



**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**

---

**FAKULTA BIOMEDICÍNSKÉHO INŽENÝRSTVÍ**  
**Katedra biomedicínské techniky**

**Metody strojového učení pro detekci událostí  
v polysomnografických záznamech**

**Event detection in polysomnographic recordings by  
using machine learning techniques**

Diplomová práce

Studijní program: Biomedicínská a klinická technika

Studijní obor: Biomedicínský inženýr

Vedoucí práce: Ing. Václava Piorecká, Ph.D.

**Bc. Martin Bartoň**

---

**Kladno 2020**

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Bartoň** Jméno: **Martin** Osobní číslo: **456640**  
Fakulta: **Fakulta biomedicínského inženýrství**  
Garantující katedra: **Katedra biomedicínské techniky**  
Studijní program: **Biomedicínská a klinická technika**  
Studijní obor: **Biomedicínský inženýr**

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Metody strojového učení pro detekci událostí v polysomnografických záznamech**

Název diplomové práce anglicky:

**Event detection in polysomnographic recordings by using machine learning techniques**

Pokyny pro vypracování:

Prostřednictvím skriptovacího jazyka Python a knihoven pro strojové učení vytvořte automatický systém hodnocení polysomnografických záznamů. Z dostupné databáze na Oddělení spánkové medicíny v Národním Ústavu Duševního Zdraví (NUDZ) vytvořte sadu nejméně 300 anonymizovaných polysomnografických záznamů a rozdělte ji na trénovací a testovací množinu. Navrhněte a otestujte vhodný model hluboké neuronové sítě, pro automatickou detekci událostí v polysomnografických záznamech na základě sady trénovacích dat. Kvantitativně vyhodnoťte úspěšnost klasifikátoru na testovací sadě dat. Pro porovnání implementujte klasifikaci dat na základě metody nejbližšího souseda (k-NN). Vyhodnoťte výsledky vámi navržené metodiky a metody nejbližšího souseda. Model hluboké neuronové sítě implementujte do rutinního řetězce zpracování polysomnografických záznamů v NUDZ.

Seznam doporučené literatury:

- [1] LIANG, Sheng-Fu, Chin-En KUO, Yu-Han HU a Yu-Shian CHENG, A rule-based automatic sleep staging method, Journal of Neuroscience Methods, ročník 205, číslo 1, 2012, DOI: 10.1016/j.jneumeth.2011.12.022
- [2] Arnaud Sors, Stéphane Bonnet, Sébastien Mirek, Laurent Vercueil, Jean-François Payen, A convolutional neural network for sleep stage scoring from raw single-channel EEG, Biomedical Signal Processing and Control, ročník 42, číslo 1, 2018, DOI: 10.1016/j.bspc.2017.12.001
- [3] Lan Ma, James W. Minett, Thierry Blu, William S-Y. Wang, Resting State EEG-based biometrics for individual identification using convolutional neural networks, 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), ročník 1, číslo 1, 2015, DOI: 10.1109/EMBC.2015.7318985
- [4] EBRAHIMI, Farideh, Mohammad MIKAEILI, Edson ESTRADA a Homer NAZERAN, Automatic sleep stage classification based on EEG signals by using neural networks and wavelet packet coefficients, online, 2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, DOI: 10.1109/IEMBS.2008.4649365, 1558-4615

Jméno a příjmení vedoucí(ho) diplomové práce:

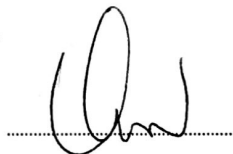
**Ing. Václava Piorecká, Ph.D.**

Jméno a příjmení konzultanta(ky) diplomové práce:

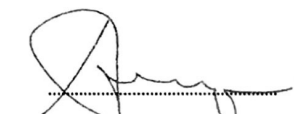
**Ing. Marek Piorecký, Ing. Vlastimil Koudelka, Ph.D.**

Datum zadání diplomové práce: **10.02.2020**

Platnost zadání diplomové práce: **19.09.2021**



prof. Ing. Peter Kneppo, DrSc., dr.h.c.  
podpis vedoucí(ho) katedry



prof. MUDr. Ivan Dylevský, DrSc.  
podpis děkana(ky)

## Prohlášení

Prohlašuji, že jsem diplomovou práci s názvem „Metody strojového učení pro detekci událostí v polysomnografických záznamech“ vypracoval samostatně a použil k tomu úplný výčet citací použitých pramenů, které uvádím v seznamu přiloženém k diplomové práci.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů.

V Kladně dne .....

.....

**Bc. Martin Bartoň**

## Poděkování

Rád bych poděkoval vedoucí mé práce Ing. Václavě Piorecké, Ph.D. za pomoc při vypracovávání této práce. Poděkování také patří konzultantovi Ing. Vlastimilovi Koudelkovi, Ph.D. za množství odborných konzultací v průběhu tvorby celé diplomové práce. Za poskytnutí výpočetního výkonu děkuji Ústavu fyziky plazmatu AV ČR a mému bratrovi Mgr. Petru Bartoňovi za jeho zprovoznění. Dále bych rád poděkoval celému BRAIN Teamu za celkovou podporu. V neposlední řadě bych rád poděkoval svým rodičům, za podporu v průběhu celého studia. Tato diplomová práce vznikla za podpory grantu SGS číslo SGS18/158/OHK4/2T/17 s názvem Topografické mapování v EEG.

## ABSTRAKT

### Metody strojového učení pro detekci událostí v polysomnografických záznamech:

Poruchy spánku se diagnostikují ve spánkové laboratoři pomocí polysomnografie, což je víceparametrové vyšetření, které monitoruje biologické signály v průběhu spánku. Následné hodnocení získaných záznamů je velice časově náročné. Cílem diplomové práce je vytvořit automatický systém hodnocení polysomnografických záznamů za pomoci strojového učení. Použitá data pro práci jsou ve formátu EASYS2 z Oddělení spánkové medicíny v Národním ústavu duševního zdraví (NUDZ). V rámci studie spánkových příznaků byla porovnána lineární (PCA) a nelineární (t-SNE) metoda redukce dimenze dat. Pro detekci spánkové apnoe a poklesu saturace krve kyslíkem byla navržena hluboká konvoluční neuronová síť, která využívá signál průtoku vzduchu dutinou nosní a signál saturace krve kyslíkem. Pro porovnání přesnosti neuronové sítě byla využita metoda  $k$  nejbližšího souseda (kNN). Detekce spánkové apnoe ze signálu průtoku vzduchu dutinou nosní dosahovala nejvyšší přesnosti na surových datech. Přesnost po aplikaci lineární redukce dimenze dat byla větší než po aplikaci nelineární redukce dimenze dat. Výsledná přesnost neuronové sítě při detekci apnoí dosáhla 84 % a při detekci desaturací 74 %. Výpočet pomocí kNN pro detekci apnoí dosáhl přesnosti 83 % a pro detekci desaturací 64 %. Výsledná hluboká konvoluční neuronová síť byla implementována do rutinního řetězce zpracování polysomnografických záznamů pomocí jednoduchého grafického rozhraní v NUDZ.

### Klíčová slova

strojové učení, konvoluční neuronová síť, spánková apnoe, polysomnografie

# ABSTRACT

## **Event detection in polysomnographic recordings by using machine learning techniques:**

The sleep disorders are diagnosed in sleep laboratory by polysomnography, which is multiparameter examination, that monitors biological signals during sleep. Subsequent evaluation of the obtained records is very time consuming. The goal of this thesis is to create automatic system for evaluation polysomnography records by usage of machine learning techniques. Data used in this thesis are in EASYS2 format from Division of Sleep Medicine in the National Institute of Mental Health (NIMH). As part of the study of sleep features the linear (PCA) and non-linear (t-SNE) method for data dimension reduction were compared. To detect sleep apnea and a decrease in blood oxygen saturation, a deep convolutional neural network was designed. This neural network uses the nasal airflow signal and the blood oxygen saturation signal. The  $k$  nearest neighbor method (kNN) was used to compare the accuracy of the neural network. A study of sleep features showed that the detection of sleep apnea from the nasal airflow signal achieved the highest accuracy on the raw dataset. The accuracy after applying the linear dimension reduction was greater than after applying the nonlinear dimension reduction. The final neural network accuracy for apnea detection reached 84 % and for desaturation detection 74 %. The kNN calculation for apnea detection reached an accuracy of 83 % and for desaturation detection 64 %. The final deep convolutional neural network was implemented into a routine chain of polysomnographic record processing using a simple graphical interface in NIMH.

## **Key words**

machine learning, convolutional neural network, sleep apnea, polysomnography

# Obsah

<b>Seznam symbolů a zkratk</b>	<b>8</b>
<b>1 Úvod</b>	<b>9</b>
1.1 Fyziologie spánku	9
1.2 Poruchy spánku	10
1.3 Syndrom spánkové apnoe	10
1.4 Polysomnografie	12
1.5 Strojové učení	14
1.6 Hluboké učení	15
<b>2 Přehled současného stavu</b>	<b>16</b>
<b>3 Cíle práce</b>	<b>17</b>
<b>4 Metody</b>	<b>18</b>
4.1 Programovací jazyk	18
4.1.1 Knihovny pro strojové učení	18
4.1.2 Záznam výsledku učení	19
4.2 Data	19
4.2.1 Souhlas etické komise	22
4.3 Studie signálových příznaků	22
4.3.1 Předzpracování dat	23
4.4 Hluboká neuronová síť	32
4.4.1 Vrstvy konvoluční neuronové sítě	33
4.4.2 Ostatní parametry	35
4.4.3 Návrh designu sítě	37
4.4.4 Hledání hyperparametrů	38
4.4.5 Trénování, výběr finálního modelu	41
4.5 Převod segmentů na značky	41
4.6 Porovnání s kNN	42
4.7 Detekce událostí	42
4.8 Implementace detektoru	42
4.9 Statistické metody	43
<b>5 Výsledky</b>	<b>45</b>
5.1 Studie spánkových příznaků	45
5.2 Design hluboké neuronové sítě	47
5.3 Výsledky detekce spánkových událostí	53
5.3.1 Detekce apnoí	53
5.3.2 Detekce desaturací	54

5.3.3	Detekce pomocí kNN . . . . .	54
<b>6</b>	<b>Diskuse . . . . .</b>	<b>56</b>
6.1	Studie signálových příznaků . . . . .	56
6.2	Návrh designu neuronové sítě . . . . .	57
6.3	Přesnost detektoru a porovnání s kNN . . . . .	58
<b>7</b>	<b>Závěr . . . . .</b>	<b>60</b>
	Seznam použité literatury . . . . .	61
	Příloha A: Seznam skriptů a funkcí . . . . .	66
	Příloha B: Obsah přiloženého CD/DVD . . . . .	67
	Příloha C: Článek publikovaný na EHB 2019 . . . . .	67



# Seznam symbolů a zkratek

## Seznam zkratek

Zkratka	Význam
CPU	Centrální procesorová jednotka (Central processing unit)
GPU	Grafický procesor (Graphicsprocessing unit)
PSG	Polysomnografie
EKG	Elektrokardiografie
EEG	Elektroencefalografie
EMG	Elektromyografie
EOG	Elektrookulografie
NUDZ	Národní ústav duševního zdraví
NN	Neuronová síť (Neural network)
CNN	Konvoluční neuronové síť (Convolutional neural network)
LSTM	Neuronová síť s dlouhou a krátkou pamětí (Long short-term memory)
RNN	Recurrent neural network (Rekurentní neuronové sítě)
SSA	Syndrom spánkového apnoe
AASM	Americká akademie spánkové medicíny (American academy of sleep medicine)
REM	Rychlý pohyb očí (Rapid eye movement)
NREM	Není přítomen rychlý pohyb očí (Non-rapid eye movement)
AHI	Apnoe-hypopnoe index
OSA	Obstrukční spánková apnoe
CSA	Centrální spánková apnoe
SSA	Smíšené spánková apnoe
PCA	Analýza hlavních komponent (Principal component analysis)
t-SNE	t-distributed stochastic neighbor embedding
kNN	k Nejbližší soused (k Nearest Neighbour)
2D CL	Dvourozměrná konvoluční vrstva (2D Convolutional layer)

# 1 Úvod

Spánek je důležitou součástí většiny živých organismů. Slouží k regeneraci těla i podpoře kognitivních funkcí. Pro dosažení plného účinku spánku je potřeba, aby byl spánek nepřerušovaný a dostatečně dlouhý. Existuje však velké množství spánkových poruch, díky kterým postižený postupně upadá do spánkové deprivace. Mezi jednu z nejrozšířenějších poruch spánku patří spánková apnoe, která se vyznačuje krátkými úseky bezdeší v průběhu spánku. Tato porucha není sama o sobě smrtelně nebezpečná (většinou je spojena s kardiovaskulárními problémy), ale zhoršuje kvalitu spánku. [1]

Poruchy spánku se vyšetřují v takzvané spánkové laboratoři pomocí polysomnografie, což je diagnostické vyšetření, které zaznamenává množství biologických signálů v průběhu spánku. Mezi základní zaznamenávané signály patří elektrogardiogram (EKG), elektroencefalogram (EEG), průtok vzduchu nosní dutinou a saturace krve kyslíkem ( $\text{SpO}_2$ ). Tento dlouhý záznam signálu z celé noci musí poté projít lékař, který ručně vytvoří hypnogram, označí úseky apnoe, poklesu  $\text{SpO}_2$  a dalších událostí. Tento proces je velice zdoluhavý a zabere několik hodin práce. Z důvodu velké časové náročnosti tohoto úkonu byl vytvořen požadavek na Národním ústavu duševního zdraví (NUDZ) na vytvoření specifického automatického detektoru spánkových událostí, který velice urychlí diagnostiku. Vzhledem k přítomnosti rozsáhlé databáze již diagnostikovaných a ohodnocených PSG záznamů se nabízí využít metody strojového učení, konkrétně hlubokých neuronových sítí, které využijí tyto ohodnocené záznamy a naučí se detekovat spánkové události.

## 1.1 Fyziologie spánku

Průměrný člověk prospí třetinu svého života. Kvalitní a dostatečně dlouhý spánek je nezbytný pro náš život stejně tak jako jídlo a voda. Délka potřebného spánku se liší s věkem. Kojenec prospí až 16 hodin, zatímco dospívajícímu mladému člověku stačí již 9 hodin. Ve stáří se může potřeba spánku dále zmenšovat. Dospělý člověk potřebuje průměrně 8 hodin spánku, avšak je to zcela individuální. Při nedostatečně dlouhém a kvalitním spánku upadáme do spánkové deprivace. Během této doby se zpomaluje reakční čas, zhoršuje se úsudek, jsme unavení a celkově se zhoršují naše kognitivní funkce. Spánkovou deprivaci můžeme napravit kvalitním spánkem. [2]

V průběhu spánku se snižuje tělesná teplota, tlak a zpomaluje se dýchání. Funkce spánku jsou rozsáhlé a zasahují celé tělo. Organismus se zotavuje a odvádí škodlivé metabolity vzniklé namáhavou aktivitou přes den, regenerují se tkáně a obnovují se energetické zdroje. Je prokázáno, že spánek je prospěšný pro dlouhodobou paměť.

I v dnešní době není zcela jasné, jak spánek přesně ovlivňuje mozek a naši mysl. [2, 3]

I přesto, že je naše tělo utlumeno, tak náš mozek je velice aktivní. V průběhu spánku se střídají různé fáze, během kterých jsou aktivované jiné části mozku. Dvě hlavní fáze spánku jsou REM (rapid eye movement, rychlé pohyby očí) a NREM (non-rapid eye movement, rychlé pohyby očí nejsou přítomny). Fáze NREM se dělí dále na 3 stádia, podle „hloubky“ spánku. Tyto dvě hlavní fáze se v průběhu noci pravidelně střídají. Během NREM fáze se zpomaluje mozková aktivita, snižuje se tělesná teplota, zastaví se pohyby očí, svalstvo je relaxované a většinou je těžké člověka v této fázi spánku probudit. Ve fázi REM se nám pohybují oči za zavřenými víčky, naše mozková aktivita se spíše podobá bdělosti, dýchání se stává rychlé a nepravidelné, tepová frekvence a krevní tlak dosahují též skoro hodnot jako při bdění. Během této fáze se nám zdá nejvíce snů, avšak ukazuje se, že je možné mít sny i v NREM fázi. Jednotlivé fáze jsou charakterizované jejich typickou elektrickou mozkovou aktivitou, která se liší frekvencí. Rozložení fází spánku a jejich typické frekvence jsou vidět v tabulce 1.1. [3]

Tabulka 1.1: Přibližné zastoupení fází při spánku a jejich typická aktivita [2, 4] .

	přibližná zastoupení (%)	typická mozková aktivita	frekvence (Hz)
bdělost	<1	mix vln	0,5-30,0
NREM I	4	alfa vlna	2,0-7,0
NREM II	50	k-komplex	0,5-14,0
NREM III	20	delta vlny	0,5-4,0
REM	25	mix vln	0,5-30

## 1.2 Poruchy spánku

Různé poruchy spánku jsou v populaci velice časté. Prevalence dosahují až desítek procent. Mezi rizikové faktory patří kardiovaskulární onemocnění, diabetes mellitus a obezita. Spánkové poruchy jsou rozděleny do sedmi kategorií podle mezinárodní klasifikace (ICSD) od Americké Akademie spánkové medicíny. Rozdělení do jednotlivých kategorií je vidět v tabulce 1.2. [5]

## 1.3 Syndrom spánkové apnoe

Syndrom spánkové apnoe (SSA) patří do druhé skupiny spánkových poruch dle členění ICSD-3. Jedná se o časté onemocnění. Jeho prevalence se pohybuje od 1%

Tabulka 1.2: Mezinárodní klasifikace spánkových poruch ICSD-3 [5]

1. Insomnie.
2. Poruchy dýchání související se spánkem (syndrom spánkové apnoe).
3. Hypersomnie centrálního původu (narkolepsie, idiopatická hypersomnie).
4. Poruchy cirkadiánního rytmu spánku (zpožděná fáze spánku, jet lag).
5. Parasomnie (noční můry, noční enuréza, spánkové halucinace).
6. Poruchy s pohyby ve spánku (syndrom neklidných nohou).
7. Jiné poruchy spánku.

do 6%. Je častější u jedinců, kteří trpí obezitou. Mezi rizikové faktory, které mohou pomoci rozvoji onemocnění patří: nadváha, konzumace nadměrného množství potravy před spaním, nepravidelný spánek, kouření a mužské pohlaví. Apnoe je většinou doprovázeno hlasitým chrápáním a narušením spánku, což vede k nekvalitnímu spánku. To může mít za následek unavenost po probuzení a následně zvýšenou spavost v průběhu dne. Většina jedinců si své nemoci není vědoma. Často to jedinec zjistí od svého partnera/partnerky, který/á je tím vyrušen ze spánku. [1]

Spánková apnoe je podle Americké akademie spánkové medicíny (AASM) definována jako bezdeší, které trvá minimálně 10 sekund a je doprovázeno poklesem saturace krve kyslíkem minimálně o 3%. Hypopnoe je definováno jako pokles celkové ventilace o 30% se stejným minimálním poklesem saturace krve kyslíkem minimálně na 10 sekund. Maximální délka apnoe není stanovena, avšak může se pohybovat až v řádu minut. Závažnost SSA se nehodnotí podle délky apnoe a hypopnoe, ale dle četnosti. Takzvaný apnoe-hypopnoe index (AHI), což znamená počet apnoí nebo hypopnoí v průběhu jedné hodiny. Třídění závažnosti SSA dle AHI můžeme vidět v tabulce 1.3. [6]

Tabulka 1.3: Třídění závažnosti dle AHI [6].

Hodnocení	Hodnota AHI
Fyziologická norma	AHI < 5
Lehká spánková apnoe	AHI < 15
Střední spánková apnoe	AHI 15 – 30
Těžká spánková apnoe	AHI > 30

Spánková apnoe se dělí na 3 základní druhy. Mají stejné projevy, avšak jejich příčina, patofyziologie a léčba je naprosto odlišná. Jedná se o obstrukční, centrální a smíšenou apnoe.

**Obstrukční spánková apnoe (OSA)** vzniká obstrukcí horních cest dýchacích. Jedná se tedy o mechanickou překážku v dýchacích cestách. Organismus se snaží tuto překážku překonat tím, že prodlužuje fázi nádechu, nikoliv výdechu. Tím pádem se

v dýchacích cestách udržuje podtlak. Tato mechanika vede k celkovému zhoršování dýchání. Na rozdíl od centrálního apnoe je zachované dechové úsilí, tuto skutečnost můžeme pozorovat na PSG záznamu, kde jsou vidět zachované pohyby hrudníku. Dále je typická elektromiografická (EMG) aktivita končetin po provedení prudkého nádechu na konci apnoe. Základem léčby je úprava životosprávy. Nejčastějším neinvazivním způsobem terapie je kontinuální přetlak v dýchacích cestách (CPAP), docílený maskou, která se nasadí na obličej a je napojena do přístroje, který vytváří přetlak. Chirurgická léčba se snaží vyřešit lokálně obstrukci. [7, 8]

**Centrální spánková apnoe (CSA)** má jinou patofyziologii než apnoe obstrukční. Jedná se o poruchu řízení respiračních mechanismů v centrální nervové soustavě. Dýchací cesty nejsou zablokované a i přesto je přítomna apnoická pauza. Během tohoto druhu apnoe nepřetrvává dechové úsilí. Na PSG záznamu nevidíme pohyby hrudníku. Mezi příčiny CSA patří: faryngeální kolaps, nadmořská výška, snížená citlivost k hyperkapnii, kmenové léze a další. K léčbě se využívají zařízení pro regulaci ventilace, případně ke stimulaci hrudních nervů. [7, 8]

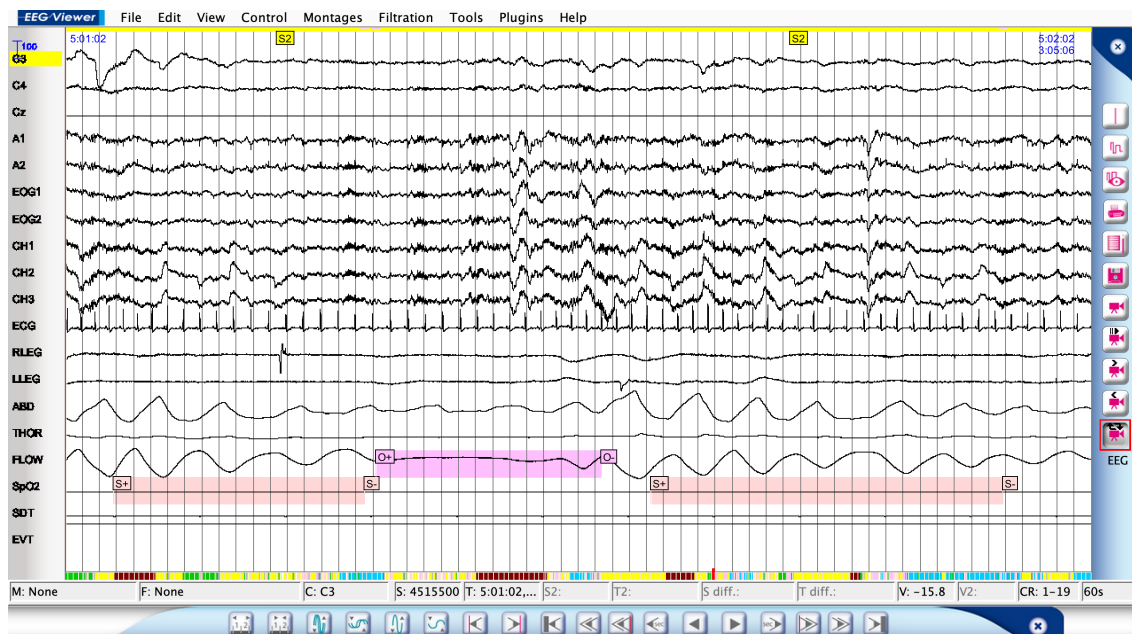
**Smíšená spánková apnoe (SSA)** je kombinací obou druhů apnoe. Nejprve dochází k CSA, díky které vymizí aktivita dilatátoru hrtanu. Ten kolabuje a vzniká tak obstrukce dýchacích cest. Tím pádem CSA přechází do OSA. Dechové úsilí je přítomno, až v druhé části apnoické pauzy při OSA. Léčba je podobná jako při OSA, CSA. [7, 8]

## 1.4 Polysomnografie

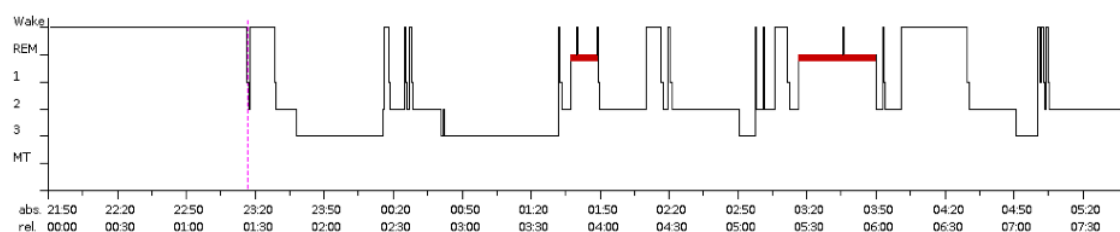
Polysomnografie je komplexní neinvazivní diagnostickou metodou, která se využívá ve spánkové medicíně. Umožňuje precizní detekci poruch dýchání a jejich přesnou klasifikaci, určení závažností apnoe, periodických pohybů končetin a dalších. Je také důležitá pro správné nastavení ventilační terapie. Výsledkem polysomnografického vyšetření je polysomnogram (PSG).

