



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

DEPARTMENT OF BIOMEDICAL ENGINEERING

INTERPRETACE UČEBNÍHO PROCESU KLASIFIKÁTORU FIBRILACE SÍNĚ

INTERPRETING THE LEARNING PROCESS OF AN ATRIAL FIBRILLATION CLASSIFIER

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Anna Lichtblauová

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Petra Novotná

BRNO 2023

Bakalářská práce

bakalářský studijní program **Biomedicínská technika a bioinformatika**

Ústav biomedicínského inženýrství

Studentka: Anna Lichtblauová

ID: 221522

Ročník: 3

Akademický rok: 2022/23

NÁZEV TÉMATU:

Interpretace učebního procesu klasifikátoru fibrilace síně

POKyny PRO VYPRACOVÁNÍ:

- 1) Nastudujte a krátce popište původ fibrilace síně, její typy a projev v signálu EKG.
- 2) Nastudujte a popište metody klasifikace a detekce fibrilace síně za využití hlubokého a strojového učení.
- 3) Seznamte se s poskytnutými daty z databáze na ÚBMI a veřejně dostupných databázích. Proveďte potřebné předzpracování dat pro následnou detekci.
- 4) Navrhněte, případně upravte, klasifikační síť a rozvrhněte varianty učícího procesu pro oddělování fibrilace síně (persistentní, paroxysmální) od jiných arytmií a sinusového rytmu. Navrhněte parametry pro porovnání jednotlivých variant sítě.
- 5) Realizované klasifikátory analyzujte a porovnejte. Optimalizujte pro použití různých délek signálů na vstupu.
- 6) Dosažené výsledky podrobně diskutujte.

DOPORUČENÁ LITERATURA:

- [1] Murat, F., Sadak, F., Yildirim, O., Talo, M., Murat, E., Karabatak, M., Demir, Y., Tan, R. S., & Acharya, U. R. (2021). Review of deep learning-based atrial fibrillation detection studies. In International Journal of Environmental Research and Public Health (Vol. 18, Issue 21). MDPI. <https://doi.org/10.3390/ijerph182111302>
- [2] Clifford GD, Liu C, Moody B, Li-wei HL, Silva I, Li Q, Johnson AE, Mark RG. AF classification from a short single lead ECG recording: The PhysioNet/computing in cardiology challenge 2017. In 2017 Computing in Cardiology (CinC) 2017 Sep 24 (pp. 1-4). IEEE. <https://doi.org/10.22489/CinC.2017.065-469>
- [3] Wang, X., Ma, C., Zhang, X., Gao, H., Clifford, G. D., & Liu, C. (2021). Paroxysmal Atrial Fibrillation Events Detection from Dynamic ECG Recordings: The 4th China Physiological Signal Challenge 2021 (version 1.0.0). PhysioNet. <https://doi.org/10.13026/ksya-qw89>.

Termín zadání: 6.2.2023

Termín odevzdání: 29.5.2023

Vedoucí práce: Ing. Petra Novotná

doc. Ing. Jana Kolářová, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

V rámci teoretické části bakalářské práce byla nastudována problematika detekce fibrilace síní (FS) a práce s konvolučními neuronovými sítěmi (CNN). Následně byly v praktické části vytvořeny dva klasifikátory. První byl určen pro klasifikaci sinusového rytmu, fibrilace síní a ostatních patologií, druhý dále rozlišoval kategorii "fibrilace síní" podle skutečnosti, zda se nacházela v celém záznamu nebo jen v jeho části. Výsledná přesnost byla 82.12 % v případě prvního, resp. 85.14 % v případě druhého klasifikátoru.

KLÍČOVÁ SLOVA

EKG, automatická detekce, fibrilace síně, strojové učení, konvoluční neuronová síť, Python

ABSTRACT

In the theoretical part of the bachelor thesis the problems of atrial fibrillation (AF) detection and principles of convolutional neural networks (CNN) are discussed. Next, two classifiers were created in the practical part. The first was designed to classify sinus rhythm, atrial fibrillation and other pathologies, while the second further distinguished the category "atrial fibrillation" according to whether it was present in the whole recording or only in a part of it. The resulting accuracies are 82.12 % and 85.14 % for the first and second classifiers, respectively.

KEYWORDS

ECG, automatic detection, atrial fibrillation, machine learning, convolutional neural network, Python

LICHTBLAU, Anna. *Interpretace učebního procesu klasifikátoru fibrilace síně*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství, 2023, 89 s. Bakalářská práce. Vedoucí práce: Ing. Petra Novotná

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Anna Lichtblau
VUT ID autora:	221522
Typ práce:	Bakalářská práce
Akademický rok:	2022/23
Téma závěrečné práce:	Interpretace učebního procesu klasifikátoru fibrilace síně

Prohlašuji, že svou závěrečnou práci jsem vypracovala samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autorka uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědoma následků porušení ustanovení § 11 a následujících autorského 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, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autorky*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Ráda bych poděkovala vedoucí mé nejprve semestrální a později bakalářské práce paní Ing. Petře Novotné za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Úvod	3
1 Fibrilace síní	5
1.1 Definice	5
1.2 Projevy v EKG	5
1.3 Léčba	6
2 Neuronové sítě	7
2.1 Perceptron	7
2.2 Hluboké neuronové sítě	9
2.2.1 Učení hlubokých neuronových sítí	9
2.2.2 Konvoluční neuronové sítě	11
2.3 Rekurentní neuronové sítě	13
2.3.1 Long-Short Term Memory	14
2.4 Metriky vyhodnocování úspěšnosti neuronové sítě	14
3 Metody detekce fibrilace síní	17
3.1 Současné přístupy pro detekci FS	17
3.2 Detekce fibrilace síně bez použití umělé inteligence	18
3.3 Detekce FS s využitím umělé inteligence	18
3.3.1 Hluboké neuronové sítě	19
3.3.2 Konvoluční neuronové sítě	19
3.3.3 Rekurentní neuronové sítě	20
3.3.4 LSTM	20
3.3.5 Hybridní modely	21
4 Popis použitých dat a jejich předzpracování	23
4.1 Možnosti rozšíření datasetu pro kategorii paroxysmů	25
4.2 Předzpracování dat	26
4.2.1 Vytvoření labelů	28
4.2.2 Úprava délky signálu	29
4.2.3 Filtrace dat	30
4.2.4 Standardizace	32
4.3 Rozdělení na trénovací a testovací množinu	32
5 Návrh a implementace neuronové sítě	35
5.1 Model pro klasifikaci do 3 kategorií	36
5.2 Model pro klasifikaci do 4 kategorií	38

5.3	Ladění parametrů klasifikátorů	38
5.3.1	Aktivační funkce	39
5.3.2	Dropout vrstvy	39
5.3.3	Pooling vrstvy	40
5.3.4	Konvoluční a plně propojené vrstvy	40
5.3.5	Epochy a batch size	41
5.3.6	Počet vzorků a optimalizace nestejně dlouhých signálů	41
5.3.7	Rozdělení testovací a trénovací množiny a rozšíření databáze	42
5.3.8	Doplňování signálu na požadovanou délku	43
5.4	Výsledné modely	43
6	Výsledky	47
7	Závěr	55
	Literatura	57
	Seznam příloh	63
A	Grafy vybraných aktivačních funkcí a jejich rovnice	65
B	Věkové rozložení datasetu z MDT	67
C	Přesnost pro vybrané aktivační funkce	69
D	Rozdíl přesnosti pro různé počty dropout vrstev	71
E	Rozdíl přesnosti pro různé parametry dropout vrstev	73
F	Přesnost pro pooling metody	75
G	Přesnost pro různé počty učících epoch	77
H	Přesnost pro různou batch size	79
I	Přesnost pro různé délky signálu	81
J	Přesnost pro různé způsoby prodloužení signálu	83
K	Přesnost pro různé rozdělení testovací a trénovací množiny	85
L	Přesnost pro různé druhy filtrace	87
M	Přesnost pro různé druhy rozšíření datasetu	89

