškumo medianų įverčiai kiekvienam duomenų poaibiui.

Scenariius	Sleep-EDF-20	Sleep-EDF-60	Sleep-EDF-80	Sleep-EDF-100
	Med. (MAD)	Med. (MAD)	Med. (MAD)	Med. (MAD)
ResNet <sub>1</sub>	0.697 (0.005)	0.598 (0.01)	0.602 (0.001)	0.53 (0.011)
ResNet <sub>2</sub>	0.694 (0.01)	0.596 (0.011)	0.608 (0.004)	0.543 (0.005)
SimCLR <sub>1</sub>	0.703 (0.002)	0.624 (0.002)	0.618 (0.008)	0.537 (0.018)
SimCLR <sub>2</sub>	0.705 (0.006)	0.621 (0.007)	0.617 (0.004)	0.551 (0.017)

48 lentelė. Suformuotų scenarijų (taikant SimCLR bei augmentacijas) makro jautrumo medianų įverčiai kiekvienam duomenų poaibiui.

Scenariius	Sleep-EDF-20	Sleep-EDF-60	Sleep-EDF-80	Sleep-EDF-100
2 <b></b> - J <b>-</b> - 2	Med. (MAD)	Med. (MAD)	Med. (MAD)	Med. (MAD)
ResNet <sub>1</sub>	0.718 (0.003)	0.695 (0.009)	0.65 (0.011)	0.551 (0.043)
ResNet <sub>2</sub>	0.72 (0.008)	0.701 (0.009)	0.668 (0.006)	0.563 (0.015)
SimCLR <sub>1</sub>	0.736 (0.002)	0.716 (0.008)	0.68 (0.002)	0.569 (0.018)
SimCLR <sub>2</sub>	0.728 (0.005)	0.72 (0.009)	0.68 (0.004)	0.576 (0.009)

49 lentelė. Suformuotų scenarijų (taikant SimCLR bei augmentacijas) Koheno kapos medianų įverčiai kiekvienam duomenų poaibiui.

Scenariius	Sleep-EDF-20	Sleep-EDF-60	Sleep-EDF-80	Sleep-EDF-100
2 <b></b> - J <b>-</b> - 2	Med. (MAD)	Med. (MAD)	Med. (MAD)	Med. (MAD)
ResNet <sub>1</sub>	0.686 (0.013)	0.573 (0.007)	0.56 (0.013)	0.529 (0.006)
ResNet <sub>2</sub>	0.686 (0.016)	0.572 (0.013)	0.564 (0.005)	0.517 (0.008)
SimCLR <sub>1</sub>	0.689 (0.008)	0.586 (0.013)	0.575 (0.006)	0.516 (0.018)
SimCLR <sub>2</sub>	0.689 (0.003)	0.589 (0.009)	0.584 (0.018)	0.519 (0.017)

Scenarijus	Sleep-EDF-20	Sleep-EDF-60	Sleep-EDF-80	Sleep-EDF-100
	Med. (MAD)	Med. (MAD)	Med. (MAD)	Med. (MAD)
ResNet <sub>1</sub>	0.76 (0.004)	0.839 (0.003)	0.785 (0.015)	0.886 (0.003)
ResNet <sub>2</sub>	0.779 (0.042)	0.841 (0.011)	0.795 (0.009)	0.852 (0.003)
SimCLR <sub>1</sub>	0.789 (0.006)	0.848 (0.006)	0.789 (0.011)	0.839 (0.013)
SimCLR <sub>2</sub>	0.783 (0.009)	0.849 (0.005)	0.795 (0.02)	0.849 (0.017)

50 lentelė. Suformuotų scenarijų (taikant SimCLR bei augmentacijas) F1 medianų įverčiai kiekvienam duomenų poaibiui, atsibudimo fazė.

51 lentelė. Suformuotų scenarijų (taikant SimCLR bei augmentacijas) F1 medianų įverčiai kiekvienam duomenų poaibiui, I fazė.

Scenarijus	Sleep-EDF-20	Sleep-EDF-60	Sleep-EDF-80	Sleep-EDF-100
	Med. (MAD)	Med. (MAD)	Med. (MAD)	Med. (MAD)
ResNet <sub>1</sub>	0.271 (0.012)	0.305 (0.018)	0.291 (0.014)	0.222 (0.008)
ResNet <sub>2</sub>	0.266 (0.032)	0.327 (0.01)	0.309 (0.009)	0.215 (0.012)
SimCLR <sub>1</sub>	0.298 (0.015)	0.339 (0.009)	0.32 (0.017)	0.226 (0.006)
SimCLR <sub>2</sub>	0.27 (0.011)	0.336 (0.011)	0.323 (0.007)	0.224 (0.019)

52 lentelė. Suformuotų scenarijų (taikant SimCLR bei augmentacijas) F1 medianų įverčiai kiekvienam duomenų poaibiui, II fazė.