Samotné vyšetření probíhá většinou v takzvané spánkové laboratoři. Jedná se o speciálně upravenou místnost s lůžkem na odděleních spánkové medicíny. Pacient, kterému bylo toto vyšetření předepsáno, se dostaví ve večerních hodinách. Zdravotnický personál mu vysvětlí, jak bude vyšetření probíhat. Poté jsou na pacienta připevněny senzory pro záznam biologických signálů. Pacient přes noc spí a jsou mu nahrávány různé biologické signály. Celou dobu je sledován personálem z vedlejší místnosti pomocí kamer. Po probuzení je odpojen od přístroje, odchází a zaznamenaná data se předají lékařům k diagnostice. [7, 8]

Většina polysomnografických přístrojů nahrává základní standardní biologické signály, ale je možné tuto sadu rozšířit pro přesnější diagnostiku. Mezi standardní nahrávané biologické signály patří: elektroencefalogram, elektrokardiogram, pneu-



Obrázek 1.1: Ukázka PSG z programu EEG Viewer.



Obrázek 1.2: Ukázka hypnogramu.

mogram, elektrookulogram, elektromyogram, průtok vzduchu dutinou nosní, saturace krve kyslíkem ( $SpO_2$ ).

Dále se většinou pořizuje videozáznam pacienta a nahrává se zvuk. Část výsledného PSG záznamu můžete vidět na obrázku 1.1, kde je zobrazeno 60s záznamu z programu EEG Viewer. Lékař při hodnocení záznamu nejprve provede určení fází spánku. Pro to se využívá především EEG a EOG na bázi epoch (jedna epocha je 30 sec kontinuálního záznamu). Výsledek hodnocení fází spánku je hypnogram, což je zjednodušené zobrazení fází spánku v podobě grafu. Příklad hypnogramu je vidět na obrázku 1.2. Dále lékař prochází záznam a podle ostatních signálů označuje spánkové události (OSA, CSA, SSA, ...) Příklad vyznačených událostí je vidět na obrázku 1.1. Značky „O“ značí obstrukční spánkové apnoe a „S“ značí pokles saturace krve kyslíkem. V horní části je vidět značení spánkové fáze pomocí značky „S2“. Tento proces je velice zdoluhavý (dle závažnosti onemocnění a kvalitě signálu), lékařům jeho zhotovení, které je klíčové pro další postup v léčbě, trvá až několik hodin. [9]

## 1.5 Strojové učení

Strojové učení je vědecká disciplína, která zkoumá jak se mohou stroje „učit“. Jedná se o součást umělé inteligence. Algoritmus strojového učení se snaží vytvořit („naučit se“) matematický model podle trénovacích dat. Na základě tohoto modelu poté vykonává požadovanou úlohu, bez předem daných příkazů. Tato technologie se využívá v mnoha oblastech, například pro filtrování škodlivých zpráv, strojové vidění, autonomní řízení aut, zacílení komerce, předvídaní vývoje dat, detekce patofyziologických útvarů z rentgenových snímků, automatickou klasifikaci dat, rozpoznávání řeči a mnoho dalšího. [10]

Jako základní dělení algoritmů strojového učení se považuje rozdělení na: učení s učitelem, učení bez učitele a zpětnovazební učení. Mezi klasické úlohy pro strojové učení patří klasifikace, shlukování a regrese. Mezi základní metody strojového učení patří rozhodovací stromy, metody nejbližších sousedů, podpůrné vektory, neuronové sítě a hluboké učení. Pro názorné vysvětlení algoritmů využijí jednoduchý příklad s jablky, která jsou buďto zdravá či červivá. [10]

**Učení s učitelem** má za cíl předpovědět výsledek na základě známých dat. Vstupem pro vytvoření matematického modelu je dvojice dat. Samotná data neboli příznaky a požadovaný výstup (reálný stav). Například pokud budeme chtít třídit jablka podle toho jestli jsou červivá, tak příznaky mohou být: váha jablka, počet děr ve šlupce, vůně a jeho barva. K tomu jablko rozřízneme a tím zjistíme jeho opravdový stav. Pokud takto jablek budeme mít dostatečné množství, tak se postupně naučíme rozpoznávat červivá jablka i bez jejich nutnosti rozříznutí. Tento typ učení budeme v práci využívat pro detekci spánkových událostí z PSG záznamu. [11]

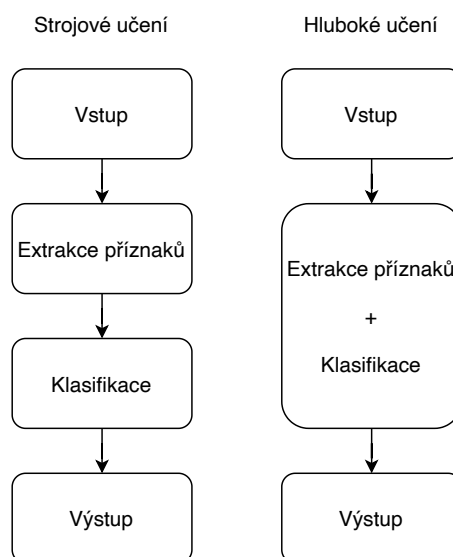
**Učení bez učitele** má za cíl rozdělení dat do podobných tříd na základě přirozeně se objevujících vzorů. Tím pádem v našem případě dokážeme jablka roztrždit do skupiny, která má hodně děr ve šlupce, jsou lehká, špatně voní a do skupiny jablek bez děr, těžkých a bez zápachu. Logicky nám přijde, že první skupina jablek bude červivá, ale pokud nikdy žádné jablko nerozřízneme, tak nikdy nemůžeme zjistit, která z našich dvou skupin je červivá a která ne. Ale máme je roztržďené a není problém ručně zjistit jaká hromádka jablek je červivá a jaká ne. Tento typ učení budeme v práci využívat pro studii signálových příznaků PSG signálů.[11]

**Zpětnovazební učení** funguje na jiném principu. Jeho hlavní výhoda je v tom, že nepotřebuje žádná předem změřená data, ale učí se za běhu. Využívá takzvaného vnějšího pozorovatele (agenta). Pokud se na to podíváme pomocí jablek, tak si představme, že každé jablko, které utrhneme sníme. Buďto bude dobré a nebo nám po něm bude špatně. Tím pádem vnější pozorovatel (my) si řekne, že příště to špatně vonící, lehké jablko jíst nebude. A tímto stylem se postupně učí. Velkou nevýhodou

tohoto přístupu je to, že vyžaduje dlouhou dobu (velké množství učících vzorků) na naučení oproti předešlým metodám. [11]

## 1.6 Hluboké učení

V předchozí kapitole jsme popsali zavedené postupy strojového učení. Nyní se soustředíme na moderní koncept strojového učení zvaný hluboké učení. Na diagramu 1.3 můžeme vidět základní strukturu konvenčního strojového učení a hlubokého učení. V obou případech začínáme se vstupními daty, která mohou být v různých formách. Při strojovém učení musíme „ručně“ extrahovat příznaky. Například určit, že budeme na jablkách pozorovat jejich váhu, barvu, vůni a počet červů. Poté přistoupíme ke samotné klasifikaci, což může být například jednoduchá neuronová síť, která nám vyhodnotí výsledek. Tento přístup je vhodný, pokud víme jaké příznaky v našich datech mají největší váhu a zároveň máme relativně málo dat. Přístup pomocí hlubokého učení na rozdíl od strojového učení nepotřebuje „ručně“ vybrat jednotlivé příznaky, protože samotná síť dokáže jednotlivé příznaky extrahovat a najít ty, které jsou nejlepší pro následnou klasifikaci. Toto dokáže díky tomu, že je struktura samotné sítě rozsáhlejší (hluboká) oproti normální neuronové síti. Přístup hlubokého učení je vhodné volit v případě, pokud nevíme přesně jaké příznaky zvolit a máme dostupnou velkou databázi dat, na kterých můžeme hlubokou neuronovou síť naučit jak extrahovat příznaky tak klasifikovat.



Obrázek 1.3: Diagram struktury konvenčního strojového učení a hlubokého učení.



## 2 Přehled současného stavu

V současné době se při diagnostice polysomnografických záznamů většina prací soustředí na spánkové apnoe, které patří mezi jednu z nejčastějších poruch spánku. Její prevalence se pohybuje od 1 % do 6 % [1]. Vzhledem k takto relativně vysoké prevalenci je poptávka po vytvoření úspěšného detektoru, který zajistí velkou časovou úsporu pro lékařský personál a rychlejší diagnostiku pro pacienty. Jelikož je spánkové apnoe komplexní spánková porucha, tak se projevuje na většině signálů přítomných v PSG. Mezi signály, kde je bezdeší možno určit bez pomoci detektoru, patří: průtok vzduchu nosní dutinou, měření obvodu hrudníku, břišní dutiny a saturace krve kyslíkem [6]. Avšak ukazuje se, že lze apnoe detekovat i ze signálů jako je EKG [12, 13, 14, 15, 16], EEG [17, 18, 19, 20] a dokonce i ze zvukového záznamu [21, 22]. Z toho vyplývá možnost kombinace různých signálů a rozdílných metod pro vytvoření vhodného detektoru.

Přirozeným vývojem byly nejprve implementovány detektory, které jsou založené na metodách bez strojového učení. Mezi tyto metody patří jednoduché prahování, expertní systémy, metoda nejbližšího souseda, spektrální analýza, diskriminační analýza a velké množství dalších. Dobrý přehled o článcích na tyto témata lze najít v přehledových článcích [23, 24]. Jelikož se ale velice rychle zvětšuje výkon výpočetní techniky běžně dostupné na odděleních spánkové medicíny, tak se začínají využívat metody strojového učení, které tento výkon potřebují a mají některé výhody oproti jednodušším detektorům.

S obecným rozvojem strojového učení a neuronových sítí začaly tyto metody pronikat i do medicíny. Klasické metody strojového učení jsou založeny na ručním určení jednotlivých příznaků. V práci Nadi Sadr [25] je vidět tento klasický postup. V prvním kroku je na EKG signálu vypočítáno 34 jednotlivých příznaků, které byli předem určeny. V druhém kroku se snaží zjistit, jaká kombinace jednotlivých příznaků je nejlepší pro samotnou detekci apnoe. Tyto metody mají však velkou nevýhodu v tom, že musíme příznaky určit ručně, což může být komplikované pro různé signály. [25]

Pokud se chceme vyhnout ručnímu určování příznaků, tak můžeme použít metodu hlubokých neuronových sítí. Tyto metody však potřebují velké množství trénovacích dat a velký výpočetní výkon pro natrénování, protože některé sítě mnohdy obsahují až desítky milionů parametrů. Existuje velké množství typů hlubokých neuronových sítí. Mezi dva nejpoužívanější typy patří rekurentní neuronové sítě s krátkou a dlouhou pamětí (LSTM-RNN) [26, 27, 28, 29] a konvoluční neuronové sítě (CNN). Ty se dají aplikovat na různé signály, například EKG [30, 15], průtok vzduchu v dutině nosní [31, 32], kombinace s pohyby hrudníku [33].

### 3 Cíle práce

Cílem této diplomové práce je vytvoření systému pro automatické hodnocení polysomnografických záznamů. Automatický systém bude detekovat spánkové apnoe a pokles saturace krve kyslíkem. Bude vytvořen ve skriptovacím prostředí Python pomocí strojového učení, konkrétně hluboké konvoluční neuronové sítě. Pro naučení sítě budou využity již ohodnocené PSG záznamy z archivu spánkového oddělení v NUDZ. Jednotlivé parametry neuronové sítě budou upraveny tak, aby byla přesnost detekce co nejvyšší. Pro ověření funkčnosti je cílem porovnání vytvořeného detektoru s metodou nejbližšího souseda. Závěrem práce je implementace vytvořeného modelu hlubkové neuronové sítě do rutinního řetězce zpracování PSG záznamů v NUDZ pomocí jednoduchého grafického rozhraní.

## 4 Metody

### 4.1 Programovací jazyk

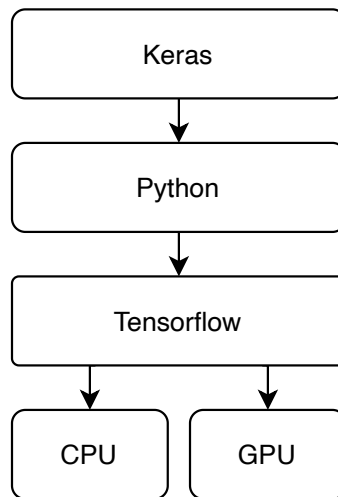
Pro vytvoření celé softwarové části práce byl využit vysokoúrovňový skriptovací programovací jazyk Python [34]. V dnešní době je využíván v mnoha odvětvích a je jedním z nejpoužívanějších programovacích jazyků dnešní doby. Veškeré dostupné verze jsou otevřené. To znamená, že je zcela zdarma pro jakékoliv využití a každý má možnost se podílet na jeho vývoji. Díky tomu lze spustit Python na jakémkoliv operačním systému. Má jednoduchou syntaxi oproti jiným programovacím jazykům, ale přesto je možné psát komplikované výrazy. Díky tomu se lze základy velice rychle naučit a Python intuitivně používat. Vzhledem k využití již existují (a stále vznikají) takzvané knihovny, které obsahují specifické funkce a rozšíření pro různá odvětví. Mezi nejpoblárnější patří *NumPy* (maticové operace) [35], *SciPy* (numerické operace) [35], *Matplotlib* (vykreskování grafů) [36], *OpenCV* (zpracování obrazu) [37], *Tensorflow* (neuronové sítě) [38], *Keras* (jednoduché ovládání Tensorflow) [39], *AppJar* (velice jednoduché grafické rozhraní, dostupné na [www.appjar.info](http://www.appjar.info)) a množství dalších. Často lze na jeden problém využít více různých knihoven. Ty se většinou liší svoji složitostí. Existují knihovny, které neobsahují tolik možností, ale jsou velice jednoduché a rychlé na použití, avšak na druhé straně může být knihovna, která je složitá, rozsáhlá a umožňuje velké množství funkcí.

Již všechny tyto výhody naznačují, proč byl Python verze 3.6 zvolen jako programovací jazyk pro tuto práci. Avšak dalším prvkem při rozhodování byla následná implementace v praxi. Na spánkovém oddělení v Národním ústavu duševního zdraví (NUDZ) se využívá program EEGviewer (datový formát EASYS2) pro diagnostiku spánkových událostí. Tento program je psán v Pythonu. Případná implementace vytvořeného software do prostředí EEGviewer by mohla být jednoduchá.

Veškeré vytvořené a použité funkce jsou popsány v seznamu skriptů a funkcí v tabulce 7.1 v příloze B.

#### 4.1.1 Knihovny pro strojové učení

Volně přístupnou otevřenou knihovnou pro numerickou matematiku, strojové učení, ale především neuronové sítě je knihovna *Tensorflow*. Byla vyvinuta týmem Google Brain pro společnost Google a uveřejněna v roce 2015. Protože je samotný *Tensorflow* poměrně složitý, pro praktický vývoj neuronových sítí byla vytvořena knihovna *Keras*, která ovládá *Tensorflow* a umožňuje jednoduchou a rychlou stavbu neuronových sítí pomocí vrstev. Jeden z nejpoužívanějšího způsobu tvorby neuronové sítě je takzvaný sekvenční model. Spočívá v postupném přidávání jednotlivých vrstev



Obrázek 4.1: Diagram závislosti knihoven pro tvorbu neuronových sítí.

sítě. Na obrázku 4.1 můžeme vidět graficky zobrazenou závislost jednotlivých prvků potřebných pro tvorbu neuronových sítí pomocí zmíněných knihoven a prostředí. [39]

#### 4.1.2 Záznam výsledku učení

Pro sledování průběhu strojového učení neuronových sítí v reálném čase byla využita služba Weights&Biases, dostupná na [www.wandb.com](http://www.wandb.com). Tento vývojový nástroj se velice jednoduše implementuje do zdrojového kódu k neuronové síti a poté umožní v reálném čase sledovat na webové stránce jak se daná síť učí, průběh veškerých parametrů a výsledků v přehledných grafech. Dále ukládá veškeré použité kódy, záznam z výstupu běžícího programu a další informace. Jeho velkou další výhodou je v tom, že umožňuje velice jednoduchou implementaci hledání nejlepších parametrů v neuronové síti. Po ručním zadání jednotlivých parametrů požadovaných pro hledání již automaticky spouští učení různých kombinací daných parametrů. Výsledky zpracuje a ukáže v přehledných grafech.

## 4.2 Data

Veškerá použitá data pro tuto práci pochází z Národního Ústavu Duševního Zdraví v Klecanech (NUDZ). Byla získána při běžných vyšetřeních ve spánkové laboratoři na polysomnografu v průběhu noci. Data mají přibližnou délku 8 hodin a byla ohodnocena lékaři z NUDZ podle standardů hodnocení spánkových událostí.

Jednotlivé polysomnogramy byly nahrány na polysomnografickém zařízení BrainScope se vzorkovací frekvencí 250 Hz. Tyto záznamy obsahují 19 kanálů EEG,

2 kanály EOG (horizontální a vertikální), kanál EMG z brady, průtok vzduchu dutinou nosní (dále jen průtok), 2 pohybové kanály umístěné na hrudníku, 2 kanály EMG umístěné na dolních končetinách a kanál saturace krve kyslíkem ( $\text{SpO}_2$ ). Dále obsahují označení jednotlivých spánkových událostí (apnoe, fáze spánku, pohyby končetin a další) pomocí značek, jejich příklady jsou uvedeny v tabulce 4.1.

Tabulka 4.1: Ukázka názvů a počtů značek z reálného polysomnogramu.

ID tagu	označení	počet	text
130	O+	225	Obstrukční apnoe začátek
131	O-	225	Obstrukční apnoe konec
132	A+	10	Centrální apnoe začátek
133	A-	10	Centrální apnoe konec
138	D+	159	Desaturace začátek
139	D-	159	Desaturace konec

Samotné záznamy jsou uloženy ve speciálním formátu s příponou „.d“. Tento formát je specifický pro přístroj BrainScope. Data byla načtena pomocí skriptu, který byl vytvořen v rámci předchozí bakalářské práce [40] s pomocí manuálu k datovému typu [41].

Dataset byl tvořen celem 800 anonymizovanými polysomnografickými záznamy. Z tohoto počtu však pouze 256 bylo vhodných pro automatické zpracování při využití kanálů saturace krve kyslíkem a proudu vzduchu dutinou nosní a 477 při využití pouze kanálu proudu vzduchu dutinou nosní. Nejčastější problém spočíval v přítomnosti lichých značek (událost v záznamu byla označena pouze začáteční či konečnou značkou). Nebylo možné tyto liché značky automaticky odstranit a proto byly celé záznamy vyřazeny z databáze.

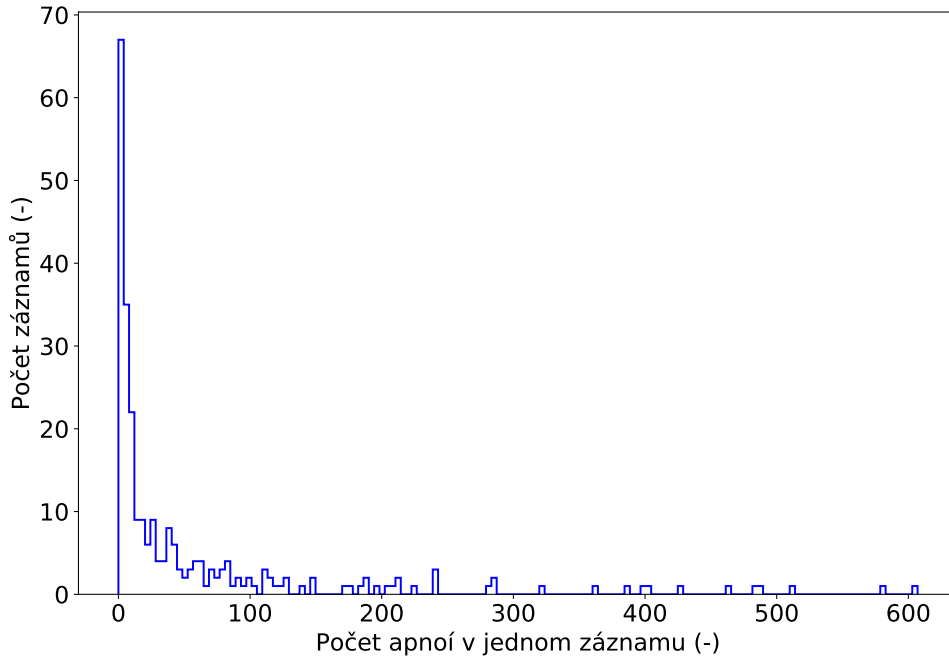
Protože se jedná o jednotlivá nezávislá měření, tak je počet přítomných apnoe v každém záznamu jiný. Obrázek 4.2 představuje histogram počtu apnoí v jednotlivých záznamech. Pokud by neuronová síť byla učena na veškerých datech, tak by malý počet jedinců s velkým počtem apnoí vyrovnal mnoho jedinců s malým počtem apnoe. Síť by v důsledku toho nebyla dostatečně generalizována. Z tohoto důvodu byl limitován maximální počet segmentů od jednoho subjektu. Tento maximální limit byl vypočten Tukeyho metodou [42]. Ta spočívá ve statistickém zpracování pomocí vztahu:

$$IQR = HK - DK \quad (4.1)$$

$$LB = DK - (IQR \cdot c) \quad (4.2)$$

$$UB = HK - (IQR \cdot c) \quad (4.3)$$

kde  $HK$  je horní kvartil,  $DK$  je dolní kvartil,  $IQR$  je mezikvartilové rozpětí,  $LB$  je spodní hranice intervalu,  $UB$  je horní hranice intervalu a  $c$  je konstanta určující přesah intervalu. Konstanta  $c$  je nastavena podle [42] na 1,5.



Obrázek 4.2: Histogram počtu apnoí v jednotlivých záznamech.