Seznam obrázků

1.1	Ukázka fibrilace síní. Převzato z [4].	6
2.1	Schéma perceptronu, kde $X = x_1, x_2, \dots, x_n$ je vstup, w váhy, b práh/-bias, f aktivační funkce a y výstup.	8
2.2	Gradientní sestup. Převzato a upraveno z [18].	10
2.3	Architektura konvoluční neuronové sítě. Převzato a upraveno z [21].	11
2.4	Princip konvoluční vrstvy. Na vstupní obrázek o rozměrech 32x32 a hloubce 3 (barevný obrázek ve formátu RGB) je aplikován filtr o velikosti 5x5 a výsledkem je příznaková mapa o rozměrech 32x32 a hloubce 1. Převzato z [23].	12
2.5	Příznaková mapa pro jednotlivé konvoluční vrstvy. Převzato z [23].	12
2.6	Max pooling vrstva. Velikost okna je 2x2 a velikost kroku je rovna 2. Převzato a upraveno z [23].	13
2.7	Porovnání sítě s dropout vrstvou a bez dropout vrstvy převzato z [25].	13
2.8	Matice záměn, převzato z [35].	16
2.9	ROC křivka, převzato z [36].	16
3.1	Souhrn výsledků: a) Počty publikovaných článků v závislosti na roce vydání, b) Počty článků podle toho, jaký druh neuronové sítě byl použit. Převzato z [14].	17
4.1	Příklad záznamu poskytnutého MDT. Zde se konkrétně jedná o záznam se sinusovým rytmem.	24
4.2	Signál bez umělého zašumění.	27
4.3	Signál s Gaussovským šumem.	27
4.4	Signál s impulsním šumem.	27
4.5	Příklad záznamu zařazeného do kategorie PAROX.	29
4.6	Příklad záznamu zařazeného do kategorie PERM.	29
4.7	Příklad záznamu zařazeného do kategorie OTHER.	30
4.8	Příklad signálu před korekcí kolísání nulové izolinie (nahore) a signálu po filtraci kolísání nulové izolinie (dole).	31
5.1	Model pro klasifikaci do 3 kategorií (model_3).	44
5.2	Model pro klasifikaci do 4 kategorií (model_4).	45
6.1	Graf závislosti přesnosti na počtu epoch pro model_3.	48
6.2	Graf závislosti ztráty na počtu epoch pro model_3.	48
6.3	Matice záměn pro tři kategorie.	49
6.4	Graf závislosti přesnosti na počtu epoch pro model_4.	51
6.5	Graf závislosti ztráty na počtu epoch pro model_4.	51
6.6	Matice záměn pro 4 kategorie.	52

Seznam tabulek

4.1	Počty dat v jednotlivých kategoriích v původním MDT datasetu.	24
4.2	Počty dat v jednotlivých kategoriích v rozšířeném datasetu.	25
4.3	Procentuální zastoupení po jednotlivé typy patologií v neaugmentovaném datasetu.	33
4.4	Procentuální zastoupení po jednotlivé typy patologií v augmentovaném datasetu.	33
6.1	Výsledky pro klasifikátor do 3 kategorií.	50
6.2	Výsledky pro klasifikátor do 3 kategorií.	53

Úvod

V posledních letech zažívá medicína rychlé změny díky rozvoji umělé inteligence a hlubokého učení. Mezi možné aplikace patří využití konvolučních neuronových sítí (CNN) pro detekci srdečních arytmií, například právě fibrilace síní (FS). Její správná detekce a diagnostika bývají přínosné pro následnou léčbu.

Fibrilace síní je nejčastější porucha srdečního rytmu a její výskyt s rostoucím věkem dožití v populaci i nadále stoupá. Je ovšem spojena s předčasným úmrtím a dalšími zdravotními komplikacemi, jako je například vyšší pravděpodobnost vzniku cévní mozkové příhody. Nejspolehlivější metodou záchytu onemocnění jak u symptomatických, tak asymptomatických jedinců, je pomocí elektrokardiogramu (EKG). Nejčastěji se přistupuje k dlouhodobější monitoraci pomocí holter EKG, která probíhá v rádech dnů až měsíců. Hodnocení těchto záznamů je poté zdlouhavou a rutinní činností. Lidé mají při této činnosti tendence chybovat, ztrácet pozornost a přehlížet nepatrné detaily. Zvláště v poslední době lze ovšem tato data hodnotit i pomocí umělé inteligence, kdy lidský expert pouze kontroluje a validuje patologie nalezené pomocí této metody.

Tato bakalářská práce se zabývá návrhem a implementací klasifikátorů srdečních patologií do 3 a 4 kategorií, které budou popsány dále. První by měl být schopen rozlišovat sinusový rytmus, fibrilaci síní a ostatní srdeční patologie. Druhý by měl být schopen rozlišovat sinusový rytmus, fibrilace s koncem nebo začátkem v zadaném signálu, fibrilaci v celém signálu a ostatní patologie. Jako ostatní patologie můžou být jmenovány například flutter síní, bradykardie nebo různé typy blokády Tawarova raménka.

Zbytek práce je členěn následovně. V teoretické části v Kapitole 1 je popsán úvod do fibrilace síní, její typy a léčba. V následující Kapitole 2 jsou popsány základy neuronových sítí s důrazem na konvoluční neuronové sítě, které jsou použité v této práci. Jsou zde také popsány hodnotící metriky těchto sítí. Následně jsou v Kapitole 3 popsány způsoby využívané k detekci fibrilace síní jak s využitím umělé inteligence, tak bez něj.

V praktické části jsou v Kapitole 4 popsána použitá data a jejich základní statistické charakteristiky. V Kapitole 5 je popsán samotný proces implementace obou neuronových sítí a jejich popis z pohledu struktury. V této kapitole je také popsán proces ladění parametrů těchto sítí a jejich výsledné hodnoty. V Kapitole 6 jsou diskutovány dosažené výsledky těchto klasifikátorů s využitím výše zmíněných metrik. V Závěru 7 jsou poté shrnuty výsledky této práce a navržena některá možná vylepšení implementovaných modelů.

1 Fibrilace síní

V této kapitole bude vysvětlena definice fibrilace síní, její rozdělení, projevy v EKG a její léčba.

Fibrilace síní (dále jen FS) je nejrozšířenější poruchou srdečního rytmu, která významně ovlivňuje kvalitu života pacientů. Nejčastěji se vyskytuje u starších osob, kdy je prevalence 5-15 %. V celkové populaci se jedná o 1-2 %, přičemž muži jsou postiženi častěji než ženy [1] [2].

1.1 Definice

Jedná se o supraventrikulární arytmii, pro kterou jsou charakteristické rychlé nepravidelné stahy síní. Může se vyskytovat samostatně nebo s jinými supraventrikulárními arytmiemi, nejčastěji však s flutterem síní, kdy mezi sebou mohou obě arytmie přecházet nebo se střídát. Příčina může být vyvolána organickým onemocněním srdce (vady chlopní, kardiomyopatie, vrozené srdeční vady, ischemická choroba srdeční), onemocněním štítné žlázy, ale i užíváním alkoholu nebo některých druhů léků. Obtíže mohou být asymptomatické, nebo se mohou vyskytnout palpitace, stenokardie, dušnost, únava nebo závrať [1] [3].

FS se dělí na čtyři základní typy [1]:

- První dokumentovaná ataka – může být symptomatická nebo asymptomatická, může se objevit izolovaně, ale také v budoucnu recidivovat.
- Paroxysmální – epizoda spontánně končí a navazuje na ni sinusový rytmus, většinou do 48 hodin.
- Perzistující – epizoda trvá 6-7 dní, případně může být přerušena kardioverzí.
- Permanentní – přetrvává i po kardioverzi a nereaguje na léčbu antiarytmiky.

1.2 Projevy v EKG

Srdeční frekvence u zdravého jedince je 60-90 tepů za minutu, u FS je typická vysoká frekvence tvorby vzruchů v síni, často 300-600 tepů za minutu. Srdeční vzruchy vznikají nekoordinovaně v síňovém myokardu a způsobují rušení fyziologického vzruchu vedeného z SA uzlu, což se na EKG zobrazí jako nepravidelné síňové tzv. vlnky f, které mají nízkou amplitudu a vysokou frekvenci. Signál je pak dále převeden normální cestou přes AV uzel a Hissův svazek na komory. V AV uzlu jsou však některé vzruchy blokovány nebo převedeny jen částečně, což způsobí nepravidelnou aktivitu komor. Frekvence komor bývá normální, okolo 80-100 stahů za minutu [1].

Typickým nálezem na EKG bývá absence vlny P ve všech svodech a nepravidelný RR interval. Komplexy QRS bývají normálního tvaru s výjimkou případů, kdy je



Obr. 1.1: Ukázka fibrilace síní. Převzato z [4].

s FS diagnostikována i blokáda Tawarových ramének nebo Wolffův-Parkinsonův-Whiteův syndrom [5].

Na Obrázku 1.1 je zobrazen svod II, ve kterém jsou vidět typické vlnky f s absencí vlny P a nepravidelný RR interval s normálním tvarem QRS.

1.3 Léčba

Léčba záleží na typu onemocnění, jeho trvání, projevech a rizicích pro pacienta. Pokud je FS projevem jiné nemoci, je nutné léčit toto primární onemocnění, v ostatních případech je léčba založena na snaze o upravení a následné udržení srdečního rytmu. Nejprve je běžně zvolena léčba pomocí antiarytmik a pokud selže, tak se přistoupí ke kardioverzi. Před jejím samotným provedením je potřeba zamezit krevním sraženinám, aby se následně nedostaly do krevního oběhu, protože pak může dojít k tvorbě trombů a systémové embolizaci, v nejhorších případech i cévní mozkové příhodě. FS je proto indikací k podávání antikoagulačních látek [6].