Scenariius	Sleep-EDF-20	Sleep-EDF-60	Sleep-EDF-80	Sleep-EDF-100
<b>J</b>	Med. (MAD)	Med. (MAD)	Med. (MAD)	Med. (MAD)
ResNet <sub>1</sub>	0.832 (0.0003)	0.733 (0.003)	0.767 (0.004)	0.609 (0.02)
ResNet <sub>2</sub>	0.826 (0.004)	0.735 (0.001)	0.76 (0.004)	0.621 (0.009)
SimCLR <sub>1</sub>	0.828 (0.001)	0.749 (0.024)	0.763 (0.01)	0.653 (0.018)
SimCLR <sub>2</sub>	0.827 (0.005)	0.734 (0.01)	0.755 (0.009)	0.628 (0.009)

53 lentelė. Suformuotų scenarijų (taikant SimCLR bei augmentacijas) F1 medianų įverčiai kiekvienam duomenų poaibiui, III fazė.

Scenariius	Sleep-EDF-20	Sleep-EDF-60	Sleep-EDF-80	Sleep-EDF-100
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	Med. (MAD)	Med. (MAD)	Med. (MAD)	Med. (MAD)
ResNet <sub>1</sub>	0.841 (0.006)	0.565 (0.018)	0.618 (0.003)	0.662 (0.003)
ResNet <sub>2</sub>	0.82 (0.003)	0.553 (0.009)	0.623 (0.005)	0.654 (0.011)
SimCLR <sub>1</sub>	0.832 (0.003)	0.563 (0.028)	0.616 (0.006)	0.674 (0.013)
SimCLR <sub>2</sub>	0.845 (0.008)	0.556 (0.014)	0.62 (0.006)	0.664 (0.029)

Scenariius	Sleep-EDF-20	Sleep-EDF-60	Sleep-EDF-80	Sleep-EDF-100
~ · · · · · · · J ····	Med. (MAD)	Med. (MAD)	Med. (MAD)	Med. (MAD)
ResNet <sub>1</sub>	0.714 (0.009)	0.598 (0.022)	0.63 (0.013)	0.253 (0.025)
ResNet <sub>2</sub>	0.724 (0.01)	0.59 (0.019)	0.654 (0.011)	0.324 (0.045)
SimCLR <sub>1</sub>	0.747 (0.012)	0.656 (0.007)	0.685 (0.006)	0.279 (0.095)
SimCLR <sub>2</sub>	0.744 (0.014)	0.652 (0.011)	0.684 (0.01)	0.342 (0.035)

54 lentelė. Suformuotų scenarijų (taikant SimCLR bei augmentacijas) F1 medianų įverčiai kiekvienam duomenų poaibiui, REM fazė.

# D. Papildomi GradCAM aktyvacijų žemėlapiai



15 pav. Vidutiniai GradCAM aktyvacijų žemėlapiai visoms miego fazėms, kompleskinė Gauso vilnelė (Sleep–EDF–20 duomenys, teisingai suklasifikuoti atvejai). Viršutinėje eilutėje matomi SqueezeNet modelio aktyvacijų žemėlapiai, apatinėje – ResNet–18.



16 pav. Vidutiniai GradCAM aktyvacijų žemėlapiai visoms miego fazėms, Mexican Hat vilnelė (Sleep–EDF–20 duomenys, teisingai suklasifikuoti atvejai). Viršutinėje eilutėje matomi Squeeze-Net modelio aktyvacijų žemėlapiai, apatinėje – ResNet–18.



17 pav. Vidutiniai GradCAM aktyvacijų žemėlapiai teisingai suklasifikuotiems REM fazės pavyzdžiams (Sleep–EDF–60 duomenys) taikant kontrastinio mokymo/augmentacijų scenarijus.



18 pav. Vidutiniai GradCAM aktyvacijų žemėlapiai teisingai suklasifikuotiems REM fazės pavyzdžiams (Sleep–EDF–100 duomenys) taikant kontrastinio mokymo/augmentacijų scenarijus.