Při následné segmentaci těchto záznamů byl zjištěn velký nepoměr segmentů s apnoe a segmentů s normálním dýcháním. Pokud by měl pacient AHI rovné například 75, tak celková doba apnoe za 8 h činí přibližně 2,5 h. Z tohoto ukázkového případu je možné usoudit, že i kdyby veškerá dostupná data měla takto vysoký AHI index, tak by pouze 31 % segmentů bylo apnoických. Toto procento bylo však v dostupných datech této diplomové práce mnohem nižší. Při trénování konvolučních neuronových sítí je podle článku [43] lepší využít vybalancovaná data s poměrem 1:1 (pozitivní:negativní segmenty). Z tohoto důvodu byly následně veškeré datové sety upraveny tak, že z každého subjektu byly využity veškeré apnoické segmenty, dokud nepřekročil vypočtený limit. V případě překročení byl využit počet odpovídající vypočtenému limitu. K těmto apnoickým segmentům byl poté náhodně vybrán stejný počet segmentů s normálním dýcháním. Tímto systémem bylo docíleno vybalancovaného datového setu vhodného pro trénování konvolučních neuronových sítí.

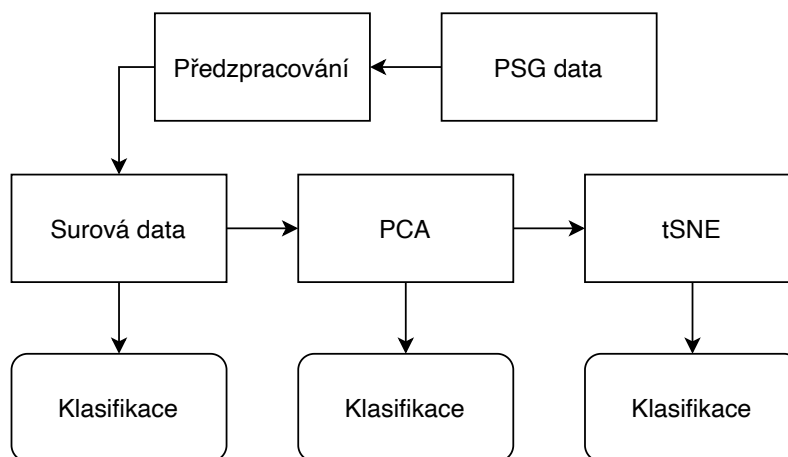
### 4.2.1 Souhlas etické komise

Polysomnografická data, použitá pro tuto práci, byla získána v rámci rutinního diferenciálně diagnostického postupu poskytování zdravotní péče o pacienty s poruchami spánku v rámci Oddělení poruch spánku, Národního ústavu duševního zdraví, Topolová 748, Klecany. Veškerá data, která byla poskytnuta pro tuto diplomovou práci (jednalo se pouze o polysomnografická data, audio ani video záznam nebyl pro diplomovou práci použit), byla poskytnuta anonymně pod kódovým označením a neobsahovala žádné identifikační údaje pacientů.

Pacienti před provedením video-polysomnografie podepsali informovaný souhlas, ve kterém byly informováni o možnosti anonymního využití těchto dat pro lékařský výzkum. Informované souhlasy pro provedení klinických video-polysomnografií a souhlas se zpracováním dat pro lékařské účely byly schváleny etickou komisí NUDZ, Topolová 748, 250 67 Klecany; e-mail: ek@nudz.cz.

## 4.3 Studie signálových příznaků

V rámci porozumění zpracovávaného signálu proudu vzduchu dutinou nosní v příznakovém prostoru byly porovnány lineární a nelineární metody redukce dimenze. Jako lineární metoda byla vybrána analýza hlavních komponent (PCA), pro nelineární metodu bylo použito t-SNE. Na diagramu z obrázku 4.3 je možné vidět postup zpracování dat a klasifikace.



Obrázek 4.3: Diagram postupu dat v procesu.

Pro výpočet bylo využito 477 záznamů. Surová data byla připravena podle postupu v kapitole 4.2. Signál byl podvzorkován na 50 Hz a segmentován na jednotlivé segmenty o délce 5 s. Protože signál obsahoval množství artefaktů (například při odpojení pacienta z PSG v průběhu noci), byly tyto segmenty odstraněny pomocí

Tukeyho metody [42]. Tímto způsobem bylo získáno 198530 segmentů, které byli rozděleny na testovací a trénovací sety v poměru 1:2. Rozdělení proběhlo na bázi celistvých subjektů (celý subjekt patří buďto do trénovacího nebo testovacího setu).

Cílem PCA je zmenšit dimensionalitu dat při co nejmenší ztrátě informace vyjádřením pomocí lineárních kombinací. Jedna z metod výpočtu PCA je analýza metodou kovariance. Tato metoda se skládá z 5 hlavních kroků: normalizace, výpočet kovarianční matice, výpočet vlastních vektorů a vlastních čísel kovarianční matice, výběr komponent a tvorba nových dat. [44, 45]

Normalizace zajistí správnou funkci celé metody. Pro normalizaci se využívá z-skore 4.4. Pomocí výpočtu kovarianční matice zjistíme, jak jsou na sobě veličiny lineárně závislé. Vypočtené vlastní vektory představují směr ve kterém mají data největší odlišnost. Vlastní číslo určují velikost této odlišnosti. Seřazením vlastních čísel od největšího po nejmenší získáme řadu, kde jsou komponenty řazeny podle významnosti. Protože chceme data redukovat, tak vybereme určitý počet komponent, které vysvětlují námi požadovanou část dat a ty zachováme. Pomocí vynásobení transponovaných vlastních vektorů a transponovaných původních dat získáme nová redukována data. [44, 45]

Oproti PCA je t-SNE nelineární metoda redukce dimenze dat. Algoritmus nejprve změří párové vzdálenosti mezi všemi body. Pro tyto body spočítá pravděpodobnost, přičemž blízkým bodům přiřadí vysokou hodnotu. Tato pravděpodobnost je počítána pomocí Gaussovy distribuce. Tímto výpočtem vznikne podobnostní matice. Dalším krokem je náhodné rozmístění bodů v nízkodimenziálním prostoru (typicky 2D). Následně se v malých krocích aktualizují polohy bodů, aby se minimalizovala Kullback-Leiblerova divergence mezi Gaussovou distribucí v původním prostoru a studentovou t-distribucí v novém nízkodimenziálním prostoru. Tato část výpočtu je iterativní a časově náročná. Skript pro výpočet t-SNE byl převzat z [https://github.com/jsilter/parametric\\_tsne](https://github.com/jsilter/parametric_tsne). [46]

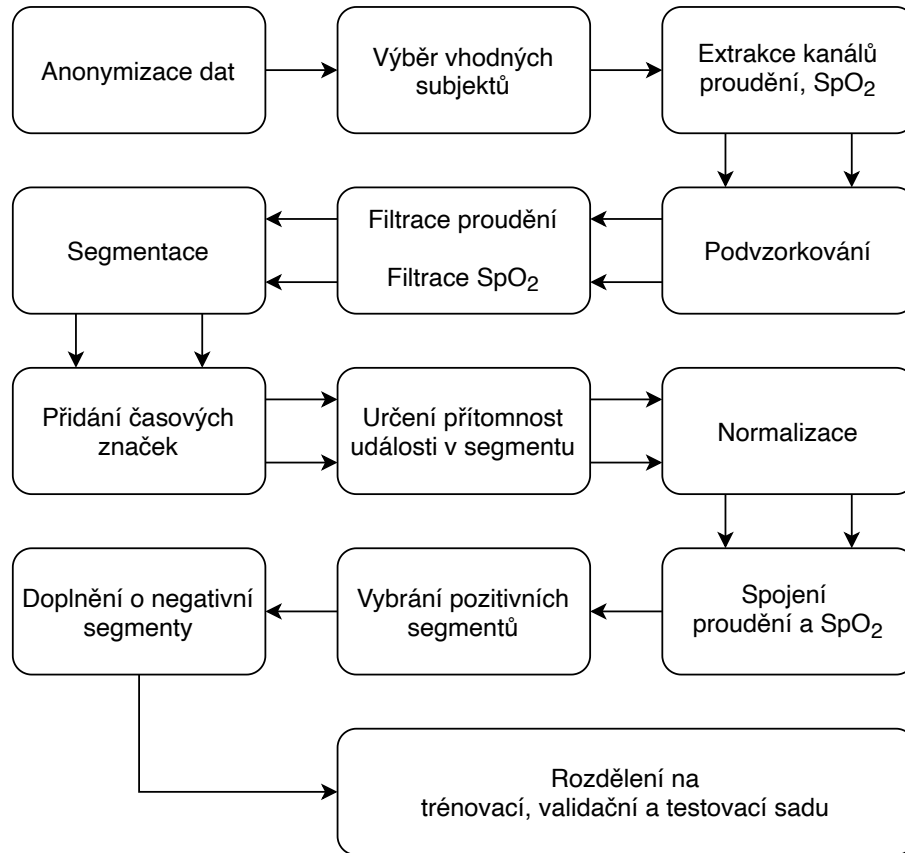
Jako systém pro klasifikaci byl využit algoritmus nejbližších sousedů neboli kNN. Protože bylo cílem porovnání metod a ne hledání nejlepšího výsledku byla konstanta  $k$  stanovena na 5. Takováto klasifikace byla aplikována na surová data, data po PCA a na data redukována pomocí t-SNE.

### 4.3.1 Předzpracování dat

Před samotným strojovým učením je potřeba vhodně předpřipravít data. Na diagramu z obrázku 4.4 je vidět celý proces přípravy. Protože se zpracovávají dva rozdílné kanály (průtok a SpO<sub>2</sub>), tak pomocí dvojité šipky jsou značeny kroky, kde kanály putují odděleně. Pokud je šipka jednoduchá, putují kanály dohromady. Toto zpracování je prováděno automaticky pomocí vytvořeného skriptu. V prvním



kroku byla veškerá data anonymizována pomocí funkce *anonymous*. Byly odstraněny veškeré informace, které by mohli vést k identifikaci subjektu.



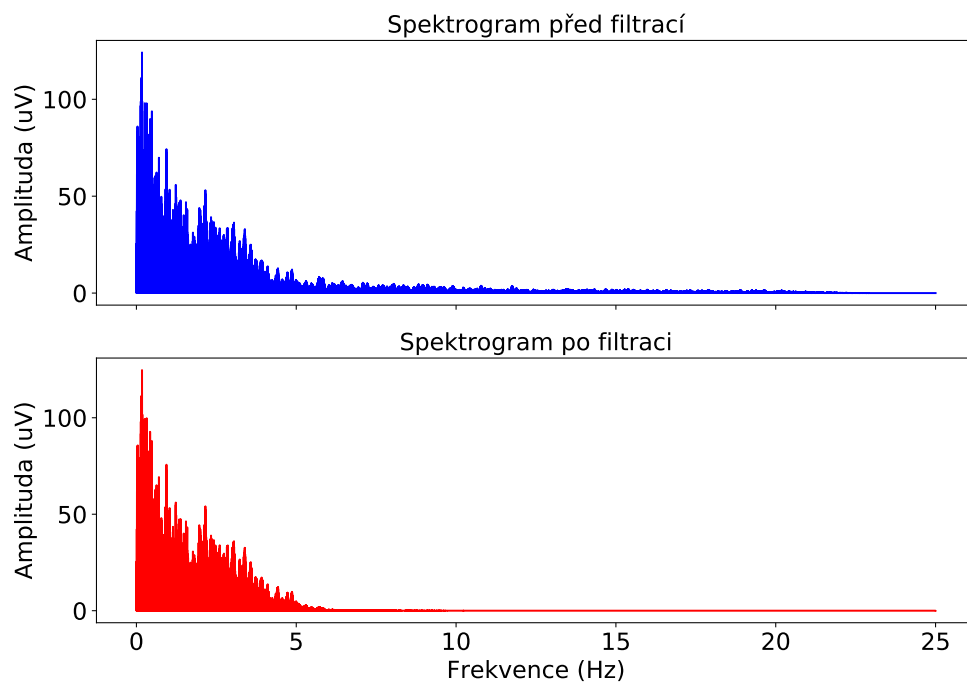
Obrázek 4.4: Diagram zobrazující postup při předzpracování dat.

Vybrání vhodných subjektů spočívalo v kontrole, jestli záznam obsahuje značky pro apnoe „O+“ nebo „A+“ (začátek obstrukčního nebo centrálního apnoe) a zároveň značky pro desaturaci „S+“ nebo „D+“ (od roku 2019 se změnila konvence značení). Po této kontrole zůstalo 256 vhodných záznamů. Tento krok je součástí skriptu *Bigdata\_multi.py*, který vybere vhodné záznamy a použije ostatní funkce pro předzpracování signálu.

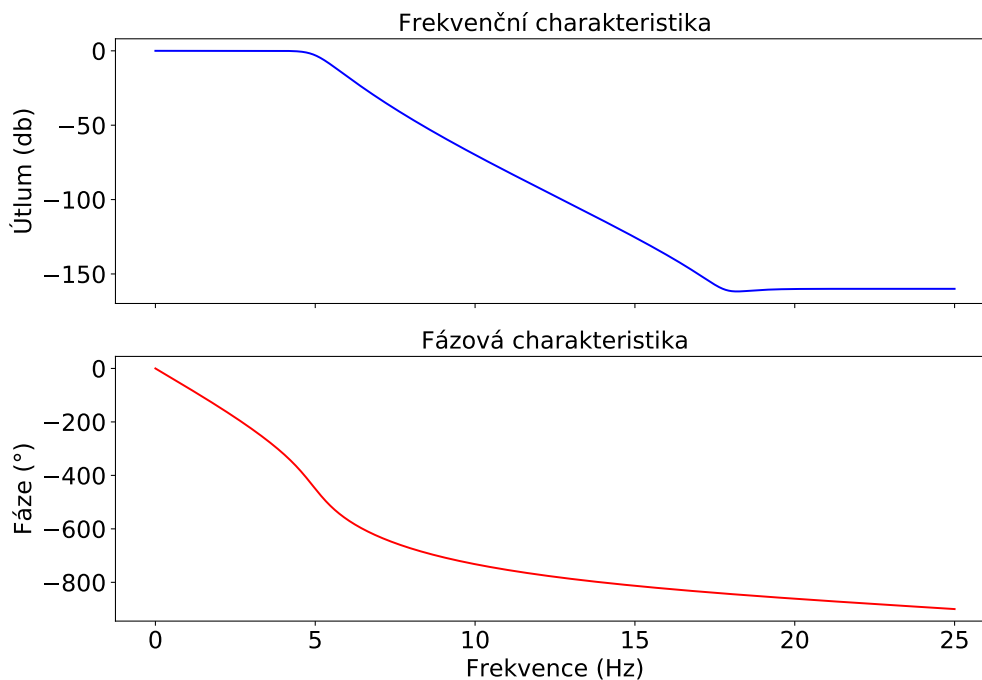
Vzorkovací frekvence všech kanálů přítomných v záznamech byla 250 Hz. Takto vysoká frekvence se využívá pro EEG. Vzhledem k tomu, že pro detekci apnoe a desaturace budeme využívat pouze signály průtoku a SpO<sub>2</sub> bylo možné signál podvzorkovat. Tvar signálu průtoku vzduchu je velice podobný sinusoidě o frekvenci přibližně 0,2 Hz (pro klidové dýchání). Saturace krve kyslíkem je ještě pomalejší signál. Tudíž byla zvolena cílová frekvence 50 Hz. To znamená, že byl signál 5x podvzorkovaný. Tento krok byl důležitý pro úsporu operační paměti a tím možný rychlejší výpočet při zachování stejné informace. Na podvzorkování byla použita funkce *decimate* z vědeckého balíčku *scipy v1.4.1*, která automaticky aplikuje anti-

aliasingový IIR filtr Chebysev typu 1, řádu 8. Pro zachování nulového posunu fáze byl filtr aplikován dvakrát (popředu a pozpátku).

Numerická filtrace průtoku zajistila jemnější vyhlazení a odstranění rychlejších frekvencí nad 5 Hz. Byl aplikován IIR filtr Butterworth 10 řádu. Na grafu z obrázku 4.5 jsou vidět vypočítaná frekvenční spektra jednoho záznamu pomocí rychlé fourierovy transformace před a po aplikaci filtru. Na grafu z obrázku 4.6 je zobrazena frekvenční a fázová charakteristika. Můžeme pozorovat, že fázový posun je od 0 Hz do 5 Hz téměř lineární a tím pádem neovlivní výsledek po filtraci.

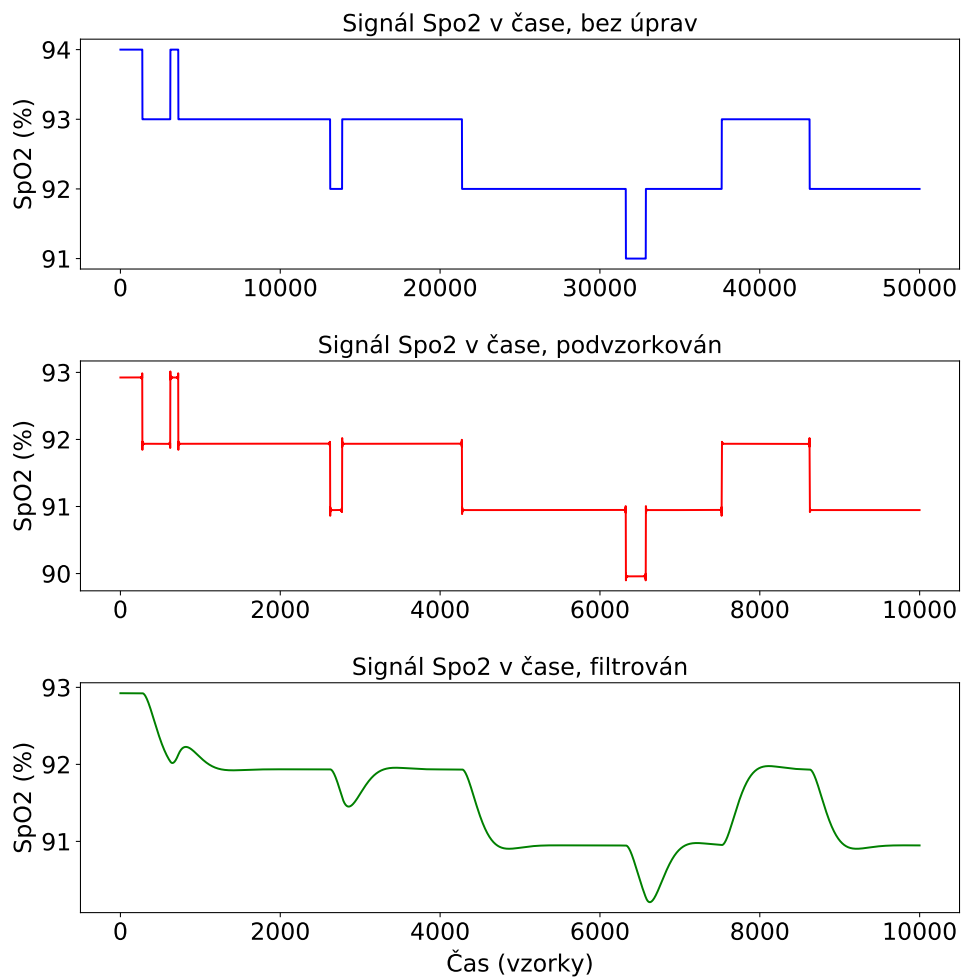


Obrázek 4.5: Ukázka spektrogramu před a po filtraci dolní propustí s mezní frekvencí 5 Hz.

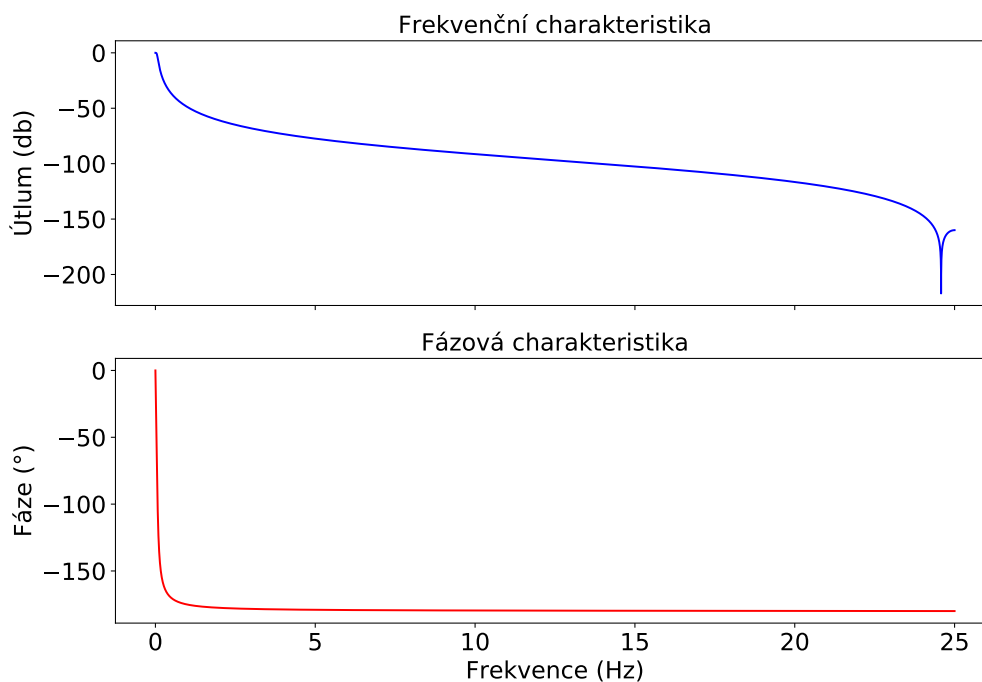


Obrázek 4.6: Frekvenční a fázová charakteristika filtru dolní propust s mezní frekvencí 5 Hz.

Pro vyhlazení signálu  $SpO_2$  byla taktéž použita numerická filtrace. Jak je vidět na grafu z obrázku 4.7, tak saturace krve kyslíkem byla originálně vzorkována s periodou 2 s avšak do celkové vzorkovací frekvence 250 Hz byly hodnoty dopočteny. Kvantizační krok o velikosti 1 % zajistil „schodovitý“ tvar signálu. Po podvzorkování na 50 Hz je vidět, že antialiasingový filtr vytvořil na ostrých přechodech záchvěvy. Pro odstranění záchvěvů a celkového vyhlazení signálu byl aplikován filtr dolní propust s mezní frekvencí 0,02 Hz (IIR Butterwort, 2 řádu). Jeho frekvenční a fázová charakteristika je vidět na grafu v obrázku 4.8.



Obrázek 4.7: Graf signálu SpO<sub>2</sub> bez úprav, po podvzorkování a po následné filtraci.

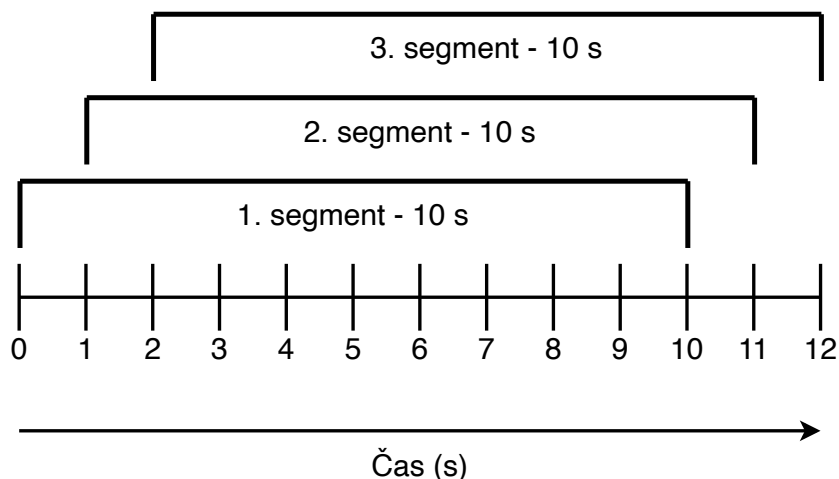


Obrázek 4.8: Frekvenční a fázová charakteristika filtru dolní propust s mezní frekvencí 0.02 Hz.