2 Neuronové sítě

V této kapitole budou popsány základní poznatky týkající se neuronových sítí. Nejprve bude popsán perceptron a jeho funkce, následně také hluboké neuronové sítě včetně jejich nejrozšířenějších typů. Na konci této kapitoly jsou zmíněny metriky používané pro vyhodnocení úspěšnosti neuronových sítí.

Neuronové sítě byly inspirovány biologickými neurony v nervové soustavě živých organismů a jejich schopností učit se, zobecňovat a interpretovat závislosti obsahující neúplné či chybné informace. Jsou vhodné pro řešení problémů, kde klasické programování často selhává, například klasifikace objektů na obrázku, rozpoznávání rukopisu nebo predikce různých jevů, např. sekundární struktury bílkovin [7], [8].

Ve 40. letech 20. století vytvořili W. McCulloch a W. Pitts první matematický model biologického neuronu, nicméně nepředpokládali, že by model byl prakticky využíván. O něco později byla D. Hebbem představena teorie založená na podmíněných reflexech, která uvažovala o učení neuronu pomocí změny vah jeho vstupů - tedy při průchodu korektního signálu konkrétní nervovou drahou je reflex posílen. V roce 1957 byl F. Rosenblattem vynalezen perceptron, který vznikl úpravou McCulloch-Pittsova neuronu [9].

2.1 Perceptron

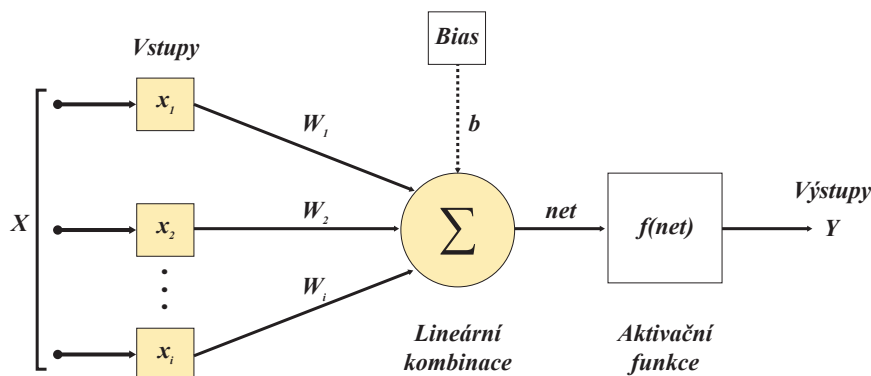
Perceptron je nejjednodušší model neuronu používaný pro řešení lineárně separovatelných problémů do dvou tříd. Obrázek 2.1 ilustruje základní schéma perceptronu. Vztah mezi vstupem $X = x_1, x_2, \dots, x_n$ a zpravidla jedním výstupem y je dán rovnicí 2.1, jejíž proměnné budou vysvětleny níže.

$$y = f\left(\sum_{i=1}^n (w_i x_i) + b\right) \quad (2.1)$$

Činnost perceptronu je následující. Na vstup neuronu je přiveden vstupní vektor $X = x_1, x_2, \dots, x_n$, jehož hodnoty jsou v první iteraci vynásobeny s náhodnými hodnotami vah $W = w_1, w_2, \dots, w_n$ a v těle neuronu sečteny. Následně je odečten práh/bias b . Tato hodnota se stane argumentem pro aktivační funkci f , která pomocí lineární či nelineární transformace převede argument na výstupní hodnotu y . Mezi nejpoužívanější aktivační funkce patří například sigmoida nebo ReLu (viz Příloha A). Perceptron tedy provádí zobrazení popsané rovnicí 2.2.

$$f : \mathbb{R}^n \rightarrow \mathbb{R}^m \quad (2.2)$$

V rovnici 2.2 je n velikost vstupního vektoru $X = x_1, x_2, \dots, x_n$ a m je velikost výstupu y . \mathbb{R} zde značí množinu reálných čísel.



Obr. 2.1: Schéma perceptronu, kde $X = x_1, x_2, \dots, x_n$ je vstup, w váhy, b práh/bias, f aktivační funkce a y výstup.

Perceptron je schopen řešit pouze lineárně separovatelné problémy, není tedy schopen např. realizovat logickou funkci XOR. Pro překonání tohoto problému bylo navrženo použít více vzájemně propojených umělých neuronů, čímž vznikla kategorie umělých neuronových sítí (ANN). Sítě měly původně jednu vrstvu neuronů (výstup neuronů byl zároveň výstupem celé sítě). Dalším vývojovým krokem bylo připojit výstupy jedné vrstvy jako vstupy další, čímž vznikly vícevrstvé neuronové sítě.

Obecně se dá neuronová síť popsat takto: jednotlivé vrstvy neuronů transformují vstupní vektor pomocí nelineární přenosové funkce a poté je jejich výstup předán jako vstup další vrstvě. První vrstva ANN se nazývá vstupní a přijímá data, která mají být zpracována. Poté většinou následují skryté vrstvy, které provádí optimalizaci vah a nelineární transformace vstupních dat. Poslední vrstva se nazývá výstupní a vytváří výstupy sítě, které jsou výsledkem zpracování všemi předchozími vrstvami [10] [11].

Dříve měly ANN pouze vstupní a výstupní vrstvy. Příkladem může být SOM, samoorganizující se mapa, občas také Kohonenova síť [12]. SOM má jedno- nebo dvouvrstvou architekturu, kde vstupní (případně jediná) vrstva neuronů je uspořádána do obdélníkového, čtvercového či hexagonálního tvaru. Princip učení je založen na přizpůsobení se sítě konkrétnímu vstupu a určení neuronu a jeho okolí, které si navzájem co nejvíce odpovídají. Pak jsou upraveny váhy vybraného neuronu tak, aby co nejvíce odpovídaly hodnotám vstupu. Takto jsou sítě předkládány další a další vstupy, dokud nevzniknou shluky reprezentující roztríděná data. Vzhledem ke způsobu samoadaptace se jedná o učení bez učitele [13].

Teprve později vznikly mělké neuronové sítě obsahující právě jednu skrytou vrstvu¹ a pak se zvyšujícím se výkonem počítačů také hluboké neuronové sítě, které obsahují více skrytých vrstev.

2.2 Hluboké neuronové sítě

Koncept hlubokých neuronových sítí (*deep neural networks*, DNN) pochází již ze 60. let 20. století, ale teprve v posledních 15 letech postoupily výpočetní možnosti počítačů na takovou úroveň, aby bylo možné trénovat dostatečně rozsáhlé DNN.

Hluboké neuronové sítě se obvykle skládají ze vstupní a výstupní vrstvy a nejméně dvou skrytých vrstev. Toto uspořádání umožňuje extrakci komplexních příznaků ze vstupních dat, nicméně se vzrůstajícím počtem vrstev roste i výpočetní náročnost sítě.

2.2.1 Učení hlubokých neuronových sítí

Učení neuronové sítě znamená obecně optimalizaci vah a prahů podle zvoleného algoritmu učení, tak aby rozdíl mezi požadovaným a skutečným výstupem byl při použití trénovací množiny minimální.

Průběh učení neuronové sítě lze sledovat pomocí chybové funkce. Existují různé druhy chybových funkcí, jako například součet kvadratických chyb (*sum squared error* – SSE, rovnice 2.3) nebo střední kvadratická chyba (*mean squared error* – MSE, rovnice 2.4).

$$SSE = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.3)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.4)$$

kde n je počet vstupů, y_i je aktuální vstup a \hat{y}_i je predikovaná hodnota výstupu [15] [16] [17]. Čím nižší hodnota chybové funkce, tím je neuronová síť přesnější.

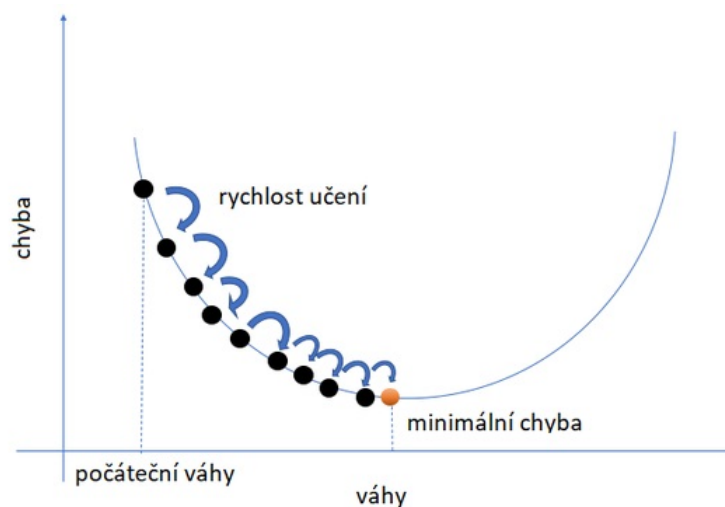
Nejčastěji používaným algoritmem pro učení vícevrstvých neuronových sítí je algoritmus zpětného šíření chyby (*backpropagation*). Obecným principem tohoto algoritmu je minimalizace chybové funkce, která jako argumenty bere všechny váhy sítě. Minimalizace poté probíhá metodou gradientního sestupu (viz níže). Činnost algoritmu je následující.

Nejprve se signál šíří dopředu od vstupní do výstupní vrstvy. Následně je z reálného a očekávaného výstupu vypočtena chybová funkce a její parciální derivace pro jednotlivé váhy. Z těchto parciálních derivací se následně určí vektor nejrychlejšího

¹Skryté vrstvy jsou takové vrstvy, které nejsou vstupní ani výstupní.

sestupu (tedy minimalizace chyby) a váhy se následně změny o určitou velikost ve směru tohoto vektoru. Tomuto postupu se říká gradientní sestup. Je však třeba zvolit správnou velikost kroku, protože při použití příliš velkého kroku funkce nemusí konvergovat. Moc malá velikost kroku oproti tomu prodlužuje čas učení a zvyšuje jeho výpočetní nároky. Chyba je postupně zpětně šířena do předchozích vrstev a postup se opakuje pro další vrstvy.

Poté je na vstup přiveden další vektor hodnot a celý proces se opakuje, dokud není minimalizován rozdíl mezi skutečnou a požadovanou hodnotou pro všechny kategorie klasifikátoru. Tímto procesem je získána natrénovaná síť [15] [16] [17].



Obr. 2.2: Gradientní sestup. Převzato a upraveno z [18].

Existují tři základní typy učení neuronových sítí: učení s učitelem, učení bez učitele a zpětnovazebné.

Učení s učitelem je nejpoužívanějším přístupem. Data jsou rozdělena na testovací a trénovací množinu. U trénovací množiny jsou člověkem k datům přiřazeny správné výstupy (anotace dat), což se dělá často ručně nebo částečně automaticky, takže je to největší nevýhoda tohoto přístupu. Cílem je dosáhnout algoritmu, který bude díky schopnosti generalizace schopen předpovědět správný výstup i pro testovací množinu, tedy pro data, která síť dosud nebyla předložena. Používá se u regresních a klasifikačních úloh.

Učení bez učitele má pouze vstupní data bez označených výstupních hodnot. Síť se snaží najít vztah mezi daty nebo je třídit podle podobných znaků. Často nemá předem požadovaný výstup. Jedná se o metodu využívanou zejména u shlukování.

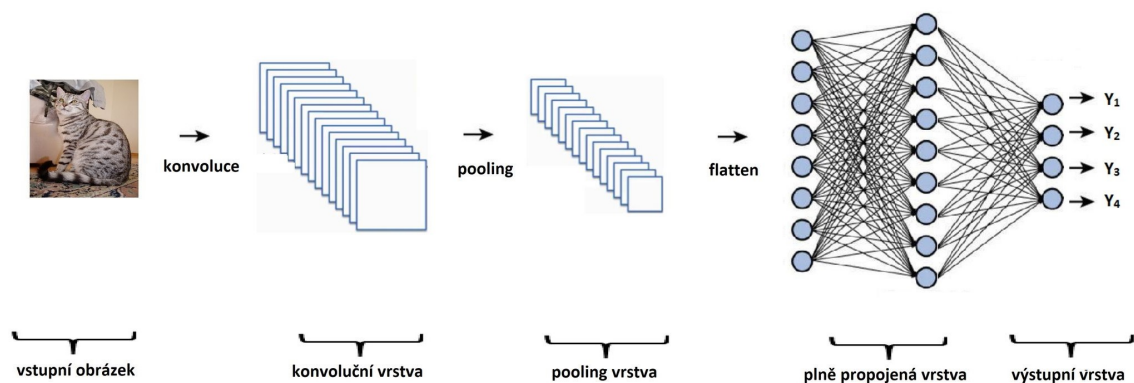
Zpětnovazebné učení (*reinforcement learning*) umožňuje najít nejvhodnější strategii pomocí maximalizace celkové odměny v daném prostředí. Algoritmus se učí z dílčích výsledků (interakce s člověkem nebo předmětem) a rozhoduje se, jakou

akci provede příště na základě zpětné vazby z předchozí akce. Používá se zejména v robotice a v počítačových hrách [19].

2.2.2 Konvoluční neuronové sítě

Konvoluční neuronové sítě jsou nejčastěji používány ke zpracování obrazových dat. Jejich největší předností je rozpoznávání vzorů v celé ploše obrazového vstupu od těch jednodušších, jako jsou hrany, až po ty složitější, jako jsou například části lidské tváře. Schopnost rozpoznat tyto vzory stoupá s hloubkou sítě.

CNN jsou složeny ze čtyř druhů vrstev: konvoluční, pooling, dropout a plně propojené (*fully-connected*). Architekturu CNN ilustruje Obrázek 2.3.



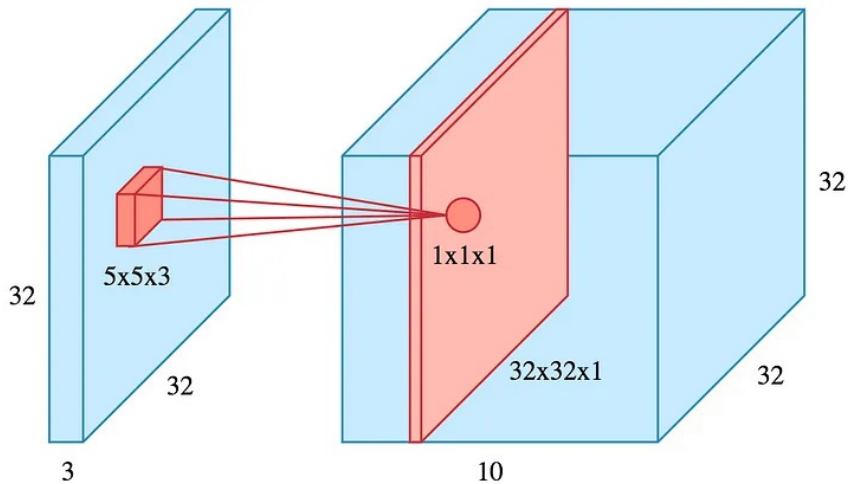
Obr. 2.3: Architektura konvoluční neuronové sítě. Převzato a upraveno z [21].

Jak již název napovídá, v konvolučních vrstvách se uplatňuje matematická operace konvoluce.

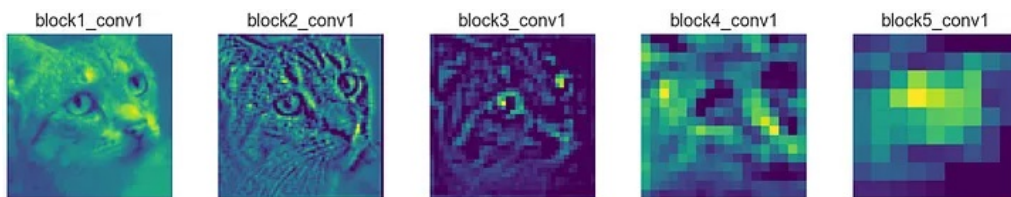
Tato operace zpracovává dvě funkce. Ve zpracování obrazu se jedná o obrázek a filtr, jinak zvaný také jako kernel nebo maska. Tento filtr má stejnou hloubku, ale (výrazně) menší velikost než vstupní obrázek. Filtr se posouvá po vstupním obrázku s předem zadaným krokem (standardně 1) a vypočítá skalární součin filtru a odpovídající části obrázku, který je zanesen do příznakové mapy s hloubkou jedna. Krokem filtru jde regulovat velikost výstupu, kdy velký krok zmenšuje velikost příznakové mapy a snižuje tím výpočetní náročnost, ale zároveň hrozí, že nebudou zaznamenány detailní vzory ze vstupního obrázku. Celková hloubka výstupu závisí na počtu použitých filtrů. Jedna dimenze výstupu se poté nazývá příznaková mapa (*feature map*) 2.5. Průběh konvoluce je znázorněn na Obrázku 2.4.