Segmentace signálu je provedena pomocí posuvného okna s překryvem. Na diagramu z obrázku 4.9 je vidět příklad segmentace na prvních 12 s signálu, které se rozdělí do 3 segmentů. Délka segmentu byla zvolena 10 s. Tato délka byla určena vzhledem k definici podle AASM, který říká, že spánkové apnoe má minimální délku právě 10 s. Velikost překryvu byla zvolena 90 %, tudíž 9 s. Takto velký překryv byl zvolen, protože cílem práce je lokalizovat danou událost s určitou přesností. Tyto zvolené parametry vychází ze studie [47], který porovnal segmenty dlouhé od 5 s do 10 s. Nejlepší výsledek byl při délce segmentu 10 s. I přesto, že studie využívala pouze signál jednoho kanálu (průtok vzduchu), dá se očekávat, že pro 2 kanály budou výsledky podobné. Podvzorkování, filtrace a segmentace jsou součástí funkce *preproces\_flow\_slide* a *preprocess\_spo2\_slide*.

Jednotlivé segmenty jsou uloženy v matici. Pro uchování informací ohledně umístění segmentu a značky události byla vytvořena matice, ve které každý řádek odpovídá jednomu segmentu a má 3 sloupce. První sloupec je zatím rezervovaný pro značení události. Do druhých dvou sloupců se zapsala absolutní lokace segmentu před podvzorkováním. Toto je důležité pro správnou rekonstrukci polohy událostí v původním signálu a možnosti určení obsažení událostí v dalším kroku.

V dalším kroku se určí, zda-li segment obsahuje určitou událost, či nikoliv. Nejprve se z původního souboru načtou požadované značky událostí s jejich umístěním.



Obrázek 4.9: Ukázka segmentace na prvních 12 s signálu.

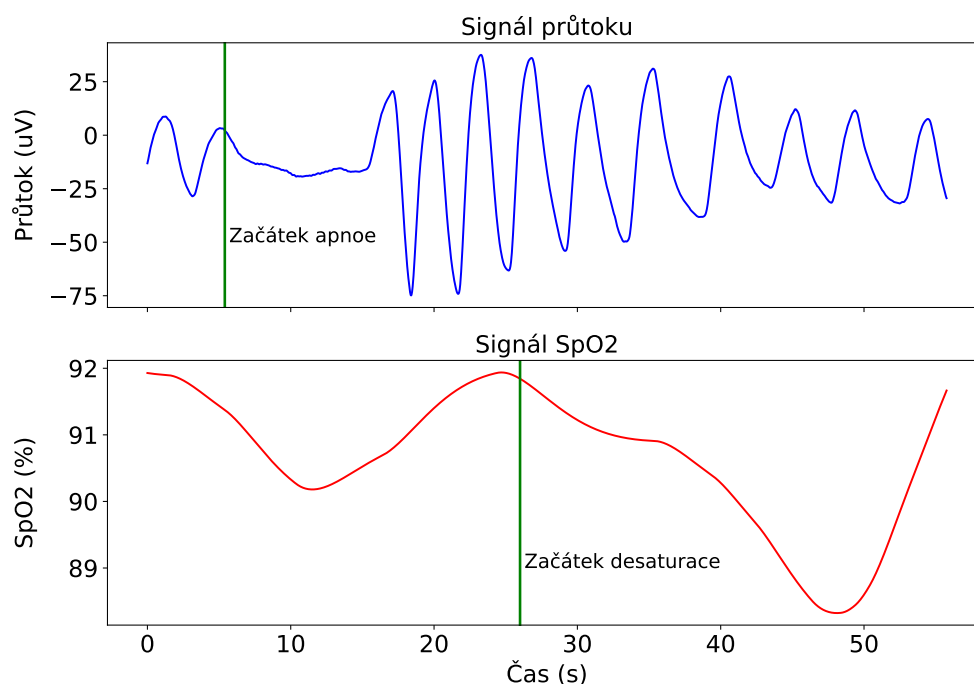
Například pro sledování apnoe se načtou značky „O+“ a „O-“ (začátek a konec). Jednotlivé segmenty jsou označeny jako pozitivní, pokud je minimálně 80 % jejich délky obsaženo v dané události. Pokud je segment označen jako pozitivní, je do pomocné matice do jeho odpovídajícího řádku zapsáno „1“. Pokud segment neobsahuje žádné události, je zapsána „0“. Určení segmentu je provedeno funkcí *istag\_slide*.

Vzhledem k tomu, že zpracováváme velké množství dat od různých pacientů, tak je žádoucí provést určitou normalizaci dat, protože každý subjekt má jinou základní amplitudu dýchání a saturaci krve kyslíkem. Zdálo by se, že pokud hledáme spánkovou apnoe podle poklesu amplitudy, tak je normalizace daného signálu kontraproduktivní. Ovšem pokud chceme zpracovávat mnoho různých záznamů, je to nezbytné vzhledem k učení neuronové sítě a zajištění jednotnosti dat. Normalizace byla vypočtená přes celý jeden záznam (pro zajištění celkové normalizaci napříč subjekty, ale zanechání rozdílností v signálu na úrovni subjektu). Pro výpočet normalizace bylo využito z-skore podle vztahu:

$$z = \frac{x - \mu}{\sigma} \quad (4.4)$$

kde  $x$  je normalizovaná hodnota,  $\mu$  je střední hodnota a  $\sigma$  je směrodatná odchylka. Normalizace je součástí funkce *preproces\_flow\_slide* a *preprocess\_spo2\_slide*.

Abychom mohli pracovat s kanály proudění a  $\text{SpO}_2$  dohromady, je potřeba tyto jednotlivé segmenty spojit do matice tvaru  $2 \times 500$  (2 řádky - proudění a  $\text{SpO}_2$ , 500 vzorků = 10 s). Na grafu z obrázku 4.10 je vidět ukázka 55 s obou signálů. Zelenou čarou je označen začátek apnoe a začátek desaturace. Jak je vidět, tak je začátek poklesu  $\text{SpO}_2$  opožděný o 20 s. Toto je fyziologická prodleva. Studie [48] zkoumala právě tento fenomén na pacientech trpících spánkovou apnoí. Její výsledek

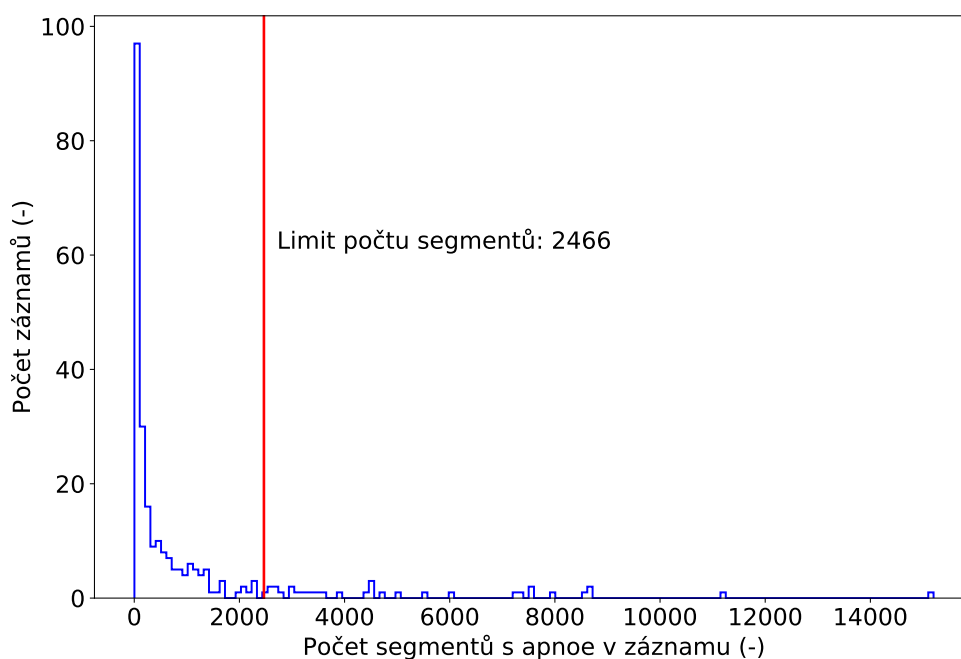


Obrázek 4.10: Ukázka reakce poklesu  $\text{SpO}_2$  při apnoe.

říká, že průměrně je tento rozdíl 25 s. Proto byl celý signál  $\text{SpO}_2$  posunut právě o 25 s napřed, aby se vyrovnalo časové zpoždění a dalo se s segmenty pracovat zvlášť.

Datové sety pro detekci desaturací a apnoí jsou tvořeny ze stejných záznamů, ale jinak se dívají na pozitivní a negativní segmenty (segment ve smyslu  $2 \times 500$  - již spojené proudění a  $\text{SpO}_2$ ). Aby se segment počítal jako pozitivní, stačí, aby v segmentu byl označen jako pozitivní sledovaný znak. Například, segment pro detekci apnoe je pozitivní, pokud obsahuje apnoe. Na stavu  $\text{SpO}_2$  nezáleží. Naopak to je pro segmenty, na kterých se detekuje desaturace.

Počet segmentů, které obsahují danou událost, se liší u každého záznamu. Protože některé záznamy mají mnohonásobně více těchto segmentů, mohla by se následně neuronová síť naučit pouze na tomto záznamu, proto byl vypočten maximální limit počtu segmentů z jednoho záznamu. Tento problém a výpočet limitu byl podrobně popsán v sekci 4.2. Histogram, který ukazuje četnost záznamů v závislosti na počtu segmentů s událostí a vypočteným limitem je pro apnoe zobrazen na obrázku 4.11 a pro desaturace na obrázku 4.12. Pokud záznam obsahoval menší počet pozitivních segmentů, než byl vypočtený limit, tak byly všechny pozitivní segmenty přidány do finální matice dat. Pokud záznam obsahoval pozitivních segmentů více, tak byl přidán pouze vypočtený limitní počet. V tomto případě byly pozitivní segmenty vybrány náhodně. Následovně byl z každého záznamu přidán stejný počet segmentů negativních (bez přítomnosti události). Tyto segmenty byly vybrány náhodně.

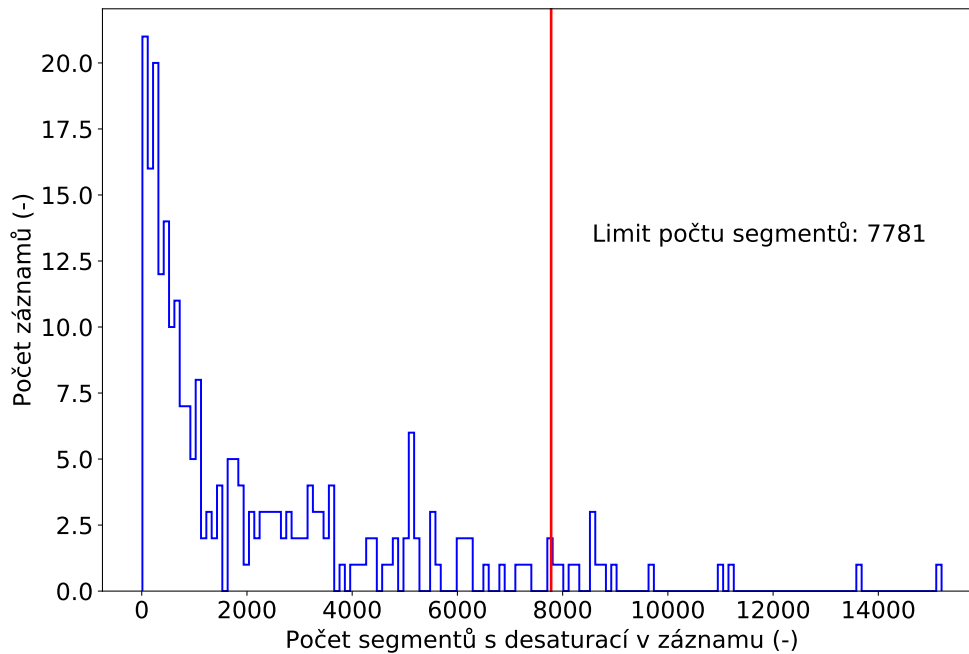


Obrázek 4.11: Histogram četnosti záznamů v závislosti na počtu pozitivních segmentů v záznamu při detekci apnoe s vypočteným limitem.

Výpočet maximálního limitu počtu událostí a jejich zařazení do matice je součástí skriptu *Bigdata\_multi.py*.

Posledním krokem v předzpracování je rozdělení dat do trénovací, testovací a validační skupiny. Toto rozdělení je provedeno pomocí funkce *TVT\_split*. Neuronová síť se poté učí pomocí sady trénovací. Sada validační slouží k průběžnému ladění parametrů. Pomocí testovací sady se na závěr vyhodnotí funkčnost sítě. Data byla rozdělena v následovném poměru: 2/3 trénovací sada, 1/6 testovací sada a 1/6 validační sada. Data byla rozdělena na úrovni celých záznamů, protože je nežádoucí, aby se algoritmus trénoval na podobných událostech, na kterých by se poté testoval. Vzhledem k tomuto opatření není finální poměr počtu segmentů přesný podle poměru, ale přibližný. Z důvodu nevyváženosti počtu segmentů v subjektech byl poměr sledován na úrovni subjektů. Finální velikost datových sad pro detekci apnoe je vidět v tabulce 4.2 a pro detekci desaturace v tabulce 4.3. Počet segmentů pro desaturaci je větší, protože samotný pokles  $\text{SpO}_2$  nezpůsobuje pouze apnoe, ale i hypopnoe (které není zahrnuto v apnoe).





Obrázek 4.12: Histogram četnosti záznamů v závislosti na počtu pozitivních segmentů v záznamu při detekci desaturace s vypočteným limitem.

Tabulka 4.2: Počty segmentů a subjektů v sadách pro detekci apnoe.

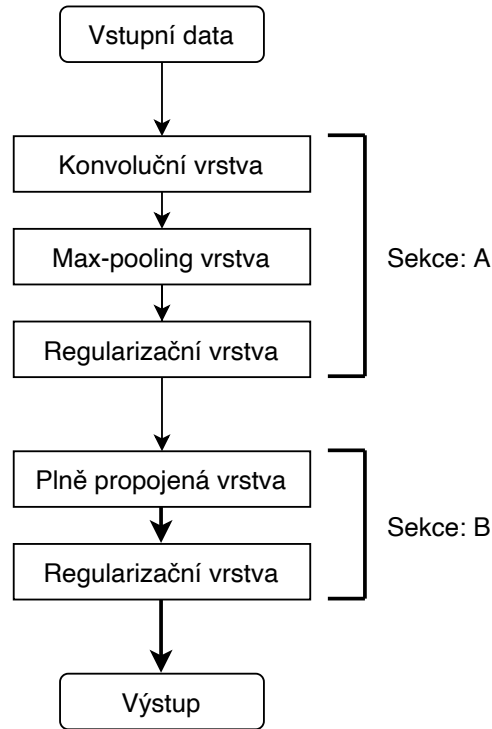
	Počet segmentů	Počet subjektů
Trénovací sada	351550	175
Validační sada	87180	38
Testovací sada	88882	42

#### 4.4 Hluboká neuronová síť

Metoda hlubokých neuronových sítí pro klasifikaci dat je velice populární. Existuje dále mnoho variant dané neuronové sítě. V dnešní době se často používá takzvaná konvoluční neuronová síť. Tato síť totiž umí velice dobře rozpoznávat jednotlivé prvky v signálu (obrazu) a proto se využívá hlavně ke strojovému vidění. Pro časové řady se využívá typ LSTM. Signál průtoku v čase by byl ideální právě pro použití LSTM neuronové sítě. Avšak vzhledem k tomu, že trénovací data na sobě nejsou časově navázaná, viz 4.2, byl zvolen typ konvoluční neuronové sítě, která má svá specifika a výhody.

Tabulka 4.3: Počty segmentů a subjektů v sadách pro detekci desaturace.

	Počet segmentů	Počet subjektů
Trénovací sada	777640	174
Validační sada	193886	36
Testovací sada	198082	45



Obrázek 4.13: Diagram zobrazující základní uspořádání vrstev pro konvoluční neuronovou síť.

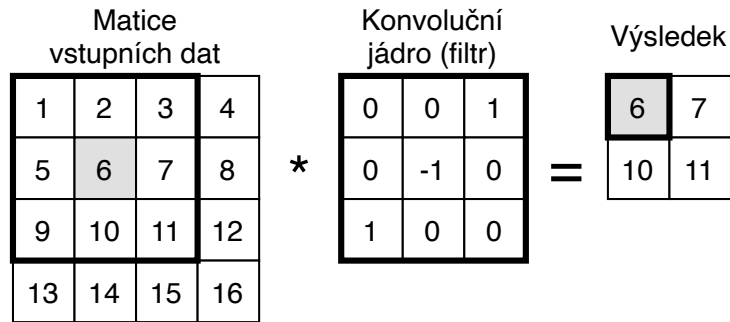
#### 4.4.1 Vrstvy konvoluční neuronové sítě

Návrh neuronové sítě pracuje s konceptem vrstev. Jednotlivé vrstvy se tak postupně přidávají a tím se tvoří celá síť. Na diagramu z obrázku 4.13 je zobrazena základní struktura typické konvoluční neuronové sítě. Jednotlivé vrstvy jsou popsány níže. [49]

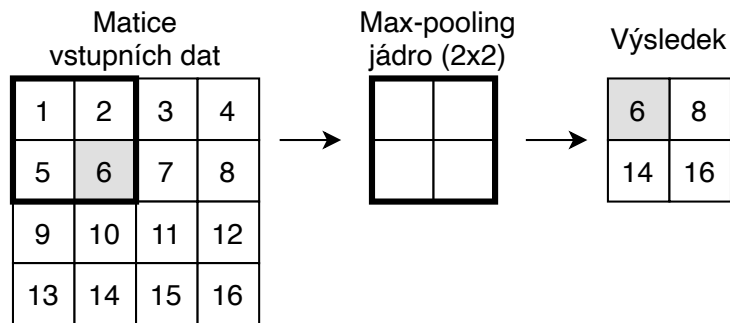
**Konvoluční vrstva** je jedna z hlavních vrstev. Jak již název napovídá, tato vrstva aplikuje matematickou operaci konvoluce na vstupní signál. Konvoluce je definována vztahem:

$$(a \cdot b)_k = \sum_{i=1}^n a_i b_{k-i} \quad (4.5)$$

kde  $a$  a  $b$  představují vektory, jejichž skalární součin je vypočten v čase  $k$ ,  $i$  značí pozici ve vektoru. [44]



Obrázek 4.14: Ukázka 2D konvoluce.

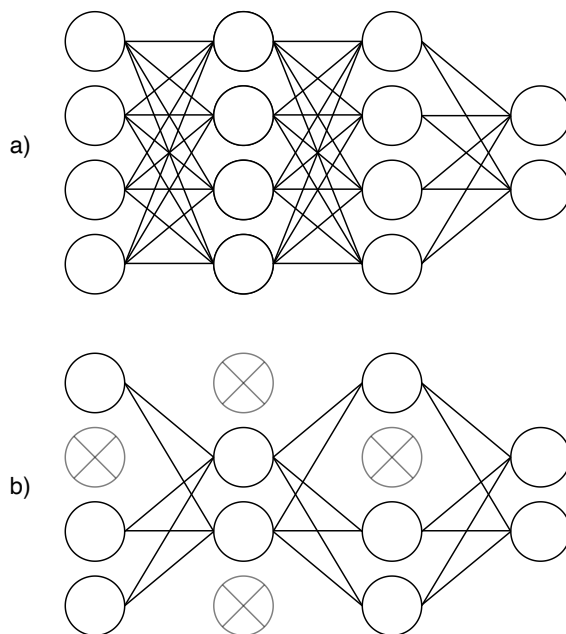


Obrázek 4.15: Ukázka max-poolingu.

Na obrázku 4.14 je ukázána jednoduchá konvoluce na matici 4x4 s konvolučním jádrem 3x3. Je vidět, že výsledná matice je menší než vstupní. Toto je důsledek velikosti jádra a nepoužití žádné techniky doplnění. Z této ukázky vyplývají parametry, které lze v konvoluční vrstvě nastavit. Rozměry konvolučního jádra neboli filtru jsou důležité, volí se podle předpokládané velikosti hledaných příznaků. Další parametr je posuv. Tento parametr volí o jakou vzdálenost se daný filtr na vstupu posune (v ukázce 4.14 je tento posuv 1). Posledním hlavním parametrem v konvoluční vrstvě je počet jednotlivých filtrů. Tento počet teoreticky odpovídá počtu hledaných příznaků.

**Max-pooling** vrstva většinou následuje po každé konvoluční vrstvě. Její význam spočívá v redukci velikosti daného prostoru. Toto zmenšení zaručí, že další konvoluční vrstva bude mít jiná vstupní data a bude hledat obecnější příznaky. Základní princip je vidět na obrázku 4.15. Vstupní data projde maska max-poolingu a vybere do nové matice největší hodnotu. Obecně se tento proces nazývá pooling (vytažení), existuje více možností pro výpočet výstupní hodnoty z masky. Kromě vybrání největší hodnoty se často používá průměrování. [49]

**Regularizační vrstva** se nachází před další konvoluční, či plně propojenou vrstvou. Funkce této vrstvy je „zobecnování“. To znamená, že různými metodami zabraňuje přeučení sítě na trénovací sadě dat. Pokud se totiž síť naučí správně



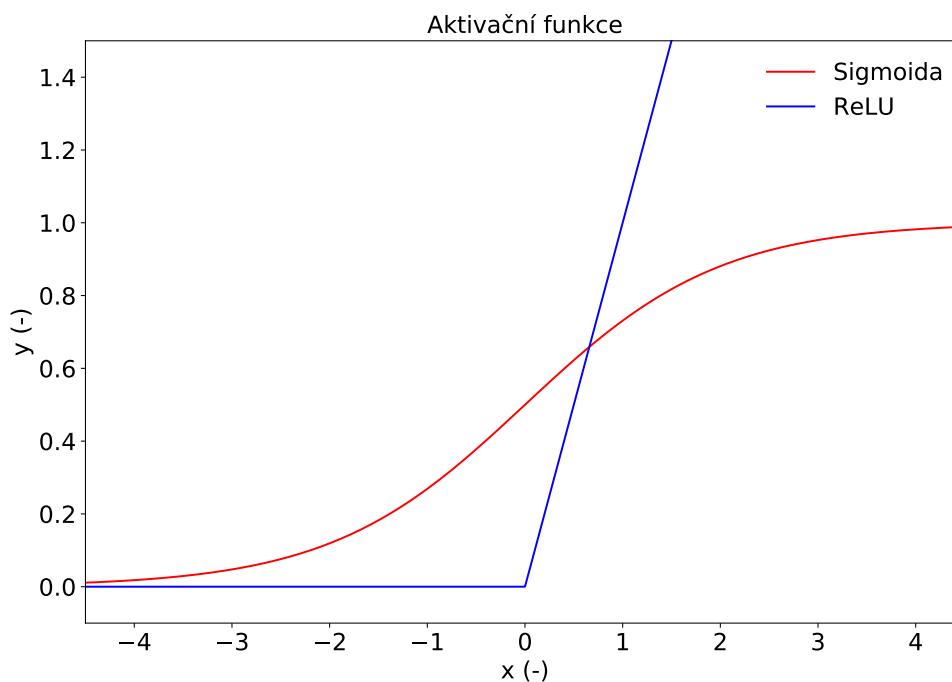
Obrázek 4.16: Ukázka funkce dropout vrstvy, **a)** obsahuje plně propojenou NN, **b)** představuje síť s aplikací dropout vrstvy.

rozhodovat pouze na trénovací sadě, ještě to neznamena, že bude správně rozhodovat i na datech, které nikdy neviděla (validační a testovací sada). Základní metodou regularizace je takzvaný dropout. Tato metoda spočívá v tom, že v průběhu trénování se náhodně vypínají vybrané neurony a jejich spojení. Příklad na klasické NN je vidět na obrázku 4.16. [50] V článku [50] parametr dropout určuje s jakou pravděpodobností bude neuron udržen. Například hodnota 1 značí, že neuron nebude nikdy vyřazen. V knihovně Tensorflow verze 2.0 však tento parametr značí pravděpodobnost s jakou bude neuron vyřazen. Hodnota 0 tedy znamená, že neuron nebude nikdy vyřazen.