Po konvoluční vrstvě obvykle následuje pooling vrstva, která má za úkol zmenšit velikost příznakové mapy. Obvykle je zachována hloubka, ale zmenšena rozměry výstupu. Nejčastějším typem je *max pooling*, kdy je z předem daného okna vybrána nejvyšší hodnota a ostatní zahozeny (Obrázek 2.6). Je tím dosaženo zachování pouze

relevantních informací a tím zrychlení procesu učení. Další možností je *average pooling*, kdy je z oblasti spočítána průměrná hodnota.



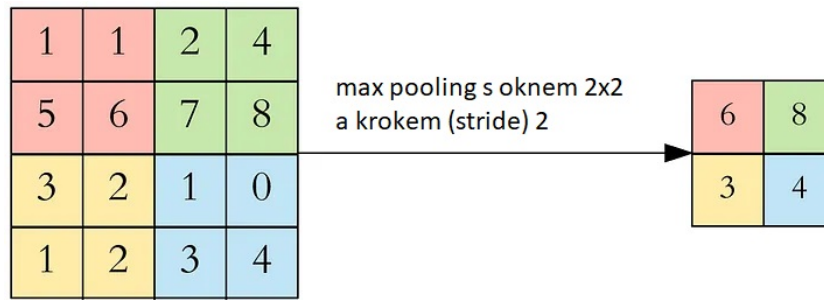
Obr. 2.4: Princip konvoluční vrstvy. Na vstupní obrázek o rozměrech 32x32 a hloubce 3 (barevný obrázek ve formátu RGB) je aplikován filtr o velikosti 5x5 a výsledkem je příznaková mapa o rozměrech 32x32 a hloubce 1. Převzato z [23].



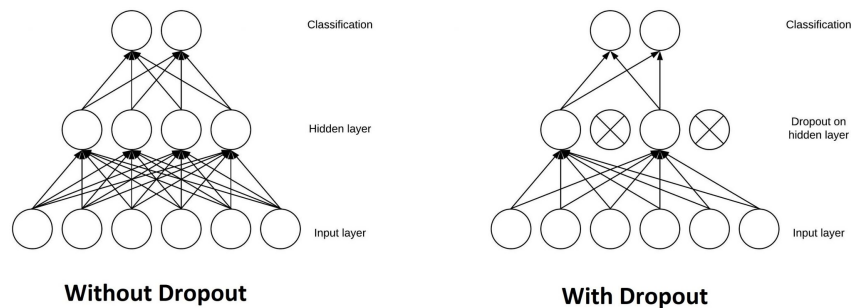
Obr. 2.5: Příznaková mapa pro jednotlivé konvoluční vrstvy. Převzato z [23].

Konvoluční a pooling vrstvy se postupně střídají, zmenšují velikost a zvětšují hloubku obrázku, čímž dochází k extrakci různých vzorů ze vstupního obrázku. Na závěr je obrázek redukován do 1D vektoru, který slouží jako vstup pro plně propojenou vrstvu. Tato vrstva slouží jako klasifikátor vyextrahovaných vzorů z předchozích vrstev a určí pomocí aktivační funkce pravděpodobnost příslušnosti k předem definovaným skupinám [22] [23].

Na vstupních nebo skrytých vrstvách může být ještě umístěna tzv. dropout vrstva, která se používá pouze při učení sítě, kde slouží proti jejímu přetrénování. V každé iteraci je dočasně „odstaven“ určitý počet neuronů v závislosti na tzv. *dropout rate*. Tento parametr bývá nastaven obvykle okolo hodnoty 0.5 - tedy 50 % neuronů bude náhodně „odstaveno“ a jejich váhy se nebudou měnit [24]. Ilustrace činnosti dropout vrstvy je na Obrázku 2.7.



Obr. 2.6: Max pooling vrstva. Velikost okna je 2x2 a velikost kroku je rovna 2. Převzato a upraveno z [23].



Obr. 2.7: Porovnání sítě s dropout vrstvou a bez dropout vrstvy převzato z [25].

Vstupní data ve formě obrázku jsou reprezentována jako 3D matice s parametry výška, šířka a hloubka, kdy hloubka odpovídá počtu barevných kanálů, tedy například pro barevný RGB obrázek je hloubka 3. EKG signál může být reprezentován jako 2D matice, kdy jedna dimenze je délka signálu a druhý hloubka, která je daná počtem EKG svodů [26].

Rozdíl mezi 1D CNN a 2D CNN je kromě konkrétní aplikace také v parametrech filtru a ve výstupu. Pro 1D se filtr pohybuje jen po jedné ose a jeho výstup je 1D vektor. Pro 2D se filtr pohybuje ve 2 směrech (x, y) a výstupem je 2D matice [27].

2.3 Rekurentní neuronové sítě

Rekurentní neuronové sítě (*recurrent neural networks*, RNN) se od předchozích sítí liší skutečností, že výstupy jedné vrstvy mohou být přivedeny na vstupy předchozí vrstvy, jedná se tedy o cyklické neuronové sítě.

RNN dokáží zachovávat informaci o sekvenčnosti dat, čímž jsou vhodné pro zpracování přirozeného jazyka, časové predikce nebo právě analýzu EKG signálu, i když v praxi se používají spíše sítě konvoluční [14]. RNN, na rozdíl od dopředných neuronových sítí, mají výstup jedné vrstvy přiveden zpět na vstup buď té samé, nebo

některé z předcházejících vrstev. Toto zapojení jim umožňuje uchovávat si určitý vnitřní stav. Když jsou tedy na vstup přivedena nová data, neuronová vrstva má k dispozici v nějaké podobě i předcházející data. Výsledná hodnota je vypočítána jako vážený součet aktuálního a předchozího vstupu. Stejně jako jiné zmiňované ANN, i RNN používá algoritmus zpětné propagace chyby, nicméně hlavní nevýhodou RNN je tzv. mizející gradient (*vanishing gradient*), kdy při jejím učení dochází ke zmenšení gradientů téměř k nulové hodnotě a přetečení či podtečení u čísel s konečnou přesností, čímž se váhy nezmění na optimální hodnotu a nedojde k nalezení minima u chybové funkce [28]. Kromě mizejícího gradientu se lze u RNN setkat i s tzv. explodujícím gradientem (*exploding gradient*), kdy je problém téměř totožný, jen s tím rozdílem, že váhy jsou násobeny hodnotou větší než 1 a tím dojde k tomu, že gradientní sestup bude divergovat. Oba tyto problémy lze řešit pomocí tzv. *gradient clipping* (jednoduše řečeno, omezení hodnoty gradientu na určitý interval).

2.3.1 Long-Short Term Memory

Long-Short Term Memory (LSTM) sítě jsou druhem RNN. Obsahují zpětnovazebné paměťové buňky, které dokáží uchovávat předchozí váhy mnohem delší dobu a pomocí toho se učit efektivněji. Oproti klasickým RNN jsou tedy vhodnější pro zpracování delších sekvencí. Jejich výhodou je, že u nich nedochází k mizejícímu ani explodujícímu gradientu [30]. Tato architektura je využívána zejména pro zpracování hudby, řeči nebo predikci slov u „chytrých“ klávesnic.

2.4 Metriky vyhodnocování úspěšnosti neuronové sítě

K vyhodnocení úspěšnosti ANN je třeba použít vhodnou metriku podle řešeného problému. Existují různé metriky pro úlohu klasifikace a regrese. Některé z těchto metrik budou popsány níže.

Matice záměn (Obrázek 2.8) je standardní způsob vyhodnocování binárních klasifikačních úloh, přičemž může být rozšířena i pro více kategorií. Jedná se o matici, která má jako řádky skutečné a jako sloupce predikované hodnoty. Uvnitř matice jsou hodnoty *true positive* (TP), *false positive* (FP), *true negative* (TN) a *false negative* (FN), pomocí kterých je možné dopočítat další metriky (přesnost, F1-skore, ...) a získat tak další pohledy na popis modelu [21] [34].

- TP: Jev je predikován jako pozitivní a ve skutečnosti je pozitivní.
- FP: Jev je predikován jako pozitivní a ve skutečnosti je negativní.
- TN: Jev je predikován jako negativní a ve skutečnosti je negativní.
- FN: Jev je predikován jako negativní a ve skutečnosti je pozitivní.

Senzitivita (Rovnice 2.5), nebo také *recall*, udává informaci o tom, jak dobře model detekuje skutečně pozitivní vzorky. Tato hodnota by měla být co nejvyšší [21][34].

$$\text{Senzitivita} = \frac{TP}{TP + FN} \quad (2.5)$$

Specifita (Rovnice 2.6) udává informaci o tom, jak správně jsou detekovány skutečně negativní vzorky [21] [34].

$$\text{Specifita} = \frac{TN}{TN + FN} \quad (2.6)$$

Přesnost (*accuracy*, Rovnice 2.7) ukazuje celkovou úspěšnost modelu a určuje poměr mezi všemi jevy celkem a predikcemi, které byly správné. Hodí se zejména na datasey s rovnoměrně zastoupenými daty [21] [34].