**Plně propojená vrstva** (dense layer) následuje většinou jako poslední. Její úkol je vzít všechny vytvořené příznakové mapy z konvolučních vrstev (převede je do jednorozměrného vektoru pomocí funkce zploštění) a naučit se vybrat jejich správnou kombinaci, která nejlépe odpovídá danému segmentu. Většinou se používají minimálně 2 vrstvy, z čehož poslední vrstva má pouze tolik neuronů, jaký tvar má výstup (v našem případě 1).

#### 4.4.2 Ostatní parametry

**Aktivační funkce** (activation function) zajišťuje kdy a jak bude daný neuron aktivován. V naší neuronové síti byla využita aktivační funkce ReLU (rectified linear unit) [51] a funkce sigmoida. Tyto funkce jsou vykresleny na grafu z obrázku 4.17. ReLU je použita ve všech vrstvách, kromě vrstvy poslední. Je vhodná pro



Obrázek 4.17: Aktivační funkce ReLU a sigmoida.

skryté vrstvy, avšak nevhodná pro poslední vrstvu klasifikační. V této poslední vrstvě se nachází funkce sigmoida, která nám zajistí, že výsledek bude v rozmezí 0-1. To prakticky znamená pravděpodobnost daného jevu. Existuje velké množství aktivačních funkcí. K dnešnímu dni je však tato kombinace v oblasti konvolučních sítí nejpoužívanější.

**Účelová funkce** (loss function) slouží jako parametr, který se neuronová síť snaží co nejvíce zmenšit a tím zajistit nejlepší výsledky. Pro naši síť byla využita binární křížová (cross) entropie, která je nejvyužívanější pro případy binární klasifikace s využitím sigmoidy. [52]

**Optimalizér** (optimizer) je metoda, která má za úkol minimalizovat účelovou funkci. V dnešní době existuje mnoho metod. Mezi nejpoužívanější patří algoritmus Adam, který se snaží vypočítat aktuální gradient z minulých a pracuje se statistickými momenty. Síla Adamu spočívá v tom, že využívá různou rychlost učení pro jednotlivé váhy. [53]

**Dávka** (batch size) je parametr, který udává, kolik vzorků projde paralelně sítí během jednoho průchodu. Například pokud máme 200 vzorků, tak při zvolené dávce 50 vzorků se bude skládat 1 epocha ze 4 průchodů. Průchodem se myslí zpětné upravení vah sítě. Velikost této dávky ovlivňuje rychlost učení sítě. Dá se říci, že čím větší dávku zvolíme, tím rychleji se síť dokáže učit. Na druhou stranu maximální

Tabulka 4.4: Přehled zkratk parametrů, použitých při popisování neuronové sítě.

Zkratka	Význam
K	Rozměr konvolučního jádra
F	Počet filtrů
M	Rozměr maxpool jádra
D	Velikost dropout hodnoty
N	Počet neuronů

velikost této dávky je limitována technickým vybavením. Zároveň, čím větší dávku použijeme, tím větší nepřesnost do učení sítě vnese a tím pádem nám klesne přesnost na trénovací sadě dat.[39]

### 4.4.3 Návrh designu sítě

Je více možností, jak jednotlivé události z jednotlivých kanálů detekovat. Byla využita varianta, při které se apnoe a desaturace detekují zvlášť. Avšak pro detekci využívají jak signál SpO<sub>2</sub> tak průtok. Proto byla využita 2D konvoluce a poslední vrstva obsahuje pouze jeden výstupní neuron. Nalezená ideální síť i její parametry byla využita pro obě sítě detekující jak apnoe tak desaturaci.

Na diagramu z obrázku 4.13 je vidět rozdělení vrstev do 2 sekcí. Tyto celé sekce se dají za sebe skládat do hloubky, čímž se zvětšuje složitost neuronové sítě. Kolik vrstev má daná síť obsahovat se liší podle jejího účelu (složitosti příznaků). Není pravidlo, že čím více tím lépe. Existuje optimum, které je možné v našem případě ručně vyhledat. Ukazuje se však, že pro složitější síť není vhodné pouze síť zvětšovat. Je výhodnější některé vrstvy přeskokovat. Takováto síť se jmenuje ResNet [54]. Ze studií [31, 32], které se zabývají podobnou problematikou jsem zjistil, že použití sítí ResNet je zbytečně komplikované stačí přibližně 6 vrstvá konvoluční neuronová síť.

Při popisování designů sítí byla použita tabulka s vrstvami a jejich parametry. Pro přehlednost byly využity zkratky, jejich seznam a význam je vidět v tabulce 4.4.

První neuronová síť s odhadnutým designem a parametry byla použita pro porovnání chování sítě při různém počtu trénovacích dat. Design a parametry sítě jsou vidět v tabulce 4.5. Na grafech z obrázku 4.18 je vidět průběh přesnosti a chyby detektoru v průběhu trénování na 50 epochách při různé velikosti datové sady. Jak můžeme vidět, tak validační chyba a přesnost se zvětšuje od přesnosti a chyby trénovacích, což nám značí přeučení sítě. Dále můžeme pozorovat, že čím méně dat použijeme pro trénování, tím rychleji se dokáže síť naučit na trénovací sadě. Na druhou stranu však s menším počtem dat síť hůře generalizuje (menší validační

Tabulka 4.5: Struktura a parametry neuronové sítě použité pro ověření vlivu počtu dat na průběh trénování. Značení parametrů je popsáno v tabulce 4.4

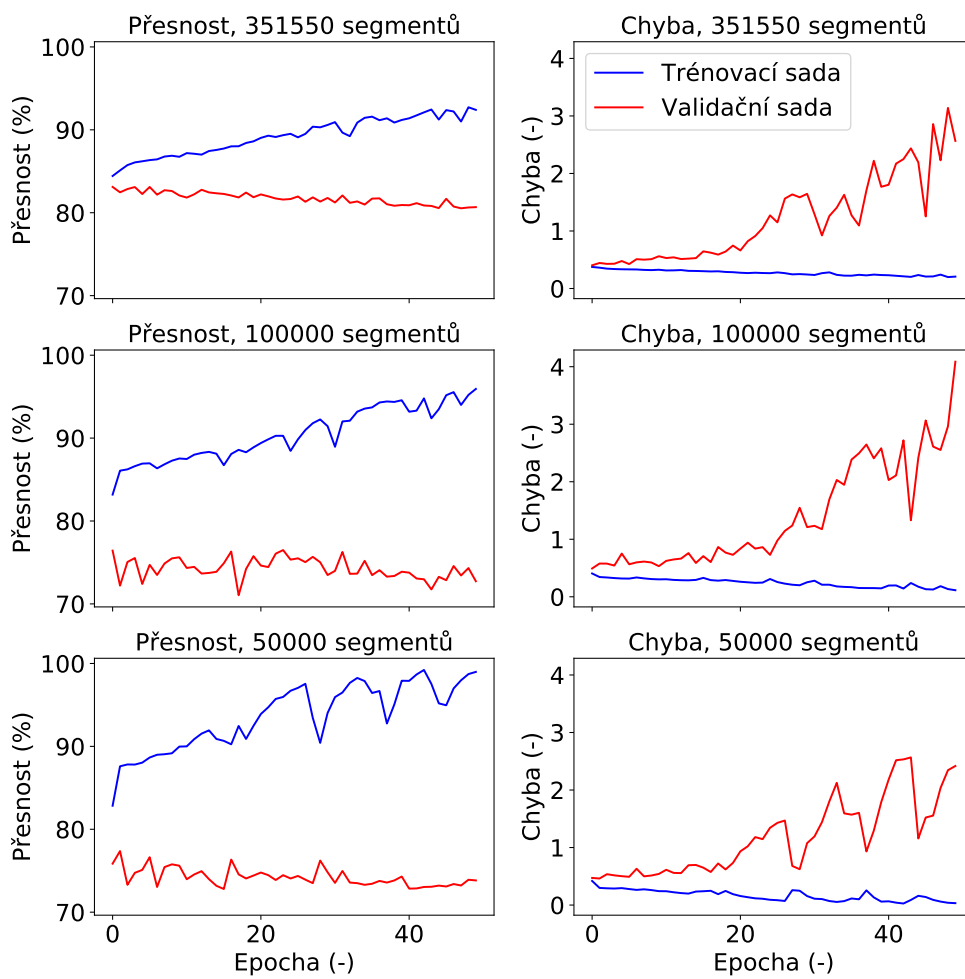
Název vrstvy	Parametry
2D CL	K (1,10) F 200
MaxPool	M (1,2)
Dropout	D 0
2D CL	K (2,20) F 100
MaxPool	M (1,2)
Dropout	D 0
2D CL	K (1,30) F 50
MaxPool	M (1,2)
Dropout	D 0
Flatten	
Dense	N 200
Dropout	D 0
Dense	N 100
Dropout	D 0
Dense	N 1

přesnost). Při učení sítě s 50000 segmenty si můžeme všimnout, že pokud lokálně klesne validační chyba, tak stoupne chyba trénovací a tím pádem klesne trénovací přesnost. Na tomto případu můžeme vidět, že tím, že se síť „vychýlila“ pomohla generalizaci. Tímto jsem zjistil, že se síť chová předvídatelně a její design je vhodný pro naši úlohu detekce.

#### 4.4.4 Hledání hyperparametrů

Celá neuronová síť obsahuje velké množství parametrů. Mezi takzvané hyperparametry se řadí: velikost konvolučního jádra, počet filtrů konvolučního jádra, velikost dávky dat, dropout a počty neuronů v plně propojených vrstvách. Tyto parametry ovlivňují finální přesnost celé sítě, proto je důležité najít jejich ideální kombinaci. Parametry se do určité části ovlivňují, ale poté je složité určit, jak který parametr ovlivnil výkon sítě.

Existuje více přístupů pro hledání ideálních hodnot těchto hyperparametrů. Mezi základní metody patří takzvané hledání naslepo. Tato metoda spočívá v tom, že se předem definuje rozsah daných parametrů a poté se vyzkouší síť naučit s různou kombinací těchto parametrů a jejich hodnot. Poté se pozoruje, které kombinace parametrů a jejich hodnot dává nejlepší výsledky. Tato metoda má jednu velkou nevýhodu, je časově náročná. Pro hledání parametrů byl využit skript *CNN\_sweep.py*.



Obrázek 4.18: Grafy průběhu přesnosti a chyby detektoru při rozdílné velikosti datové sady.



Tabulka 4.6: Rozmítané hodnoty velikosti konvolučního jádra pro jednotlivé konvoluční vrstvy (rozměr délky).

Vrstva	Hodnoty
1.	1, 5, 11, 21, 31, 41, 51
2.	1, 5, 11, 21, 31, 41, 51, 61
3.	1, 5, 11, 21, 31, 41, 51, 61, 71

Tabulka 4.7: Rozmítané hodnoty počtu neuronů v plně propojených vrstvách a počet filtrů v konvolučních vrstvách.

Vrstva	Zkoušené hodnoty
1. Plně propojená	1, 30, 100, 500
2. Plně propojená	1, 30, 100, 500
3. Konvoluční	1, 30, 60, 120, 300, 500
4. Konvoluční	1, 30, 60, 120, 300, 500
5. Konvoluční	1, 30, 60, 120, 300, 500

Prvním zkoumaným parametrem byla velikost konvolučního jádra. Tento parametr je nezávislý na počtu dat v datové sadě, tudíž bylo použito pouze 100000 segmentů. Předpokládá se, že ideální velikost konvolučního jádra nebude stejná v každé vrstvě. Hledané hodnoty jsou vidět v tabulce 4.6. Jedná se o druhý rozměr (délky). Maximální hodnoty se snižují, protože operace konvoluce každou vrstvou zmenšuje velikost matice a tím pádem by při velikosti 71 ve všech vrstvách (s efektem max-poolingu) nezbyla dostatečná velikost pro výpočet.

Hledání ideálního počtu filtrů pro jednotlivé konvoluční vrstvy a počtu neuronů v plně propojených vrstvách bylo provedeno na celé trénovací datové sadě. V tabulce 4.7 jsou vidět možnosti, jaké hodnoty mohly být vybrány. Prvních 5 nejlepších vypočtených kombinací bylo zprůměrováno, aby se docílilo větší generalizace sítě.

Cílem regularizačních vrstev je zabránit přeučení sítě. V našem designu sítě byl zvolen již popsáný dropout 4.4.1. Tento parametr byl hledán na celé datové sadě. Jeho rozmítané hodnoty byly od 0 do 0,8. Ideální velikost parametru dropout se pozná tak, že validační přesnost postupně stoupá s trénovací přesností (chyba společně klesá). Pokud pozorujeme, že přesnost na trénovací sadě rychle roste, ale přesnost na validační sadě rychle klesá, tak je dropout nastaven na příliš nízkou hodnotu. To je způsobeno tím, že se síť specializuje na trénovací sadu dat a my ji v tom málo zabraňujeme. Na druhou stranu, pokud nastavíme dropout na moc vysokou hodnotu (počet vyřazených neuronů bude vysoký), tak se síť nebude schopná dobře učit na trénovací sadě. Budeme pozorovat malou přesnost na trénovací sadě.

Parametr velikosti dávky byl studován, jako poslední. Rozmítané hodnoty velikostí byly 100, 250, 500, 1000, 2000. Větší dávky nebylo možné vyzkoušet z důvodu hardwarové limitace (výpočetní jednotka má omezenou velikost paralelních výpočtů).

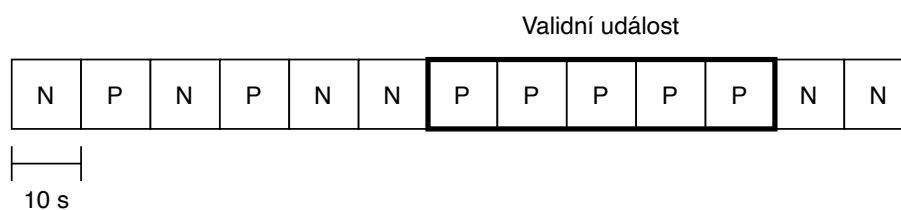
#### 4.4.5 Trénování, výběr finálního modelu

Trénování finální sítě (i všechny předchozí výpočty) byly provedeny za pomoci Akademie Věd ČR, ústavu fyziky plazmatu na grafických kartách: Nvidia GeForce RTX 2080 a Nvidia Titan V, což velice urychlilo výpočet. Pro trénování finální neuronové sítě byl použit skript *CNN\_2D.py*.

Datové sady jsou generovány náhodně, proto bylo vyzkoušeno více běhů trénování, aby se vyloučilo, že jsou data nerovnoměrně zastoupeny v jednotlivých sadách. Pro dosažení nejlepšího výsledku byla z více natrénovaných sítí vybrána ta, která měla nejmenší chybu na validační sadě dat. Pokud validační chyba postupně stoupala, tak byla vybrána epocha s nejnižší hodnotou chyby. Tato metoda se nazývá brzké zastavení (early stopping) [10].

### 4.5 Převod segmentů na značky

Pro účely trénování a samotné detekce byl původní signál rozdělen na segmenty. Protože však EEGviewer nepracuje se segmenty, ale s celým signálem, kde jsou události označeny značkami, tak je zapotřebí převést námi detekované segmenty zpět na původní signál a značky. Jelikož každý segment obsahuje minimálně 80 % události, znamená to, že pokud bylo přítomna událost v signálu, mělo by být pozitivních více segmentů za sebou viz obrázek 4.19. Tento přístup použili ve studii [47], kde událost brali za validní, pokud za sebou následovalo 5 segmentů s pozitivní událostí.



Obrázek 4.19: Ukázka převodu segmentů do reálného signálu. P značí segment pozitivní (obsahující detekovanou událost), N značí segment negativní (událost nedetekována).

## 4.6 Porovnání s kNN

Pro porovnání výsledků detekce hluboké neuronové sítě byla vybrána metoda nejbližšího souseda (kNN). Jedná se o klasickou metodu strojového učení s učitelem. Pokud je algoritmu předložen nový vzorek, tak se porovnají vzdálenosti ke  $k$  sousedním vzorkům z trénovací množiny. Podle těchto vzdáleností se vyhodnotí do jaké třídy předložený segment spadá. [55]

Protože algoritmus kNN pracuje pouze s jednorozměrným vektorem, byl vstupní signál průtoku a  $\text{SpO}_2$  spojen za sebe. Toto může být velká nevýhoda této metody. Volba velikosti koeficientu  $k$  se byla zvolena podle nejlepší přesnosti na menší sadě dat (50000 segmentů v každé datové sadě) z důvodu velké časové náročnosti na celé sadě. Pro výpočet kNN byl použit skript *knn.py*.

## 4.7 Detekce událostí

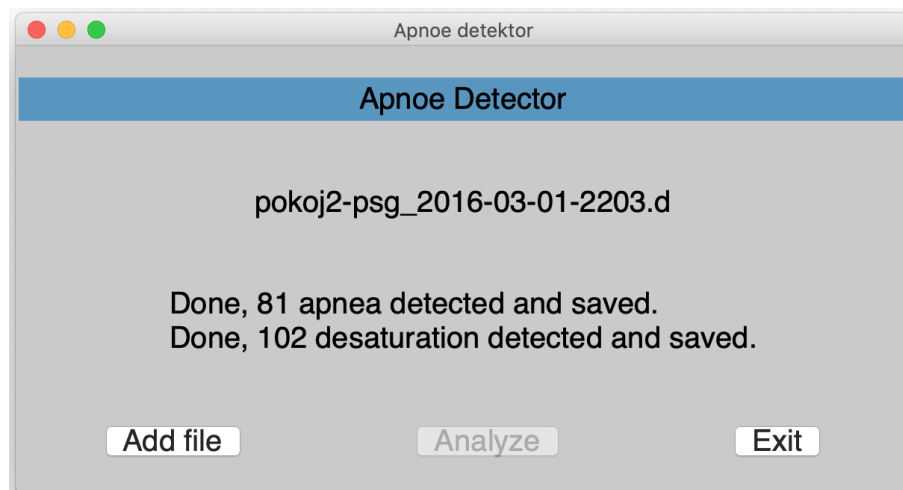
Proces detekce událostí na neohodnocených záznamech začíná stejným předzpracováním dat jako pro data, na kterých se detektor učil viz 4.3.1. Poté je provedena zvlášť detekce apnoe a desaturací. Detekované události jsou uloženy do nově zkopírovaného souboru s příponou „AUTO“. Proces ukládání je vytvořen podle manuálu k datovému formátu .d [41] a je převzat z předchozí bakalářské práce [40].

Výstupem z neuronové sítě je pravděpodobnost s jakou daný segment obsahuje událost. To nám umožňuje zvolit si práh podle toho, zda-li chceme větší specifitu nebo senzitivitu. Vztah těchto dvou veličin je zobrazen pomocí ROC křivky.

## 4.8 Implementace detektoru

Pro implementaci finálního detektoru do spánkové laboratoře v Národním stavu duševního zdraví v Klecanech byla zvolena forma jednoduchého grafického okna. Grafické rozhraní bylo vytvořeno pomocí knihovny AppJar (dostupné na [www.appjar.info](http://www.appjar.info)). Ukázkou vytvořeného okna lze vidět na obrázku 4.20. Postup je jednoduchý. Prvním krokem pomocí tlačítka „Add file“ vybereme soubor, který chceme zpracovat. Pokud se zobrazí jméno souboru, tak vybrání proběhlo bez problému. Tlačítkem „Analyze“ zapneme samotnou detekci událostí. Tento proces může zabrat jednotky minut. Po skončení se na obrazovce vypíše počet nalezených apnoe a desaturací. Nalezené události se uloží do nově zkopírovaného souboru. Tlačítko „Exit“ slouží k vypnutí programu. Grafické rozhraní se spustí pomocí skriptu *AD\_GUI.py*. Tento skript využívá funkce pro detekci ze skriptu *AD\_lunch.py*.

Pro správnou funkčnost je potřeba Python 3.6+ a tyto knihovny:



Obrázek 4.20: Ukázka implementace detektoru v grafickém okně.

- Keras 2.3.1,
- Tensorflow 2.0.0,
- AppJar 0.94.0,
- Numpy 1.18.0,
- Scipy 1.4.1,
- Struct,
- Window\_slider 0.8,
- Collections,
- Pytictoc,
- Shutil.

Všechny tyto knihovny jsou dostupné z <https://pypi.org/>.

## 4.9 Statistické metody

Pro sledování výsledků trénování a vyhodnocení účinnosti vytvořeného detektoru byla využita ROC analýza [56]. V tabulce 4.8 je vidět základní hodnocení segmentů. Pokud je 80 % dané události obsaženo v detekovaném segmentu, tak se počítá jako správně detekované (opačný proces při tvorbě datové sady z již ohodnocených záznamů). Vzhledem k tomu, že při trénování pracujeme s balancovanou datovou sadou, tak je jako hlavní parametr využita přesnost. Při detekci na reálném záznamu

však není možné zjistit celkový počet pravdivě negativních segmentů, proto nelze přesnost vypočítat. Výpočet je prováděn pomocí skriptu *CNN\_TK.py*, který obsahuje jednotlivé výpočty.

Tabulka 4.8: Možné hodnocení daného segmentu.

Zkratka	Význam	Vysvětlení
TP	True Positive	Správně pozitivní
FP	False Positive	Falešně pozitivní
TN	True Negative	Správně negativní
FN	False Negative	Falešně negativní

**Senzitivita** ukazuje poměr správně detekovaných událostí vůči všem pozitivním událostem. Výpočet senzitivity je definován vztahem [56]:

$$\text{Senzitivita} = \frac{TP}{TP + FN} \quad (4.6)$$

**Specificita** ukazuje poměr správně nedetekovaných událostí vůči všem správně negativních událostí. Výpočet specificity je definován vztahem [56]:

$$\text{Senzitivita} = \frac{TN}{TN + FP} \quad (4.7)$$

**Přesnost** ukazuje poměr správně událostí vůči všem událostem. Výpočet přesnosti je definován vztahem [56]:

$$\text{Senzitivita} = \frac{TP + TN}{TP + TN + FN + FP} \quad (4.8)$$

**Pozitivní prediktivní hodnota** (PPV) ukazuje poměr správně detekovaných událostí, vůči všem detekovaných událostí. Výpočet pozitivní prediktivní hodnoty je definován vztahem [56]:

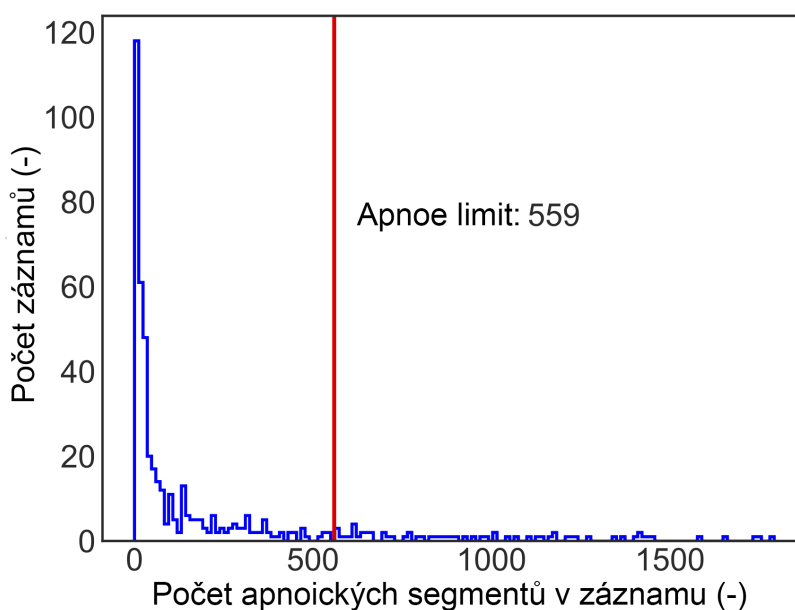
$$PPV = \frac{TP}{TP + FP} \quad (4.9)$$

## 5 Výsledky

Výsledky práce jsou rozděleny do tří částí. První část prezentuje výsledky studie spánkových příznaků. Část druhá ukazuje výsledky návrhu designu hluboké konvoluční neuronové sítě a jejích parametrů. Část třetí obsahuje výsledky detektoru pomocí NN a porovnává je s výsledky získané metodou kNN.