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN} \quad (2.7)$$

Preciznost (*precision*, Rovnice 2.8) nebo též *positive predictive value* (PPV) určuje kolik pozitivně detekovaných jevů je opravdu skutečně pozitivních. Tato hodnota by také měla být co nejvyšší [21][34].

$$\text{PPV} = \frac{TP}{TP + FP} \quad (2.8)$$

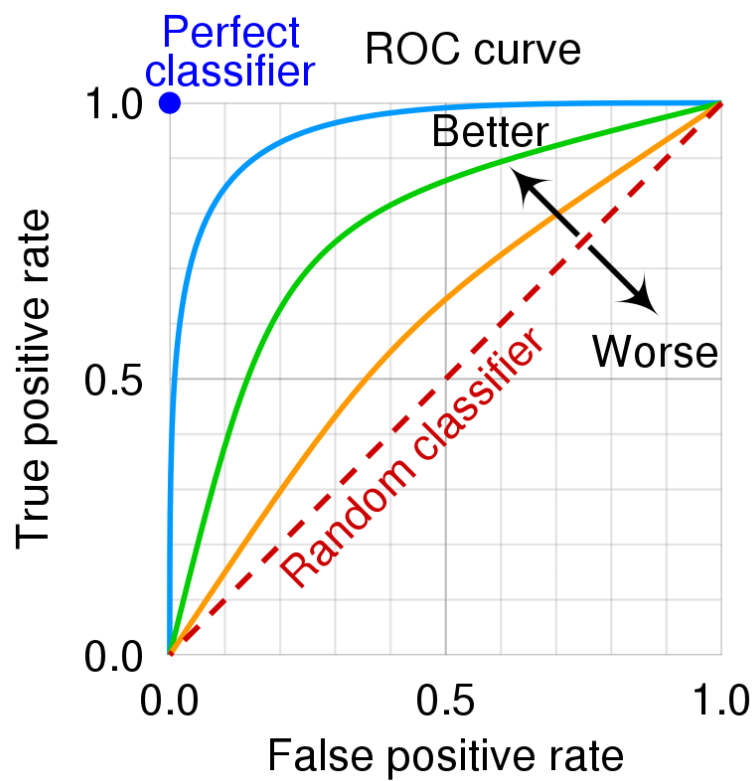
F1-skore (Rovnice 2.9) vyjadřuje harmonický průměr preciznosti a senzitivity, kdy matematicky se jedná o vážený průměr. Hodnota F1-skore se pohybuje od 0 po 1, kde 0 je nejhorší a nejlepší. Oproti přesnosti je vhodnější pro datasey s nerovnoměrným zastoupením dat. Umožňuje porovnávat různé modely s rozdílnou senzitivitou a precizností [21][34].

$$\text{F1-skore} = \frac{2 \cdot \text{PPV} \cdot \text{Se}}{\text{PPV} + \text{Se}} \quad (2.9)$$

ROC křivka (*receiver operating characteristic curve*) je grafické znázornění kvality klasifikátoru v závislosti na nastavení jeho klasifikačního prahu. Na svislé ose je senzitivita a na vodorovné ose je specifita. Nejlepší ROC křivka má největší plochu pod křivkou (*AUC – area under curve*). Pokud je velikost AUC rovna 1 (tmavě modrý bod vlevo nahoře), pak se jedná o ideální klasifikátor a má 100% specifitu a senzitivitu, avšak v praxi to budou tyto hodnoty vždy o něco nižší (světle modrá, zelená a žlutá křivka). Nejhorší možností je náhodný klasifikátor znázorněný červenou přerušovanou čarou (Obrázek 2.9) [21][34].

		Prediction	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Obr. 2.8: Matice záměn, převzato z [35].



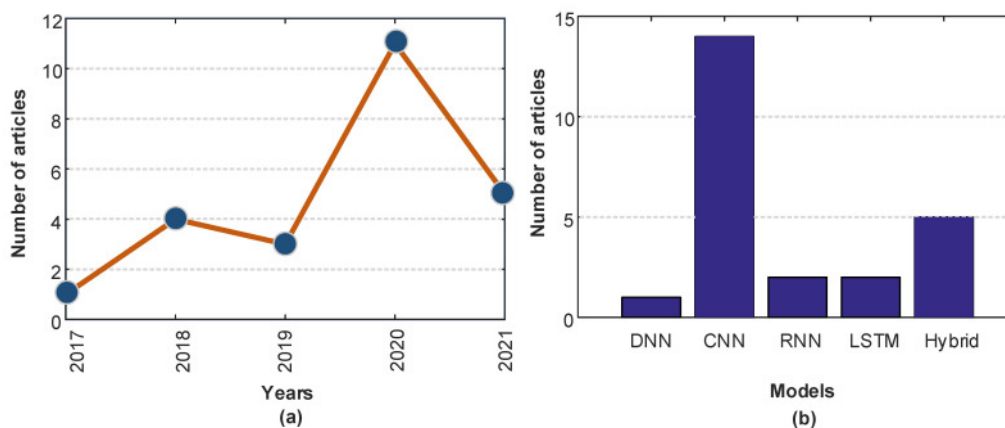
Obr. 2.9: ROC křivka, převzato z [36].

3 Metody detekce fibrilace síní

V této kapitole budou popsány metody fibrilace síní. Nejprve budou diskutovány současné přístupy k detekci FS a následně popsány jednotlivé metody, nejprve bez použití umělé inteligence a následně i s jejím využitím.

3.1 Současné přístupy pro detekci FS

Článek [14] z října 2021 byl zaměřen na vyhledávání článků obsahující slovní spojení týkající se FS nebo detekce srdečních arytmií a modelů hlubokého strojového učení v letech 2016 až 2021. Vyhledání bylo provedeno v databázích Google Scholar¹, Mendeley² a na ScienceDirect³. Celkově bylo vybráno 24 článků, z nichž bylo nejvíce publikováno v roce 2020. Typy nejčastěji používaných neuronových sítí byly *Deep Neural Networks* (DNN), *Convolutional Neural Networks* (CNN), *Recurrent Neural Networks* (RNN), *Long Short-Term Memory* (LSTM) a hybridní architektury, kdy CNN byly zastoupeny nejvíce [14]. Podrobnější souhrn výsledků je zobrazen na Obrázku 3.1.



Obr. 3.1: Souhrn výsledků: a) Počty publikovaných článků v závislosti na roce vydání, b) Počty článků podle toho, jaký druh neuronové sítě byl použit. Převzato z [14].

¹<https://scholar.google.com/>

²<https://www.mendeley.com/>

³<https://www.sciencedirect.com/>

3.2 Detekce fibrilace síně bez použití umělé inteligence

V této kapitole budou stručně představeny některé metody pro detekci FS bez použití umělé inteligence. Ve většině případů tyto metody vychází z abnormalit viditelných na křivce EKG. Jak již bylo zmíněno v Kapitole 1, pro FS je typický nepravidelný RR interval a chybějící či abnormální vlna P. Všechny uvedené metody využívají předchozí detekci R kmitu a zaměřují se na nepravidelnost RR intervalů.

První možností je detekce pomocí variačního koeficientu. Jedná se o velmi jednoduchý přístup. Výpočet (Rovnice 3.1) variačního koeficientu je dán podílem směrodatné odchylky všech RR intervalů (RR_{sorr}) a průměru hodnot těchto RR intervalů (RR_p). Pokud je hodnota variačního koeficientu překročena, je úsek označen jako FS [32].

$$VK = \frac{RR_{sorr}}{RR_p} \quad (3.1)$$

Detekce pomocí střední kvadratické odchylky postupných rozdílů je dalším poměrně jednoduchým přístupem. Jeho základem je rovnice 3.2, kde RR značí jeden RR interval a N je počet RR intervalů v tomto úseku. Obdobně jako u variačního koeficientu, pokud je hodnota střední kvadratické odchylky postupných rozdílů překročena, je úsek označen jako FS [32].

$$RMSSD = \sqrt{\frac{1}{N-1} \left(\sum_{i=1}^{N-1} (RR_{i+1} - RR_i)^2 \right)} \quad (3.2)$$

Poslední zmíněný způsob je detekce pomocí Poincarého map. Do mapy je vykreslena závislost délky RR intervalu na délce předchozího RR intervalu. Rozdíly by měly být zjevné na první pohled, kdy pro zdravého jedince s fyziologickým EKG se budou body nacházet zejména podél diagonály a pro jedince s FS budou rozptýleny po celé ploše mapy [33].