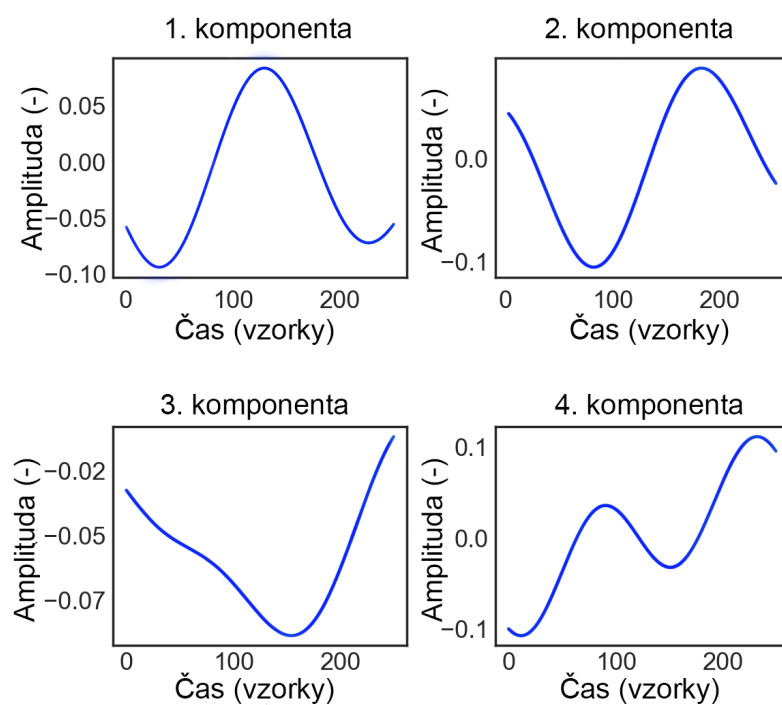
### 5.1 Studie spánkových příznaků

Pro tvorbu datové sady byl vypočten maximální limit počtu apnoických segmentů z jednoho záznamu na 559 podle vzorce 4.3. Histogram počtu apnoických segmentů v jednotlivých záznamech s vypočteným limitem můžeme vidět na grafu z obrázku 5.1.

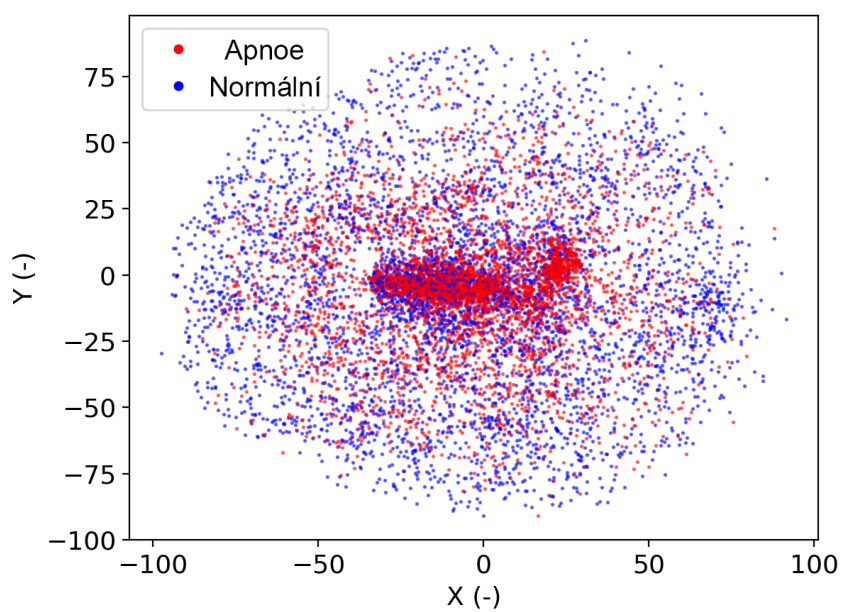


Obrázek 5.1: Histogram počtu apnoických segmentů v záznamu s maximálním limitem.

Klasifikace pomocí kNN byla provedena na data surová a na data u nichž byla zredukována dimenze pomocí PCA a t-SNE. Výsledky klasifikace jsou uvedeny v tabulce 5.1. Na grafu z obrázku 5.2 jsou zobrazeny první 4 komponenty (vlastní vektory), které celkem reprezentují 89% dat. Graf z obrázku 5.3 zobrazuje redukováná data pomocí t-SNE. Bylo vykresleno pouze 10000 vzorků, z důvodu přehlednějšího zobrazení.



Obrázek 5.2: Zobrazení prvních 4 komponent (vlastních vektorů) z PCA.



Obrázek 5.3: Zobrazení dat po redukci dimenze pomocí t-SNE.

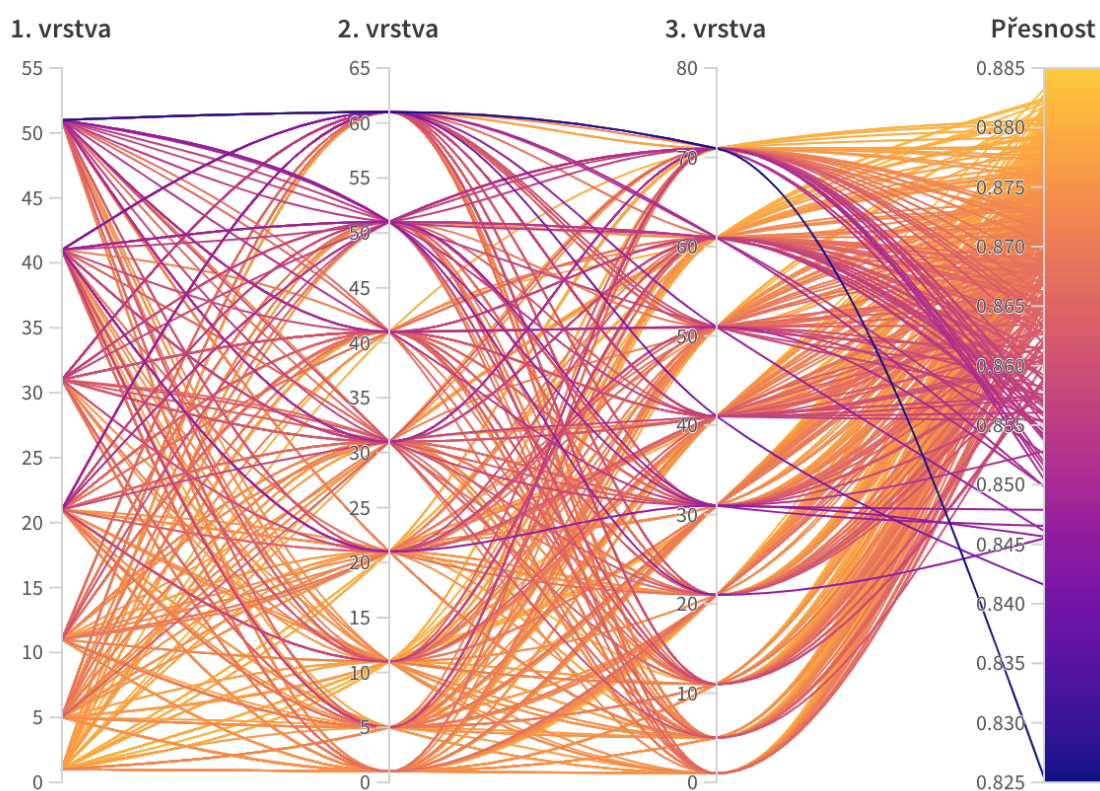
Tabulka 5.1: Výsledky klasifikace pro metody redukce dimenze.

	Přesnost (%)	Sensitivita (%)	Specifická (%)	Čas (s)
Surová data	75,89	76,72	75,06	234,00
PCA	75,01	74,26	75,73	2,32
t-SNE	71,52	68,85	74,21	0,22

## 5.2 Design hluboké neuronové sítě

Výsledky designu hluboké neuronové sítě se skládají z průběžných výsledků hledání ideálních parametrů a z finální struktury hluboké konvoluční neuronové sítě.

Pro zjištění ideální velikosti konvolučních jader bylo náhodně vyzkoušeno 416 kombinací o délce 50 epoch. Celkově tento výpočet trval 81 hodin. Na paralelním grafu z obrázku 5.4 můžeme vidět jednotlivé vyzkoušené kombinace a jejich výsledek na barevné škále. V tabulce 5.2 je vybráno 5 nejlepších kombinací dle trénovací přesnosti. Protože se výsledky liší minimálně, byl udělán aritmetický průměr každého rozměru vrstvy. Výsledné rozměry vrstev byly spočítány na: 1, 15, 45. Pro ověření byly vyzkoušeny dodatečně další kombinace viz tabulka 5.3, ale i přesto vyšla délka 1 v 1. konvoluční vrstvě nejlépe.



Obrázek 5.4: Paralelní graf zobrazující jednotlivé kombinace a jejich výsledek (generováno z W&B).



Tabulka 5.2: Výsledky prvních 5 nejlepších kombinací při hledání optimálních velikostí konvolučního jádra (rozměru délky, dle trénovací přesnosti).

Délka konvolučního jádra			Přesnost (%)	Chyba (-)
1.vrstva	2.vrstva	3.vrstva		
1	11	41	88,32	0,2930
1	11	31	88,27	0,2919
1	11	31	88,26	0,2911
1	21	61	88,23	0,2903
1	21	61	88,22	0,2927

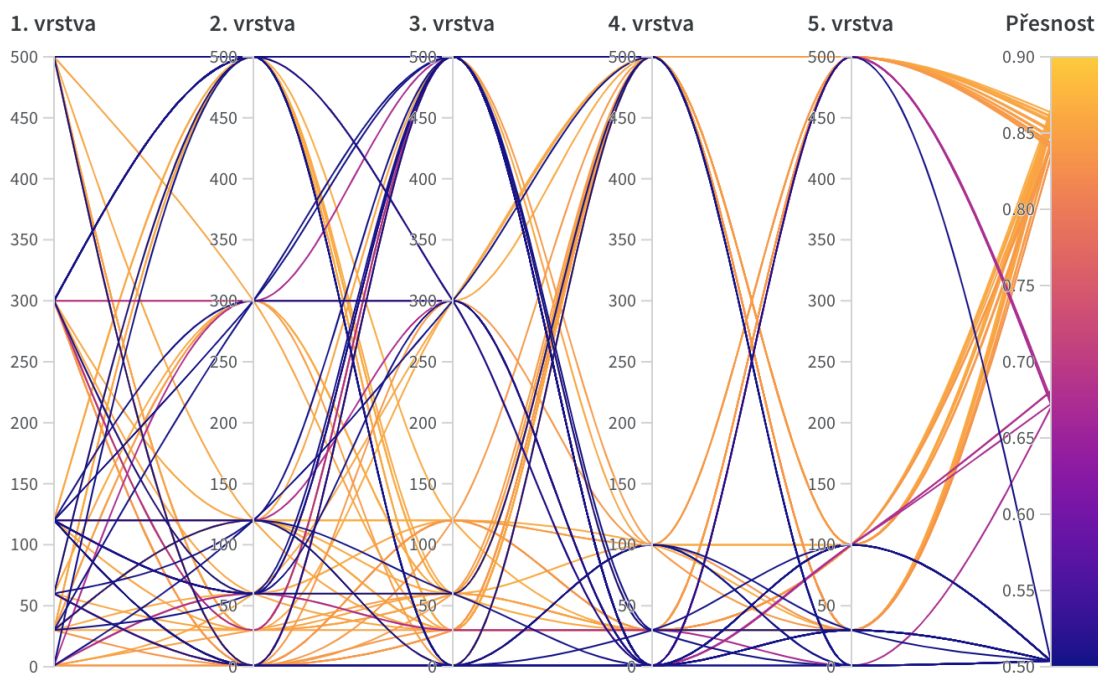
Tabulka 5.3: Výsledek dodatečné zkoušky kombinací velikostí konvolučního jádra pro 1. vrstvu (rozměru délky, dle trénovací přesnosti).

Délka konvolučního jádra			Přesnost (%)	Chyba (-)
1.vrstva	2.vrstva	3.vrstva		
1	15	45	88,11	0,2947
2	15	45	87,84	0,2988
3	15	45	87,76	0,2992

Pro zjištění ideálního počtu neuronů v plně propojených vrstvách a počtu filtrů pro vrstvy konvoluční bylo vypočteno 79 náhodných kombinací. Celkově výpočet trval 92 hodin. Na paralelním grafu z obrázku 5.5 můžeme vidět jednotlivé vyzkoušené kombinace a jejich výsledky pomocí barevné škály. V tabulce 5.4 je zobrazeno prvních 5 nejlepších kombinací (řazeno dle přesnosti na trénovací sadě). Počet neuronů v plně propojené vrstvě je 148 a 86 (1. a 2. plně propojená vrstva). Počet filtrů pro jednotlivé konvoluční vrstvy je: 174 (3. konvoluční vrstva), 308 (4. konvoluční vrstva) a 96 (5. konvoluční vrstva).

Tabulka 5.4: Výsledek hledání ideálního počtu neuronů v plně propojených vrstvách a počet filtrů ve vrstvách konvolučních (dle trénovací přesnosti).

Počet filtrů v konvoluční vrstvě			Počet neuronů v plně propojené vrstvě		Přesnost (%)	Chyba (-)
1.	2.	3.	4.	5.		
30	120	120	100	30	86,58	0,3349
120	500	120	30	100	86,51	0,3361
300	300	60	500	100	86,47	0,3330
120	120	120	100	100	86,45	0,3339
300	500	60	100	100	86,38	0,3386



Obrázek 5.5: Paralelní graf zobrazující jednotlivé kombinace počtu neuronů v plně propojených vrstvách, počtu filtrů v konvolučních vrstvách a jejich výsledek (generováno z W&B).

Pro zjištění ideálního parametru dropout jsou výsledky zobrazeny v tabulce 5.5 pro trénovací a validační sadu. Na grafu z obrázku 5.6 je zobrazen postupný vývoj přesnosti a chyby během trénování. Největší validační přesnosti dosáhl dropout 0,6.

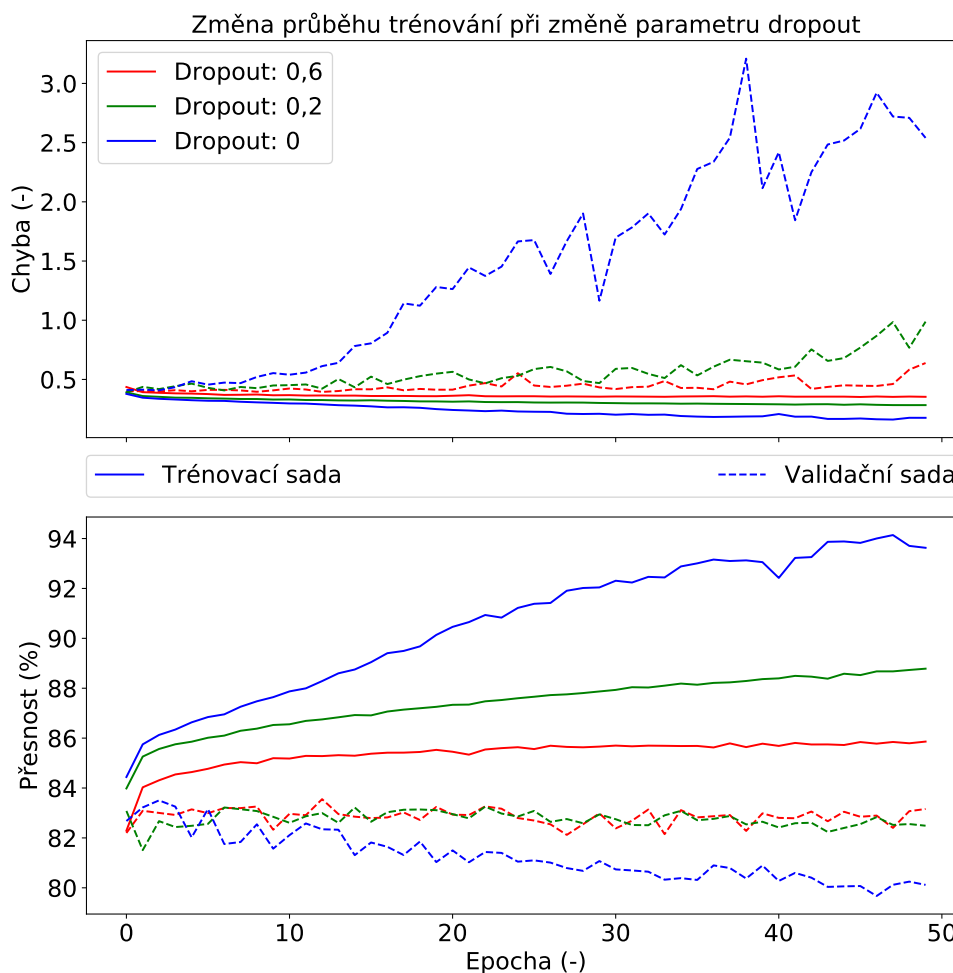
Nejlepší velikost dávky vzhledem k přesnosti na validační sadě vyšla 1000. Výsledky pro rozmítané hodnoty jsou vidět v tabulce 5.6.

Finální design sítě je zobrazen v tabulce 5.7. Celkový počet vah je 3477817. Velikost dávky byla zvolena 1000. Ostatní parametry byly ponechány podle výchozího nastavení knihovny Keras. Tento design sítě byl využit pro trénování sítí, které slouží k detekci apnoí a desaturací.

Trénování finální sítě proběhlo na více náhodně vygenerovaných datových sadách. Na grafu z obrázku 5.7 je zobrazen průběh trénování neuronové sítě pro detekci apnoí. Na grafu z obrázku 5.8 je zobrazen průběh trénování neuronové sítě pro detekci desaturací. Pro finální model byla vybrána síť a konkrétní epocha s nejmenší validační chybou. Epochy číslo 5 pro detekci apnoí a epocha číslo 15 pro detekci desaturací obsahují běh s nejmenší validační chybou. Finální modely sítí pro tyto epochy jsou uloženy na příloženém CD/DVD viz příloha B.

Tabulka 5.5: Výsledky trénování pro různé nastavení parametru dropout (T. = trénovací sada, V. = validační sada).

Dropout	T. přesnost (%)	T. chyba (-)	V. přesnost (%)	V. chyba (-)
0,0	93,63	0,1758	80,12	2,5400
0,1	91,77	0,2140	81,50	1,4671
0,2	88,78	82,49	0,9880	
0,3	87,74	0,3042	82,74	0,7851
0,4	87,14	0,3191	82,51	1,3302
0,5	86,60	0,3305	82,54	0,8224
0,6	85,86	0,3531	83,16	0,6386
0,7	85,18	0,3714	82,75	0,4185
0,8	82,87	0,4556	80,41	0,5000



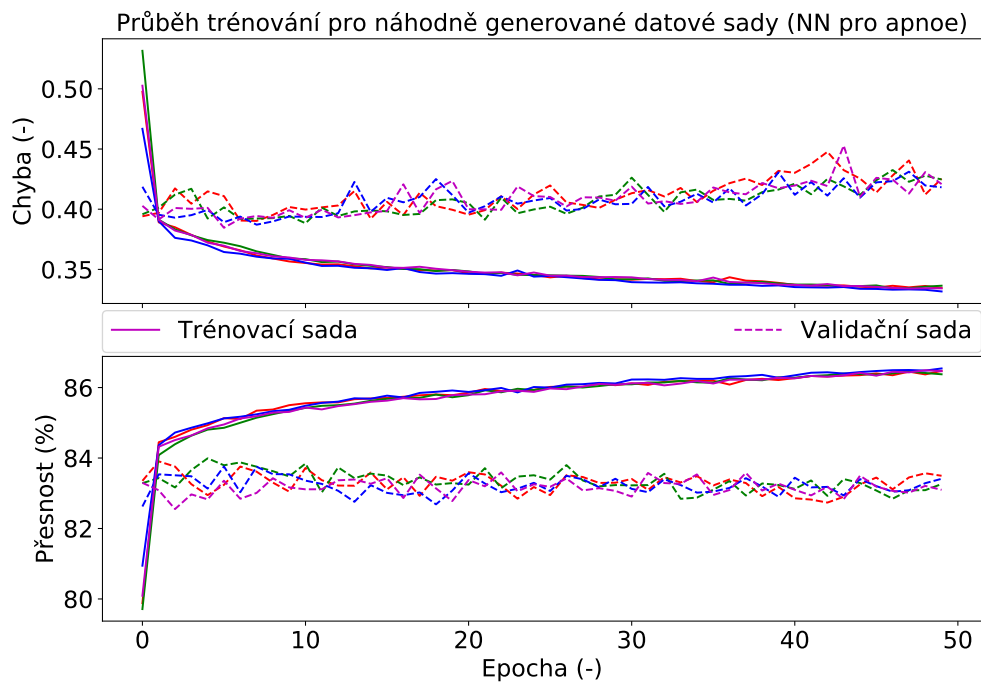
Obrázek 5.6: Graf zobrazující průběh trénování pro různé nastavení parametru dropout (legenda platí pro oba grafy).

Tabulka 5.6: Výsledky trénování pro různé nastavení velikosti dávky.

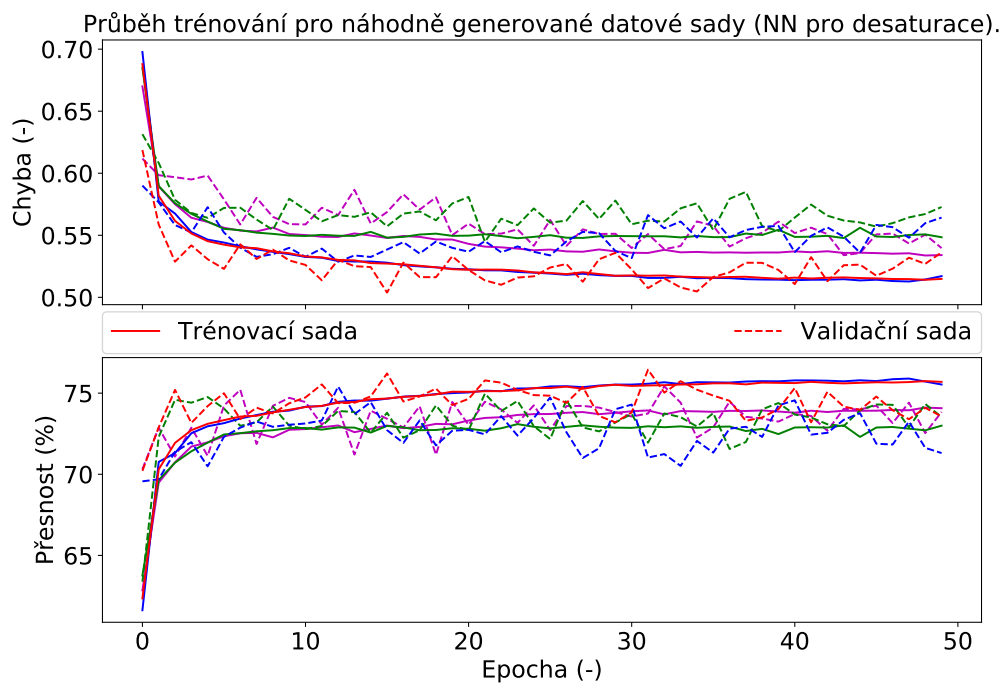
Velikost dávky	V. přesnost (%)	V. chyba (-)
2000	83,23	0,3973
1000	83,39	0,4004
500	82,58	0,4252
250	82,58	0,4248
100	82,81	0,4173

Tabulka 5.7: Finální struktura konvoluční neuronové sítě použité pro detekci apnoí a desaturací. Značení parametrů je popsáno v tabulce 4.4

Název vrstvy	Parametry	Počet vah	Rozměr dat
2D CL	K (1,1) F 174	348	2 x 500 x 174
MaxPool	M (1,2)		2 x 250 x 174
Dropout	D 0,6		
2D CL	K (2,15) F 308	1608068	1 x 236 x 308
MaxPool	M (1,2)		1 x 118 x 308
Dropout	D 0,6		
2D CL	K (1,45) F 96	1330656	1 x 74 x 96
MaxPool	M (1,2)		1 x 37 x 96
Dropout	D 0,6		
Flatten			3552
Dense	N 148	525844	148
Dropout	D 0,6		
Dense	N 86	12814	86
Dropout	D 0,6		
Dense	N 1	87	1



Obrázek 5.7: Průběh trénování finální neuronové sítě pro detekci apnoí.