3.3 Detekce FS s využitím umělé inteligence

Jak již bylo zmíněno v úvodu této kapitoly, umělá inteligence je pro detekci FS hojně využívána. Proces zpracování EKG se obecně skládá z předzpracování vstupních dat, zejména z filtrování šumu a standardizace, dále ze zvolení typu vstupních dat (např. jeden či více svodů), zvolení modelu hlubokého učení a nakonec klasifikace výstupu. Níže budou popsány nejčastěji používané architektury neuronových sítí pro detekci FS. Dělení převzato z [14].

3.3.1 Hluboké neuronové sítě

Hluboké neuronové sítě (DNN) jsou takové vícevrstvé neuronové sítě, které obsahují velké množství skrytých vrstev. Jejich výhodou oproti ostatním zmiňovaným architekturám je jejich rychlost [14]. V článku [14] je zmiňován model, který je schopný detekovat FS při klasifikaci na dvě (FS a ostatní) a tři (FS, SR a ostatní) kategorie. Jako vstupní data používá nezpracované 12svodové EKG o délce 10 s. Síť se skládá ze 4 bloků hustě propojených konvolučních vrstev s různou kombinací filtrů a také využívá jednotku *squeeze-and-excitation*, která zlepšuje rozpoznávací schopnosti. Pracuje na principu přidání váhového parametru do každého kanálu konvolučního bloku tak, aby síť mohla adaptivně upravit váhu výsledné příznakové mapy [20]. Výsledná přesnost sítě byla $99.35 \pm 0.26 \%$, senzitivita $99.19 \pm 0.31 \%$ a specifita $99.44 \pm 0.17 \%$ [52].

3.3.2 Konvoluční neuronové sítě

Konvoluční neuronové sítě jsou v detekci FS používány nejčastěji. Existuje velké množství CNN použitých pro tento účel, jednotlivé sítě se přitom liší počtem jednotlivých vrstev a způsobem reprezentace signálu EKG. Na vstup CNN může být přiveden samotný signál ve formě časové řady, ale také jako jeho vizualizace v časově frekvenční oblasti (spektrogram) [14].

V článku [14] je zmiňováno poměrně velké množství různých druhů CNN. Jeden z možných přístupů je použití multi-scale konvolučních neuronových sítí (MS-CNN) [51], které jsou obvykle založeny na procesu odvozování příznakových map pomocí analýzy kernelu filtru v různém měřítku. Tímto lze zachytit větší množství vhodných příznaků a také odhadnout rozložení relevantních příznaků vzhledem ke vstupním datům.

V [50] byla navržena MS-CNN se dvěma větvemi obsahujícími 13 konvolučních vrstev a 3 plně propojené vrstvy. Každá konvoluční vrstva obsahuje jinou velikost kernelu filtru. Byly testovány různé délky jednosvodových EKG signálů, jmenovitě 5 s a 20 s, kdy pro delší záznamy byla vyšší přesnost klasifikace a to 98.13%. Klasifikace probíhala pouze do dvou kategorií, na přítomnost a nepřítomnost FS [50].

Dále jsou zmiňovány klasické CNN s různými počty konvolučních vrstev, kdy se jejich počet pohybuje od 1 až po 18. Taktéž se liší počtem použitých svodů a délkou signálu. Jako další zajímavý přístup lze jmenovat použití FRM-CNN [49], která je použita k automatickému vyhledávání zajímavých segmentů z mobilních EKG záznamů. Může k tomu používat jak rytmus EKG, tak morfologické znaky EKG. Samotná FRM-CNN se skládá z 34 reziduálních bloků a 2 plně propojených vrstev

s aktivační funkcí softmax ve výstupní vrstvě. Klasifikace je rozdělena do 3 kategorií (FS, SR a ostatní) a nejvyšší dosažená přesnost je 87.22 ± 0.71 %.

Dále byl zvolen přístup zkombinování klasické CNN a vylepšené Elmanovy neuronové sítě (CNN-IENN) [48]. Kvůli ověření klasifikačních schopností CNN-IENN byly vytvořeny ještě konvoluční neuronová síť s plně propojenými vrstvami (CNN-MLP) a konvoluční síť s klasickou Elmanovou neuronovou sítí (CNN-ENN). Všechny implementované sítě měly velmi podobné parametry. Klasifikace signálů probíhala do 3 kategorií (FS, SR a ostatní). Nejvyšší přesnosti dosáhla CNN-IENN s přesností okolo 99 %, přičemž se přesná hodnota liší pro použití dat z různých zdrojů.

Další zajímavý přístup je kombinace klasické CNN a metody podpůrných vektorů (SVM - *support vector machine*). CNN extrahovala jednotlivé příznaky a SVM automaticky přiřazovala konkrétní signály k předem definovaným kategoriím (FS, SR, ostatní a šum). Pro vyhodnocení výsledků klasifikace sítě bylo použito F1-skore, které dosáhlo hodnoty 84.19 % [47].

3.3.3 Rekurentní neuronové sítě

RNN jsou obecně považovány za vhodné v případě zpracování sekvenčních dat s pevně danou strukturou, jako je např. EKG. Díky zpětnovazebným neuronům tak může síť lépe určovat změny v EKG rytmu, což by mohlo být přínosné zejména v určování začátku a konce paroxysmů nebo obecně ve změnách srdečních rytmů.

V článku [14] je zmiňována obousměrná RNN, kdy byly vytvořeny 4 rozdílné varianty: bez tzv. *attention* mechanismu a tři tzv. *hierarchical attention networks* s jedním, dvěma a třemi *attention* mechanismy. *Attention* mechanismus je blok v RNN, který umožňuje zaměřit se na určité části vstupu při predikci výstupu, což zlepšuje kvalitu a rychlost učení [29]. Nejlepší přesnosti dosáhly varianty RNN s více *attention* mechanismy. Klasifikace probíhala do dvou kategorií, na přítomnost a nepřítomnost FS. Přesnost pro RNN bez *attention* mechanismu je 98.58 %, pro RNN s jedním *attention* mechanismem 98.74 %, pro RNN se dvěma *attention* mechanismy 98.83 % a pro RNN se třemi *attention* mechanismy 98.81 % [46].

3.3.4 LSTM

Jelikož jsou sítě typu LSTM druhem RNN, tak se předpokládá obdobné využití jako u RNN. Nicméně by mohly být lépe využitelné u delších sekvencí.

V článku [14] je zmiňována obousměrná LSTM, která byla naučena extrahovat vzory z RR intervalu a dosáhla při tom přesnosti 98.51 %. Pro samotnou detekci FS dosáhla přesnosti 99.77 %. Jako další je zmíněna dvouvrstvá neuronová síť, která

dosáhla lepšího F1-skore pro detekci FS s augmentací dat (76.4 %) než bez augmentace (35.6 %). Tato síť klasifikovala do 4 kategorií (FS, SR, ostatní a šum) a byla trénována na neveřejném datasetu [45].

3.3.5 Hybridní modely

Hybridní modely jsou kombinací několika druhů rozdílných přístupů, které byly popsány výše. Nejčastějším přístupem je použití CNN vrstvy a následné LSTM, které lze použít pro vytvoření vysoce výkonných modelů [14]. V článku [31] byl navržen nenáročný CNN-LSTM model k detekci 8 různých srdečních arytmií včetně FS z nezpracovaného EKG signálu. Nejprve jsou extrahovány vzory EKG signálu a poté jsou zapamatovány jejich sekvenční umístění. Opačný přístup je zmiňován v článku [14], kdy jsou segmenty EKG přivedeny do LSTM a následný výstup přiveden na CNN, kde jsou extrahovány různé vzory, které jsou nakonec binárně klasifikovány na přítomnost nebo nepřítomnost FS.

4 Popis použitých dat a jejich předzpracování

V této kapitole budou popsána použitá data z hlediska základních statistických veličin, jejich uspořádání, formát a doplňující informace o měření použitých signálů. Dále bude popsáno předzpracování dat.

První část použitého datasetu byla poskytnuta brněnskou firmou MDT-Medical Data Transfer s.r.o. (dále jen MDT) ¹, která půjčuje pacientům z celé České republiky přístroje na dlouhodobé snímání srdeční aktivity. Firma zašle pacientovi přístroj poštou, pacient si sám nasadí přístroj podle návodu a po dobu, kdy má měření probíhat, jsou data z přístroje odesílána do telemedicínského centra, kde mohou být vyhodnocena buď v reálném čase nebo až po ukončení monitorace.