Obrázek 5.8: Průběh trénování finální neuronové sítě pro detekci desaturací.

### 5.3 Výsledky detekce spánkových událostí

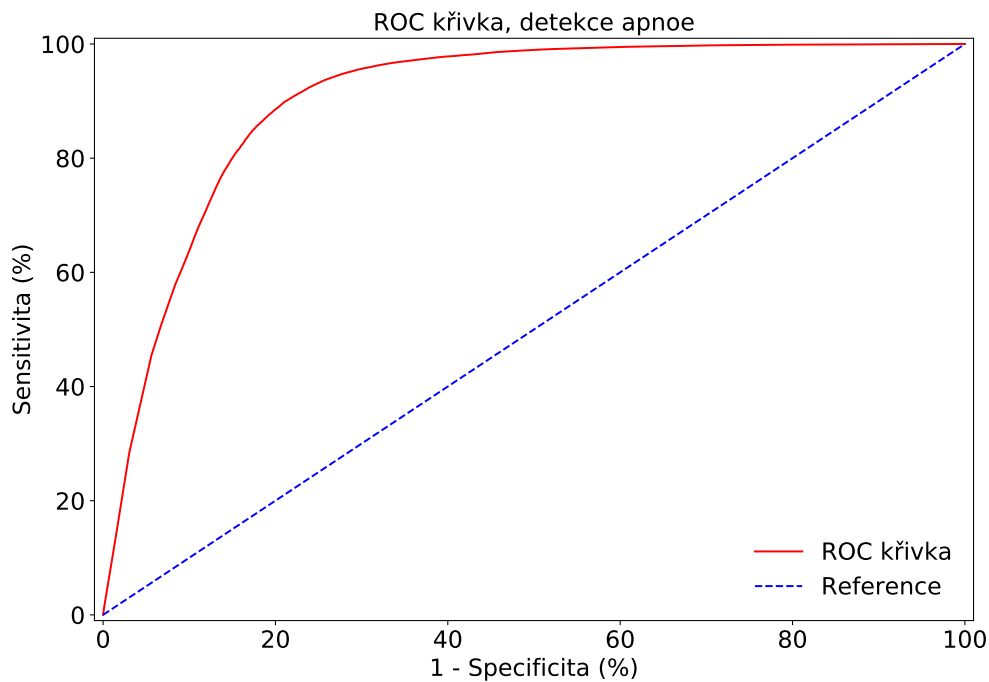
Výsledky detektoru spánkových událostí jsou vypočítány na testovací sadě dat, která obsahovala vždy celé subjekty. Stejná testovací sada dat byla použita pro detekci pomocí metody kNN. Výsledky byly zpracovány pomocí ROC analýzy.

#### 5.3.1 Detekce apnoí

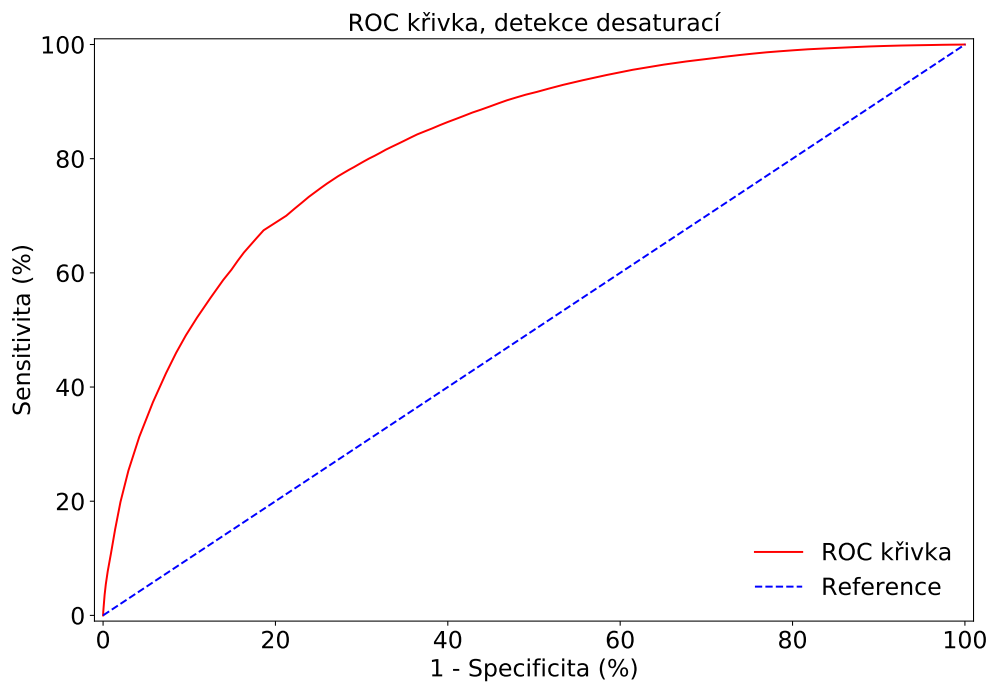
Výsledek detekce apnoických segmentů pomocí hluboké neuronové konvoluční sítě je zobrazen v tabulce 5.8 pro různě zvolené prahy. Graf z obrázku 5.9 zobrazuje ROC křivku, která popisuje vztah senzitivity a specifity v závislosti na hodnotě prahu.

Tabulka 5.8: Výsledky klasifikace apnoí pomocí neuronové sítě pro různé prahy (testovací sada).

Práh	Přesnost (%)	Senzitivita (%)	Specifita (%)
0,3	82,21	96,22	68,21
0,5	84,27	88,39	80,14
0,7	82,92	81,64	84,21



Obrázek 5.9: ROC křivka pro klasifikaci apnoí pomocí neuronové sítě.



Obrázek 5.10: ROC křivka pro klasifikaci desaturací pomocí neuronové sítě.

Tabulka 5.9: Výsledky klasifikace desaturací pomocí neuronové sítě pro různé prahy.

Práh	Přesnost (%)	Senzitivita (%)	Specifická (%)
0,3	69,04	93,74	44,34
0,5	74,19	82,62	65,76
0,7	74,41	67,46	81,37

### 5.3.2 Detekce desaturací

Výsledek detekce segmentů s přítomnou desaturací pomocí hluboké neuronové konvoluční sítě je zobrazen v tabulce 5.9 pro různě zvolené prahy. Graf z obrázku 5.10 zobrazuje ROC křivku, která popisuje vztah senzitivity a specifické v závislosti na hodnotě prahu.

### 5.3.3 Detekce pomocí kNN

Výsledky pro různá  $k$  na zmenšené (50000 segmentů) validační sadě jsou vidět v tabulce 5.10. Nejlepší koeficient  $k$  pro detekci apnoí je  $k = 20$  a pro detekci desaturací  $k = 500$ .

Tabulka 5.10: Výsledné přesnosti na zmenšené validační sadě, pro volbu ideálního  $k$ . Zvlášť počítané pro apnoe a desaturace.

Koeficient $k$	Přesnost (%)	Přesnost (%)
	Detekce apnoí	Detekce desaturací
5	83,06	55,77
20	<b>84,22</b>	57,74
50	84,03	59,11
100	83,71	59,93
200	82,81	60,78
500	80,96	<b>61,61</b>
1000	78,91	61,37

Tabulka 5.11: Výsledky klasifikace pomocí kNN.

	Přesnost (%)	Sensitivita (%)	Specificita (%)
Apnoe	83,39	88,31	78,47
Desaturace	64,33	54,05	74,61

V tabulce 5.11 je vidět výsledek klasifikátoru pro apnoe a desaturace. Hodnota koeficientu  $k$  pro klasifikaci apnoe je 20, pro klasifikaci desaturací 500, viz tabulka 5.10.



## 6 Diskuse

Hlavním výstupem této práce je vytvořená hluboká konvoluční neuronová síť, která byla naučena pro detekci spánkové apnoe a desaturace v PSG signálu. Tato neuronová síť byla porovnána s metodou nejbližšího souseda. Dalším výstupem této práce je studie signálových příznaků, ve které jsem porovnal lineární a nelineární metody redukce dimenze dat a jejich vliv na detekci spánkového apnoe.

Diskuze byla rozdělena na 3 části, které na sebe navazují. První část se věnuje diskutování výsledků studie signálových příznaků. Část druhá diskutuje výsledky návrhu designu neuronové sítě a volbu jejích parametrů. Poslední část se věnuje výsledkům detekce natrénované neuronové sítě, jejího porovnání s metodou nejbližšího souseda a obecným závěrům.

### 6.1 Studie signálových příznaků

Studie signálových příznaků nám ukázala, že pokud je potřeba použít metodu redukci dimenze pro detekci apnoí na signálu průtoku, tak je vhodné použít lineární metodu (PCA).

Výsledky v tabulce 5.1 ukazují, že detekce na surových datech má nejlepší výsledky. Její časová náročnost je však řádově větší. Výsledek klasifikace aplikací PCA je téměř shodný, ale klasifikace trvala pouze zlomek času. Nejhoršího výsledku dosahuje klasifikátor na datech po použití t-SNE, avšak časová náročnost následné klasifikace je v tomto případě nejnižší.

Graf z obrázku 5.2 zobrazuje první 4 komponenty z PCA. První dvě komponenty nám ukazují sinovou a kosinovou složku rozloženého signálu. Tento jev pozorujeme především z toho důvodu, že pracujeme pouze se záznamem dechu, který je těmito signály primárně tvořen. Třetí a čtvrtá komponenta mohou zobrazovat počátek a konec apnoické pauzy. Čtvrtá komponenta zobrazuje zvýšení amplitudy, což může ukazovat na konec apnoické pauzy. Tyto 4 komponenty reprezentují celkem 89% dat. Limitací metody PCA je, že předpokládá frekvenci signálu za stálou. Ve skutečnosti stálá frekvence není, protože každý subjekt dýchá s jinou frekvencí.

Jak můžeme vidět, tak výsledky klasifikace z tabulky 5.1 naznačují, že použití t-SNE není vhodné. Předpokládáme, že amplituda dechu je lineární, tudíž není vhodné využít nelineární redukci dimenzi dat. Dále je možné, že algoritmus t-SNE našel další nelineární příznaky, které mohou být nerelevantní k detekci apnoe. Na grafu z obrázku 5.3 je možné pozorovat, že se apnoické vzorky sdružují v centru.

## 6.2 Návrh designu neuronové sítě

Základní přibližná hloubka neuronové sítě byla stanovená podle studie [47]. Její parametry (velikost konvolučního jádra, počet neuronů v plně propojených vrstvách, počet filtrů v konvolučních vrstvách, velikost parametru dropout a velikost dávky) byly dále laděny pro dosažení největší validační přesnosti na validační sadě dat. Limitací metodiky návrhu hluboké neuronové sítě je návrh parametrů. Parametry hluboké neuronové sítě byly laděny každý separátně, což by bylo efektivní v případě, že by na sobě byly tyto parametry nezávislé. V případě závislosti jednotlivých parametrů hluboké neuronové sítě by bylo vhodné testovat tyto parametry simultánně. Negativem takového testování je, že neumožňuje sledovat a hodnotit efekt jednotlivých parametrů na trénování hluboké neuronové sítě.

Výsledný nalezený rozměr konvolučních jader byl 1, 15, 45. Postupné zvětšování konvolučního jádra bylo očekáváno, protože tím je zaručeno, že postupně do vrstev pronikají příznaky o různé délce. Velikost konvolučního jádra 1 v první konvoluční vrstvě nemá zdánlivě žádný vliv (z podstaty výpočtu konvoluce). Proto byly dodečně otestovány velikosti 2 a 3 (při původním testování byla nejbližší hodnota 5). Výsledky tohoto dodatečného testování v tabulce 5.3 ukazují, že velikost 1 podávala nejlepší výsledky. Tento poznatek je vhodné dále analyzovat. Je možné, že by se tato vrstva mohla vynechat, či nahradit vrstvou jednodušší a tím by se zrychlil výpočet a samotná detekce.

Graf z obrázku 5.6 ukazuje vybrané průběhy trénování neuronové sítě. Je vidět, že ať byla velikost parametru dropout jakákoliv, tak se nám nepodařilo docílit stoupající validační přesnosti při delším počtu epoch. Lze pozorovat, že při hodnotě dropout 0 validační přesnost prudce klesala a velikost chyby rychle stoupala. Na druhou stranu, lze pozorovat, že trénovací přesnost byla nejvyšší. Při nastaveném parametru dropout na 0,2 a 0,6 pozorujeme mírné stoupání chyby a klesání validační přesnosti. Avšak trend stoupající přesnosti na trénovací sadě dat s poklesem parametru dropout zde pozorujeme též. Byla využita metoda brzkého zastavení učení, aby se zabránilo přeučení sítě [10]. Tato metoda není ideální, protože zabrání dalšímu učení sítě. Bylo by vhodné vyzkoušet i jiné regularizační metody.

Parametr velikosti dávky dat, které projdou neuronovou sítí současně (paralelně) byl po provedení testování 5 hodnot nastaven na 2000. Čím větší velikost dávky je, tím více klesá přesnost na trénovací sadě [57]. V našem případě větší dávka zajistila i větší regularizaci a tím pádem vyšší přesnost na validační sadě. Tento výstup byl očekáván, protože se využitím vyšší dávky do sítě zanesou určitá nepřesnost, která zvýší regularizaci. Vyšší dávka nemohla být v této studii testována z důvodu limitace hardwaru.

Celá neuronová síť a všechny parametry byly laděny pro síť, která má za úkol detekovat spánková apnoe. Síť pro detekci desaturací byla naučena na stejném finálním designu sítě se stejnými parametry. Protože vstupní data jsou generována ze stejných biologických signálů, předpokládáme, že jejich požadavky na parametry sítě budou stejné. Tento přístup byl zvolen z důvodu nedostatku času. V budoucí práci by bylo žádoucí provést hledání ideálních parametrů pro každou síť zvlášť.

Trénování finálního designu sítě bylo provedeno 4x na náhodně generované sadě dat. Na grafu z obrázku 5.7 je zobrazen průběh trénování neuronové sítě pro detekci apnoí. Podle podobného průběhu lze usoudit, že síť se chová stejně pro různá trénovací a validační data u sítě pro detekci apnoí. Na grafu z obrázku 5.8 je zobrazen průběh trénování neuronové sítě pro detekci desaturací. Byl vybrán průběh s nejmenší validační chybou. Lze pozorovat, že validační chyba vzrůstá s průběhem trénování u obou sítí. Zároveň klesá validační přesnost. To je známka přeučení sítě. Pro zabránění tohoto jevu byla využita ne finální epocha, ale ta, která měla nejmenší validační chybu v průběhu trénování. Tato metoda brzkého zastavení učení se využije pouze v tomto případě.

### 6.3 Přesnost detektoru a porovnání s kNN

Finální přesnost navržené neuronové sítě je pro detekci apnoí 84 % a pro detekci desaturací 74 %. Přesnost detektoru za využití metody nejbližšího souseda dosáhla 83 % při detekci apnoí a 64 % při detekci desaturací. Z těchto výsledků je patrné, že pro detekci apnoí dosahuje metoda nejbližšího souseda podobný výsledek, avšak pro detekci desaturací je přesnost nižší o 10 % oproti hluboké neuronové síti.

Při převodu detekovaných segmentů do jednotlivých značek jsem vycházel z práce [47], kde jako validní událost značí takovou, která má minimálně 5 za sebou pozitivně detekovaných segmentů. Tento parametr má velký vliv na výslednou senzitivitu a specificitu detektoru. V budoucí práci by bylo vhodné prostudovat chování detektoru při změně minimálního počtu pozitivních segmentů.

Detekce pomocí hluboké neuronové sítě má své výhody oproti detekci pomocí kNN. Mezi hlavní patří časová náročnost výpočtu. Detekce apnoí pomocí neuronové sítě trvá na 8 h záznamu 36 s zatímco s využitím kNN 45 min (měřeno na procesoru AMD Ryzen 9 3900X). Výpočet kNN pro tak velké množství trénovacích dat je velmi náročný a na pomalejším procesoru by trval násobně déle, což je pro praktické využití nevhodné. Nevýhoda využití neuronové sítě z hlediska hardwaru je taková, že vyžaduje procesor, který podporuje instrukční sadu AVX (Advanced Vector Extensions). Podpora AVX byla uvedena u většiny procesorů od roku 2013, tudíž při implementaci v NUDZ bylo možné detektor nainstalovat pouze na novější počítače na oddělení. Další výhoda detekce pomocí neuronové sítě je možnost řídit práh,

což ovlivňuje senzitivitu a specificitu, viz ROC křivky 5.9 a 5.10. Tato skutečnost je důležitá pro praktické využití. Požadavek od lékařek z NUDZ byl takový, aby detektor nacházel raději více událostí s rizikem falešně pozitivních segmentů. To znamená, že požadují, aby měl detektor vysokou senzitivitu. V aktuální implementaci detektoru je práh pro uznání segmentu jako pozitivního pevně stanovený na 0,5 a proto není možné ručně volit práh při uživatelském užívání detektoru. V budoucí verzi by bylo možné tuto volbu přidat. To by zajistilo větší využitelnost, protože uživatel by si mohl zvolit jak moc senzitivní chce mít provedenou detekci, což by pomohlo správné diagnostice u pacientů s extrémně velkým či extrémně malým počtem spánkových událostí.

Výslednou přesnost detekcí je možné v další práci vylepšit za pomoci využití více PSG kanálů, jako je například EKG, EEG, EMG a další. Všechny tyto biologické signály obsahují informace o spánkových událostech, proto předpokládáme, že by jejich využití pomohlo s detekcí.

## 7 Závěr

Byl vytvořen detektor spánkové apnoe a poklesu saturace krve kyslíkem na bázi hluboké konvoluční neuronové sítě. Pro implementaci byl použit jazyk Python a knihovna TensorFlow.

Byla vypracována metodika předzpracování dat pro neuronovou síť, zahrnující studii vlivu metod redukce dimenze dat. V té byl porovnán vliv lineární (PCA) a nelineární (t-SNE) metody na klasifikaci. Ve výsledku dosahoval klasifikátor nejvyšší přesnosti na surových datech, přesnost na datech redukováných lineárně byla mírně nižší, avšak o několik řádu rychlejší. Nelineární redukce se neukázala jako vhodná.

Pro natrénování neuronové sítě byla použita data z Oddělení spánkové medicíny v NUDZ. Po optimalizaci a vyhodnocení parametrů neuronové sítě bylo provedeno porovnání s detekcí pomocí metody nejbližšího souseda.

Výsledná přesnost neuronové sítě na testovací datové sadě při detekci apnoí je 84 % a při detekci desaturací 74 %. Výsledek pomocí kNN pro detekci apnoí je 83 % a pro detekci desaturací 64 %. Z praktického hlediska je výpočet pomocí neuronové sítě řádově rychlejší a má možnost volitelného prahu, který mění výslednou senzitivitu a specifitu, podle preferencí uživatele.

Vytvořený automatický detektor byl vybaven uživatelsky přívětivým grafickým rozhraním a byl nasazen do použití na Oddělení spánkové medicíny v NUDZ.

Studie signálových příznaků byla publikována na konferenci EHB (e-Health and Bioengineering) v roce 2019 viz příloha C.

## Seznam použité literatury

1. CRUZ, Alvaro A. *Global surveillance, prevention and control of chronic respiratory diseases: a comprehensive approach*. World Health Organization, 2007.
2. PURVES D Augustine GJ, Fitzpatrick D. *Neuroscience*. 2. vyd. Westchester Illinois: Sunderland (MA): Sinauer Associates, 2001.
3. *Brain Basics: Understanding Sleep*. Bethesda, MD 20892: Office of Communications a Public Liaison, 2019. Dostupné také z: <https://www.ninds.nih.gov/Disorders/Patient-Caregiver-Education/Understanding-Sleep>.
4. FRAIWAN, Luay; LWEESY, Khaldon; KHASAWNEH, Natheer; WENZ, Heinrich; DICKHAUS, Hartmut. Automated sleep stage identification system based on time–frequency analysis of a single EEG channel and random forest classifier. *Computer methods and programs in biomedicine*. 2012, roč. 108, č. 1, s. 10–19.
5. SATEIA, Michael J. International classification of sleep disorders. *Chest*. 2014, roč. 146, č. 5, s. 1387–1394.
6. CONRAD, Iber; ANCOLI.ISRAEL, Sonia; CHESSON, Andrew L. *The AASM Manual for Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications*. 1. vyd. Westchester Illinois: AASM in U.S.A, 2007. Dostupné z DOI: [10.9657220-4-X](https://doi.org/10.9657220-4-X).
7. ŠOŇKA, Karel. *Apnoe a další poruchy dýchání ve spánku*. 1. vyd. Praha: Grada, 2004. ISBN 80-247-0430-7.
8. VAŠUTOVÁ, Pharmdr Kateřina; SLUNCE, Lékárna. Spánek a vybrané poruchy spánku a bdění. *Praktické lékařství*. 2009, roč. 5, č. 1, s. 17–20.
9. BĚLEHRAD, M et al. Vyšetření poruch dýchání ve spánku od A do Z. *Kardiologickarevue*. 2017, roč. 15, č. 2, s. 79–86.
10. GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
11. DEO, Rahul C. Machine learning in medicine. *Circulation*. 2015, roč. 132, č. 20, s. 1920–1930.
12. HASSAN, Ahnaf Rashik. Computer-aided obstructive sleep apnea detection using normal inverse Gaussian parameters and adaptive boosting. *Biomedical Signal Processing and Control*. 2016, roč. 29, s. 22–30.
13. SONG, Changyue; LIU, Kaibo; ZHANG, Xi; CHEN, Lili; XIAN, Xiaochen. An obstructive sleep apnea detection approach using a discriminative hidden Markov model from ECG signals. *IEEE Transactions on Biomedical Engineering*. 2015, roč. 63, č. 7, s. 1532–1542.

14. VARON, Carolina; CAICEDO, Alexander; TESTELMANS, Dries; BUYSE, Bertien; VAN HUFFEL, Sabine. A novel algorithm for the automatic detection of sleep apnea from single-lead ECG. *IEEE Transactions on Biomedical Engineering*. 2015, roč. 62, č. 9, s. 2269–2278.
15. URTNASAN, Erdenebayar; PARK, Jong-Uk; JOO, Eun-Yeon; LEE, Kyoung-Joung. Automated detection of obstructive sleep apnea events from a single-lead electrocardiogram using a convolutional neural network. *Journal of medical systems*. 2018, roč. 42, č. 6, s. 104.
16. LI, Kunyang; PAN, Weifeng; LI, Yifan; JIANG, Qing; LIU, Guanzheng. A method to detect sleep apnea based on deep neural network and hidden markov model using single-lead ECG signal. *Neurocomputing*. 2018, roč. 294, s. 94–101.
17. DE CARLI, Fabrizio; NOBILI, Lino; GELCICH, Paola; FERRILLO, Franco. A method for the automatic detection of arousals during sleep. *Sleep*. 1999, roč. 22, č. 5, s. 561–572.
18. DHONGADE, Dayanand Vishwanath; RAO, TVKH. Classification of sleep disorders based on EEG signals by using feature extraction techniques with KNN classifier. In: *2017 International Conference on Innovations in Green Energy and Healthcare Technologies (IGEHT)*. 2017, s. 1–5.
19. LEONG, Wai Yie. Feature extraction and optimisation for sleep apnea. In: *2014 IEEE International Symposium on Robotics and Manufacturing Automation (ROMA)*. 2014, s. 200–205.
20. PRUCNAL, Monika A; POLAK, Adam G. Effectiveness of Sleep Apnea Detection Based on One vs. Two Symmetrical EEG Channels. In: *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2019, s. 4056–4059.
21. SABIL, AbdelKebir; GLOS, Martin; GÜNTHER, Alexandra; SCHÖBEL, Christoph; VEAUTHIER, Christian; FIETZE, Ingo; PENZEL, Thomas. Comparison of apnea detection using oronasal thermal airflow sensor, nasal pressure transducer, respiratory inductance plethysmography and tracheal sound sensor. *Journal of Clinical Sleep Medicine*. 2019, roč. 15, č. 2, s. 285–292.
22. ALMAZAYDEH, Laiali; ELLEITHY, Khaled; FAEZIPOUR, Miad; ABUSHAKRA, Ahmad. Apnea detection based on respiratory signal classification. *Procedia Computer Science*. 2013, roč. 21, s. 310–316.
23. ALVAREZ-ESTEVEZ, Diego; MORET-BONILLO, Vicente. Computer-assisted diagnosis of the sleep apnea-hypopnea syndrome: a review. *Sleep disorders*. 2015, roč. 2015.

24. ALVAREZ-ESTEVEZ, Diego; MORET-BONILLO, Vicente. Computer-assisted diagnosis of the Sleep Apnea-Hypopnea Syndrome: An overview of different approaches. In: *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2015, s. 4544–4549.
25. SADR, Nadi; CHAZAL, Philip de. Comparing Different Methods of Hand-crafted HRV, EDR and CPC Features for Sleep Apnoea Detection. In: *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2019, s. 3870–3873.
26. PATHINARUPOTHI, Rahul K; VINAYKUMAR, R; RANGAN, Ekanath; GOPALAKRISHNAN, E; SOMAN, KP. Instantaneous heart rate as a robust feature for sleep apnea severity detection using deep learning. In: *2017 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*. 2017, s. 293–296.
27. PATHINARUPOTHI, Rahul K; VINAYKUMAR, R; RANGAN, Ekanath; GOPALAKRISHNAN, E; SOMAN, KP. Instantaneous heart rate as a robust feature for sleep apnea severity detection using deep learning. In: *2017 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*. 2017, s. 293–296.
28. PATHINARUPOTHI, Rahul Krishnan; RANGAN, Ekanath Srihari; GOPALAKRISHNAN, EA; VINAYKUMAR, R; SOMAN, KP et al. Single sensor techniques for sleep apnea diagnosis using deep learning. In: *2017 IEEE international conference on healthcare informatics (ICHI)*. 2017, s. 524–529.
29. BISWAL, Siddharth; SUN, Haoqi; GOPARAJU, Balaji; WESTOVER, M Brandon; SUN, Jimeng; BIANCHI, Matt T. Expert-level sleep scoring with deep neural networks. *Journal of the American Medical Informatics Association*. 2018, roč. 25, č. 12, s. 1643–1650.
30. DEY, Debangshu; CHAUDHURI, Sayanti; MUNSHI, Sugata. Obstructive sleep apnoea detection using convolutional neural network based deep learning framework. *Biomedical engineering letters*. 2018, roč. 8, č. 1, s. 95–100.
31. HAIDAR, Rim; KOPRINSKA, Irena; JEFFRIES, Bryn. Sleep apnea event detection from nasal airflow using convolutional neural networks. In: *International Conference on Neural Information Processing*. 2017, s. 819–827.
32. HAIDAR, Rim; MCCLOSKEY, Stephen; KOPRINSKA, Irena; JEFFRIES, Bryn. Convolutional neural networks on multiple respiratory channels to detect hypopnea and obstructive apnea events. In: *2018 International Joint Conference on Neural Networks (IJCNN)*. 2018, s. 1–7.



33. CEN, Ling; YU, Zhu Liang; KLUGE, Tilmann; SER, Wee. Automatic system for obstructive sleep apnea events detection using convolutional neural network. In: *2018 40th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*. 2018, s. 3975–3978.
34. VAN ROSSUM, Guido; DRAKE, Fred L. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN 1441412697.
35. VIRTANEN, Pauli et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*. 2020, roč. 17, s. 261–272. Dostupné z DOI: <https://doi.org/10.1038/s41592-019-0686-2>.
36. HUNTER, J. D. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*. 2007, roč. 9, č. 3, s. 90–95. Dostupné z DOI: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55).
37. BRADSKI, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*. 2000.
38. MARTIN ABADI et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. Dostupné také z: <https://www.tensorflow.org/>. Software available from tensorflow.org.
39. CHOLLET, François et al. *Keras* [<https://keras.io>]. 2015.
40. BARTOŇ, Martin. *Automatická detekce fyziologických událostí ve spánku: Bakalářská práce*. ČVUT, FBMI, 2018. Dostupné také z: <https://dspace.cvut.cz/handle/10467/79696>.
41. NEUROSCIENCE TECHNOLOGY RESEARCH LTD Prague, Czech Republic. *EASYS2 Reference Manual*. 3rd edition. NTR Ltd, 2002. Č. July.
42. SULLIVAN, Lisa; LAMORTE, Wayne. *Summarizing Data: InterQuartile Range (IQR)*. Boston University. Dostupné také z: [http://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/BS704\\_SummarizingData/BS704\\_SummarizingData7.html](http://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/BS704_SummarizingData/BS704_SummarizingData7.html). InterQuartile Range (IQR), Boston University, Cit. 10.2. 2018.
43. MASKO, David; HENSMAN, Paulina. *The impact of imbalanced training data for convolutional neural networks*. 2015.
44. COHEN, Mike X. *Analyzing neural time series data: Theory and practice*. MIT, 2014. ISBN 9780262019873.
45. SMITH, Lindsay I. *A tutorial on principal components analysis*. 2002. Technická zpráva.
46. VAN DER MAATEN, L.; HINTON, G. Visualizing Data Using t-SNE. *Journal of Machine Learning Research*. 2008, roč. 9, č. 1, s. 1–48.

47. CHOI, Sang Ho; YOON, Heenam; KIM, Hyun Seok; KIM, Han Byul; KWON, Hyun Bin; OH, Sung Min; LEE, Yu Jin; PARK, Kwang Suk. Real-time apnea-hypopnea event detection during sleep by convolutional neural networks. *Computers in biology and medicine*. 2018, roč. 100, s. 123–131.
48. KULKAS, A; TIIHONEN, P; JULKUNEN, P; MERVAALA, E; TÖYRÄS, J. Desaturation delay, parameter for evaluating severity of sleep disordered breathing. In: *World Congress on Medical Physics and Biomedical Engineering May 26-31, 2012, Beijing, China*. 2013, s. 336–339.
49. SKANSI, Sandro. *Introduction to Deep Learning: from logical calculus to artificial intelligence*. Springer, 2018.
50. SRIVASTAVA, Nitish; HINTON, Geoffrey; KRIZHEVSKY, Alex; SUTSKEVER, Ilya; SALAKHUTDINOV, Ruslan. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*. 2014, roč. 15, č. 1, s. 1929–1958.
51. AGARAP, Abien Fred. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*. 2018.
52. DE BOER, Pieter-Tjerk; KROESE, Dirk P; MANNOR, Shie; RUBINSTEIN, Reuven Y. A tutorial on the cross-entropy method. *Annals of operations research*. 2005, roč. 134, č. 1, s. 19–67.
53. KINGMA, Diederik P.; BA, Jimmy. *Adam: A Method for Stochastic Optimization*. 2014. Dostupné z arXiv: <1412.6980> [cs.LG].
54. HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing; SUN, Jian. *Deep Residual Learning for Image Recognition*. 2015. Dostupné z arXiv: <1512.03385> [cs.CV].
55. COVER, Thomas; HART, Peter. Nearest neighbor pattern classification. *IEEE transactions on information theory*. 1967, roč. 13, č. 1, s. 21–27.
56. FAWCETT, Tom. An introduction to ROC analysis. *Pattern Recognition Letters*. 2006, roč. 27, č. 8, s. 861–874. ISBN 01678655. ISSN 01678655. Dostupné z DOI: <10.1016/j.patrec.2005.10.010>.
57. YOU, Yang; GITMAN, Igor; GINSBURG, Boris. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*. 2017.

## Příloha A: Seznam skriptů a funkcí

Tabulka 7.1: Seznam použitých skriptů a funkcí.

Název skriptu	Název Funkce	Popis
dfileTK.py	DFile class	Hlavní třída obsahující funkce
	__init__	Inicializace *.d souboru
	data_load	Načtení dat
	tag_info	Vypsání informací o značkách
	chan_info	Vypsání informací o kanálech
	chan_data	Uložení dat z vybraného kanálu
	istag_slide	Kontrola přítomnosti značky v segmentu
	anonymous	Anonymizace dat
	preprocess_flow_slide	Předzpracování signálu průtok
	preprocess_spo2_slide	Předzpracování signálu saturace
	save_tags	Uložení značek do *.d souboru
	tag_seg2tags_slide	Převod pozitivních segmentů do značek
	CNN_TK.py	
sensitivity		Výpočet senzitivity
specificity		Výpočet specificity
f1_score		Výpočet F1 skóre
accuracy		Výpočet přesnosti
TVT_split		Rozdělení dat na jednotlivé sady
	The_answer	Vyhodnocení detektoru
CNN_2D.py		Trénování neuronové sítě
CNN_sweep.py		Hledání ideálního parametru
Bigdata_multi.py		Příprava dat k trénování
knn.py		Výpočet metody nejbližšího souseda
AD_lunch.py		Knihovna obsahující funkce pro detektor
AD_GUI.py		Grafické rozhraní detektoru

## **Příloha B: Obsah přiloženého CD/DVD**

- Klíčová slova v ČJ
- Klíčová slova v AJ
- Abstrakt v ČJ
- Abstrakt v AV
- Zadání diplomové práce
- Diplomová práce
- Publikovaný článek
- Složka se skripty (obsahuje Python skripty a finální natrénované hluboké konvoluční neuronové sítě)

## **Příloha C: Článek publikovaný na EHB 2019**

Článek s názvem: Comparison of Linear and Non-linear Dimension Reduction Techniques for Automatic Apnea Detection, prezentovaný na IEEE International Conference on e-Health and Bioengineering, 2019.



# Comparison of Linear and Non-linear Dimension Reduction Techniques for Automatic Apnea Detection

Martin Bartoň<sup>1</sup>, Vlastimil Koudelka<sup>2</sup>, Václava Piorecká<sup>1,2</sup>, Jitka Bušková<sup>2</sup>, Vladimír Krajča<sup>1</sup>

*Affiliation 1: Department of Biomedical Technology, Faculty of Biomedical Engineering,  
Czech Technical University in Prague, martin.barton@fbmi.cvut.cz*

*Affiliation 2: National Institute of Mental Health, Klecany, Czech Republic*

**Abstract**—Oronasal thermal sensor signal recording within sleeping period was analyzed in sense of detection of respiratory events. The goal was to evaluate what extent an up to date non-linear dimension reduction technique increases performance of apnea classification compared to classifications based on linear dimension reduction and raw data. Based on an extensive database of recordings in apneic patients, we concluded that a non-linear approach didn't lead to better classification performance in comparison with a linear decomposition technique. We suggested, that it was due to the signal amplitude is naturally the main feature of respiratory events. However, nonlinear methods do not naturally maintain the amplitude based structure in data. Due to this fact we get the worse signal representation in comparison with linear methods.

**Keywords**—sleep apnea; classification; preprocessing; tSNE; PCA.

## I. INTRODUCTION

Sleep is an important part of our lives. Unfortunately, there are a lot of sleeping disorders causing sleep deprivation in population sleep apnea for example. In simple words, the sleep apnea is characterized by shallow breathing events occurring during the night, which has to be examined in a sleep laboratory by polysomnography. Subsequent expert diagnosis is very time demanding due to extensive length of the polysomnographic (PSG) recording. There is a strong motivation to search for respiratory events in an automatic mode. The goal of this paper is to compare linear and non-linear reduction dimension techniques of airflow signal for the future automatization.

The airflow signal was naturally identified as the source of the information about respiratory events. This is also supported by American Academy of Sleeping Medicine (AASM) scoring manual [1], where we can find predefined features for sleep apnea scoring based on airflow amplitude. The apnea is defined as decreasing of airflow amplitude in 90 % at least in comparison to baseline and it should have a duration 10 seconds at least.

There are numerous methods for automatic sleep apnea detection. Some of them are based on the electrocardiogram (ECG) as main signal for detection. There are different

methods to be applied, for example: Long Short-Term memory [2], Hidden Markov model [3], Normal inverse Gaussian parameters [4] and Novel algorithm [5]. Above all that studies [6], [7], [8] are based on signal combination: ECG and pulse oximetry for better results. Some studies do not use standard PSG signal at all, they are working with sound as the main source of information [9], [10]. The modern approach was selected by Ling Cen et al. [11], because they applied convolution neural network on three signals (saturation, airflow and abdominal movements). Similar method used by Choi S. et al. [12] with real-time apnea detection by convolution T neural network on single airflow signal.

None of the above mentioned methods examined an intrinsic structure of airflow signal in raw time domain when it is assumed to feed a classifier, e.g. a deep network. Since the airflow is considered to be the most important signal, our work is heading for dimensional reduction techniques and give us the answer if a non-linear approach is generally better in the signal representation compared to a linear one.

Three approaches were compared in this study: raw, linear and non-linear. The raw approach did not involve any reduction of dimensions. In the linear approach the standard Principal Component Analysis (PCA) [15] was implemented. In the non-linear approach the parametric t-distributed Stochastic Neighbor Embedding (t-SNE) method [16] was tested. The training and testing sets were created based on 477 number of PSG recordings in patients.

## II. DATABASE AND METHODS

### A. Database

The examined dataset consists of 477 patients suffering from apnea. One sample corresponds to a polysomnographic record acquired overnight at the sleep laboratory in National Institute of Mental Health in Klecany, Czech Republic (NIMH). All of the recordings were scored in sense of respiratory events by the experts from NIMH and the protocols were approved by the Ethical committee of NIMH.

The recordings were obtained on the BrainScope polysomnographic device with sampling frequency of 250 Hz. These recordings consist of 19 EEG channels, two

electrooculography channels (horizontal and vertical), one electromyography channel placed on the chin, one airflow channel, one channel for chest movements, two channels for legs movements and one SpO<sub>2</sub> channel.

A number of apnea events during one session varies strongly across subjects. The histogram of apnea events across subjects is visualized in Fig. 1. In order to create a consistent dataset a limit number of apnea events during one session per subject was calculated by the Tuckey IRQ method [13]. If the subject exceeded the limit of 559 apnea per night only the limit number of apneas was added to the final dataset, see the red line in Fig. 1. In this way, we avoided overfitting the decomposition methods to a few patients with a high number of apnea events. Finally, the training and testing sets consist of 132353 and 66177 segments respectively, see section C for more detail.

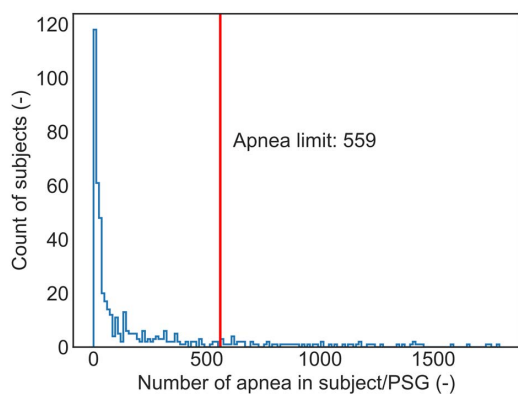


Fig. 1. Histogram of apnea count in subjects.

## B. Methods

The whole process of airflow analyses and apnea classification is described in Fig. 2. The first step is preprocessing of the airflow signals. The second step is creating raw segments, which are subsequently projected to subspaces via linear and nonlinear approaches. The last step quantifying performance of particular approach is classification.

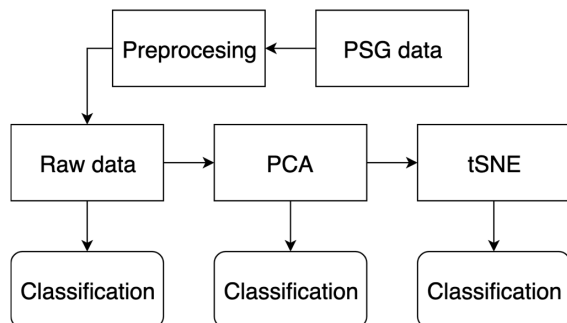


Fig. 2. The diagram of data flow process.

## C. Preprocessing

The airflow signals were downsampled to 50 Hz. The antialiasing filter was applied before downsampling [14]. The signals were divided into 5 seconds long segments. The IQR Tukey method was used to remove segments exhibiting extreme amplitudes (e.g. noise segments and segments with disconnection from PSG artefacts). Artefact-free segments were normalized by z-scores.

Finally 198530 segments were obtained. Half of them were considered segments with apnea while the other were randomly chosen segments without apnea (normal breathing was observed in this case). All segments were split into training dataset ( $\frac{2}{3}$ ) and testing dataset ( $\frac{1}{3}$ ). The partitioning was done on the subject level such that the training and testing datasets included unique subjects to test generalization of final classifiers.

The goal of PCA is to construct the set of principal components based on the covariance of a set of correlated variables. All of the principal components explain the 100 % of the signal and they are linearly uncorrelated. [15]

The PCA was used for dimension reduction in our study. The PCA transformation was fitted to training dataset and applied to testing dataset. Only the first 5 components were used, since they represented more than 95 % of the data variance.

The t-SNE is nonlinear method of dimension reduction. This method uses the local objects relationships in high-dimensional space. The t-SNE constructs the probability distribution using Gaussian distribution. The similarity between objects is defined by the conditional probability  $p_{ji}$  [16]:

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma_i^2)} \quad (1)$$

where  $\sigma$  represents the variance of the Gaussian that is centered on datapoint  $x_i$ . [16]

The transformed training data by PCA were used to train parametric t-SNE. The t-SNE was used to transformation in to two dimensions.

The  $k$ -NN ( $k$ -Nearest Neighbors) algorithm classifier method is based on supervised learning. When a new object is presented the algorithm finds his  $k$  nearest neighbors and evaluate to which class belongs to based on the neighbors classes. Due to this principle, the algorithm needs a training set of objects (etalons) for proper function. [17]

At first, classification was performed on the raw dataset. The  $k$ -NN algorithm was used for that purpose. The parameter  $k$  was set to 5 by experiment. Since the aim is to compare more dimension reduction techniques rather than searching the optimal  $k$ , the chosen value was fixed here across all three approaches. If one needs to find the best  $k$ , it is required to implement a model selection technique based on training dataset, e.g. a cross-validation approach.

Then, the same classification procedure was applied to reduced subspaces by PCA and t-SNE. We suggest, the better classification performance on the testing set, the better signal

## V. CONCLUSIONS

It is clear from our results that the linear dimension reduction technique gives almost the same results as classification on raw data. Moreover, we can see that nonlinear methods not always outperform the linear ones. This is an unexpected and interesting result which gives us important message for future development. We think that nonlinear methods distort amplitude feature and at the same time fit another unknown features.

In case of the airflow signal is considered linear method of dimension reduction are better than non-linear one. At the same time it is time saving operation rather than using raw data for classification.

We suggest that further research should be more focused on improvement in preprocessing than to fit nonlinear models to data. A meaningful improvement in dimensional reduction is seen in matching the individual breathing frequencies which have been fixed by commonly used PCA technique.

## ACKNOWLEDGMENT

This work was supported by the Grant Agency of Czech Republic reg. n. 17-20480S: Temporal context in analysis of long-term non-stationary multidimensional signal and by the Grant Agency of the Czech Technical University in Prague, reg. no. SGS18/158/OHK4/2T/17 with topic: The topographic brain mapping and by the project Nr. LO1611 with a financial support from the MEYS under the NPU I program.

## PROTECTION OF HUMAN SUBJECTS AND ANIMALS IN RESEARCH

The procedures followed were in compliance with the ethical standards of the responsible committee on human experimentation (institutional and national) and with the World Medical Association Declaration of Helsinki on Ethical Principles for Medical Research Involving Human Subjects.

## REFERENCES

- [1] Conrad, Chesson, "The AASM Manual for Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications" Westchester Illinois: AASM in U.S.A, 2007.
- [2] D. Novak, K. Mucha, Al-Ani, Tarik "Long short-term memory for apnea detection based on heart rate variability" Conference proceedings: Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society Conference 2008 5234-7 DOI: 10.1109/IEMBS.2008.4650394.
- [3] C. Song, K. Liu, X. Zhang, L. Chen, X. Xian, "An obstructive sleep apnea detection approach using a discriminative hidden Markov model from ECG Signals", 2016, IEEE Transactions on Biomedical Engineering, 63(7), 1532–1542. DIO: 10.1109/TBME.2015.2498199.
- [4] Hassan, A. Rashik, "Computer-aided obstructive sleep apnea detection using normal inverse Gaussian parameters and adaptive boosting", 2016, Biomedical Signal Processing and Control, 29, 22–30. DOI: 10.1016/j.bspc.2016.05.009.
- [5] C. Varon, A. Caicedo, D. Testelmans, B. Buyse, S. V. Huffel, "A novel algorithm for the automatic detection of sleep apnea from single-lead ECG", 2015, IEEE Transactions on Biomedical Engineering, 62(9), 2269–2278 DOI: 10.1109/TBME.2015.2422378.
- [6] L. Haoyu, L. Jianxing, N. Arunkumar, A. F. Hussein, M. M. Jaber, "An IoT cloud-based real time sleep apnea detection scheme by using the SpO2 estimation supported by heart rate variability", 2019, Future Generation Computer Systems, 98, 69–77.
- [7] H. Hilmisson, N. Lange, S. P. Duntley, "Sleep apnea detection: accuracy of using automated ECG analysis compared to manually scored polysomnography (apnea hypopnea index)", 2019, Sleep and Breathing, 23(1), 125–133. DOI: 10.1007/s11325-018-1672-0.
- [8] B. Xie, H. Minn, "Real-time sleep apnea detection by classifier combination" 2012, IEEE Transactions on Information Technology in Biomedicine, 16(3), 469–477. DOI: 10.1109/TITB.2012.2188299.
- [9] Y. Fang, Z. Jiang, H. Wang, "Research article a novel sleep respiratory rate detection method for obstructive sleep apnea based on characteristic moment waveform", 2018, Jurnal of Healthcare Engineering, DOI: 10.1155/2018/1902176.
- [10] L. Almazaydeh, K. Elleithy, M. Faezipour, A. Abushakra, "Apnea detection based on respiratory signal classification", 2013, Procedia Computer Science, 21, 310-316.
- [11] L. Cen, Z. L. Yu, T. Kluge, W. Ser, "Automatic system for obstructive sleep apnea events detection using convolutional neural network", 2018, (MI), 3975–3978, DOI: 10.1109/EMBC.2018.8513363.
- [12] S. H. Choi, H. Yoon, H. S. Kim, H. B. Kim, Kwon, H. Bin, et al., "Real-time apnea-hypopnea event detection during sleep by convolutional neural networks", 2018, Computers in Biology and Medicine, 100(February), 123–131, DOI: 10.1016/j.combiomed.2018.06.028.
- [13] L. Sullivan, W. LaMorte, "Summarizing data interquartile range (IQR)", Boston University, [http://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs\\_704\\_summarizingdata/bs704\\_summarizingdata7.html](http://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs_704_summarizingdata/bs704_summarizingdata7.html) [Online; accessed 2019-09-18].
- [14] E. Jones, E. Oliphant, P. Peterson, "SciPy: Open Source Scientific Tools for Python", <http://www.scipy.org/> [Online; acces 2019-09-19].
- [15] Mike X Cohen, "Analyzing neural time series data: Theory and practice", 2014, ISBN 9780262019873.
- [16] L. Van Der Maaten, G. Hinton, "Visualizing Data using t-SNE. Journal of Machine Learning Research", 2008. Vol. 9, p. 2579–2605.
- [17] T. M. Cover, P. E. Hart, "Nearest neighbor pattern classification." IEEE Trans, Information Theory 13 1967: 21-27.