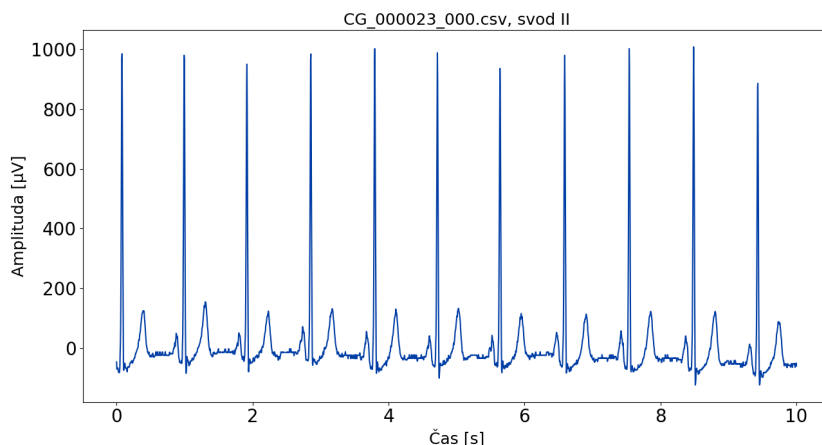
Dataset od MDT obsahuje celkem 60579 EKG záznamů získaných od pacientů centra. Každý záznam je označen jedním nebo více tagy. Možných tagů je 30 a označují různé patologie, například blokáda levého Tawarova raménka nebo komorové extrasystoly, sinusový rytmus nebo přítomnost šumu v signálu. Samotný EKG signál je uložen jako CSV soubor a obsahuje naměřená data zpravidla ze tří svodů – I, II a aVF. Právě 6908 signálů ovšem obsahuje pouze data ze svodu II. Ke každému CSV souboru se signálem náleží JSON soubor, který obsahuje informace o pacientovi (např. věk nebo pohlaví), jeho diagnózu, informace o měření, základní rozměření signálů (např. detekované pozice R peaků) a podmínkách snímání (např. vzorkovací frekvence). Dále jsou k dispozici souhrnné JSON soubory s celkovým zastoupením konkrétních patologií pro každý tag. Po zobrazení některých signálů bylo zjištěno, že signály nejsou filtrovány. Všechny záznamy mají vzorkovací frekvenci 250 Hz a kvantování 1 $\mu\text{V}/\text{LSB}$.

Analýzou anotací signálů bylo zjištěno, že průměrný věk pacientů je 59.79 let s nejvyšším zastoupením pacientů ve věku 65 let v počtu 1775 (viz Příloha B). Databáze dále obsahuje záznamy 31774 žen a 28805 mužů. Dále bylo zjištěno, že obsažené záznamy jsou v délce od 5 s do 144 s, většina souborů ovšem spadá do intervalu 10s až 40s, kdy nejčastější jsou ve dvou délkách, 10 s (2500 vzorků) a 40 s (10000 vzorků). Delší záznamy často pocházejí ze svodu II.

Data byla už předběžně rozdělena do zamýšlených kategorií, které by měly být detekovány (sinusový rytmus, FS celkově, ostatní patologie, paroxysmus FS a permanentní FS). V Tabulce 4.1 jsou zobrazeny číselné výčty hodnot pro jednotlivé kategorie.

Bylo tedy rozhodnuto o doplnění méně zastoupených kategorií více daty. Jako zdroj byla vybrána databáze George B. Moody PhysioNet Challenge 2021. Jedná

¹<https://www.mdt.cz>



Obr. 4.1: Příklad záznamu poskytnutého MDT. Zde se konkrétně jedná o záznam se sinusovým rytmem.

Název	SR	PERM/PAROX	OSTATNÍ	PREM	PAROX
Počet	13852	10245	36417	9220	1025

Tab. 4.1: Počty dat v jednotlivých kategoriích v původním MDT datasetu.

se o 12svodové EKG záznamy, které se skládají ze dvou částí - první je soubor typu .mat (binární MATLAB v4 soubor), ve kterém je samotný signál a druhý soubor je anotace signálu ve formátu .hea (prostý text) obsahující kódy patologií podle standardu SNOMED-CT² a další údaje, jako věk, pohlaví pacienta, vzorkovací frekvenci nebo počet vzorků. Data byla převzata z následujících zdrojů [37]:

- China Physiological Signal Challenge in 2018 (CPSC2018 a CPSC2018_2): Signály obsažené v tomto datasetu jsou dlouhé 6-144 s a mají vzorkovací frekvenci 500 Hz. Z tohoto zdroje byly převzaty záznamy obsahující převážně FS a paroxysmy FS.
- Physikalisch-Technische Bundesanstalt (PTB a PTB-XL): Záznamy dlouhé 10-120 s s vzorkovací frekvencí 500 Hz nebo 1000Hz. Z tohoto zdroje byly převzaty zejména záznamy obsahující FS.
- Georgia 12-Lead ECG Challenge Database (Ga): Záznamy dlouhé 5-10 s se vzorkovací frekvencí 500 Hz. Z tohoto zdroje byly převážně převzaty záznamy obsahující FS.
- Chapman University, Shaoxing People's Hospital (Chapman-Shaoxing): Záznamy jsou dlouhé 10 s se vzorkovací frekvencí 500Hz. Z tohoto zdroje byly převzaty převážně záznamy obsahující FS.

²<https://browser.ihtsdotools.org/>

- Ningbo First Hospital (Ningbo): Záznamy jsou dlouhé 10 s se vzorkovací frekvencí 500 Hz. Z tohoto zdroje byly převážně převzaty záznamy obsahující SR.

Vzhledem k tomu, že přidávané záznamy byly navzorkovány s jinou vzorkovací frekvencí než záznamy poskytnuté MDT, tak byly převzorkovány. Bylo zvoleno 250 Hz, jelikož je to vzorkovací frekvence již upravených dat. Užitečná informace v EKG se nachází do frekvence 150 Hz, takže nebudou odstraněny žádné důležité informace [38].

V Tabulce 4.2 jsou zobrazeny výsledné počty dat po rozšíření datasetu.

Název	SR	PERM/PAROX	OSTATNÍ	PREM	PAROX
Počet	24731	13985	36417	11585	2400

Tab. 4.2: Počty dat v jednotlivých kategoriích v rozšířeném datasetu.

4.1 Možnosti rozšíření datasetu pro kategorii paroxysmů

Bylo rozhodnuto o rozšíření nejméně početné kategorie paroxysmů FS. Kategorie paroxysmů FS původně obsahovala 2400 vzorků, což bylo přibližně 16 % z celkového množství záznamů označených jako fibrilace. Takto velké rozdíly v počtech záznamů bývají zejména u jevů, které nejsou příliš časté pro zadanou množinu případů (např. záznamy se srdeční patologií v databázi se zdravými jedinci). Tato nerovnováha pak může vést u neuronové sítě k nedostatečnému naučení se vzorů z nejméně zastoupené kategorie, protože se při rozdělení do dávky (*batch*) nemusí vůbec vyskytnout.

Pro vyrovnání počtů obou skupin je nejjednodušším řešením přidání zástupců kategorií, které nejsou příliš zastoupené, což bylo již provedeno v sekci výše s využitím další databáze.

Další velmi jednoduché metody jsou *undersampling* a *oversampling*. Při *undersamplingu* jsou odstraněny záznamy v kategoriích s více záznamy, což bylo použito v semestrální práci, kdy počet všech záznamů v kategorii byl redukován na 2400. *Oversampling* je jednoduše řečeno nakopírování dat z méně zastoupených kategorií na počty v kategoriích, které obsahují více dat. Záznamy je však nutné rozkopírovat až po rozdělení dat na trénovací a testovací množinu, aby nedošlo ke zkreslení výsledků. Tato metoda byla taktéž použita v semestrální práci. Všechny kategorie byly rozkopírovány na celkový počet 20000 záznamů pro každou kategorii.

Pokud však nejsou nejsou předchozí metody dostačující, pak se přistupuje k umělému vytváření nových dat na základě těch stávajících. Tento přístup je také známý jako augmentace dat.

Mezi klasické metody augmentace lze zařadit transformace dat. U obrazových dat se nejčastěji přistupuje k rotacím, výřezům určitých částí obrazu, zvýraznění hran, změně barvy nebo přidání šumu. Další možností, nicméně poměrně složitou, jsou sítě typu GAN (*Generative Adversarial Network*), což je druh neuronové sítě, která vytváří nová realisticky vypadající data na základě dat ze vstupujícího datasetu. Skládá se ze dvou neuronových sítí, jedné zvané generátor, která je zodpovědná za vytváření nových záznamů a diskriminátor, který se snaží odhalit, jestli záznam na jeho vstupu je z původního datasetu nebo nově vygenerovaný. Generátor se snaží generovat záznamy nerozpoznatelné od originálu a diskriminátor se snaží vylepšovat v jejich správném odhalování [53].

V bakalářské práci byly implementovány filtry na filtraci síťového rušení 50 Hz a kolísání nulové izolinie. Myopotenciály byly však ponechány, takže by bylo možné použít jak signály s korekcí nulové izolinie, tak bez ní, popřípadě vytvořit funkci přičítající Gaussovský šum k původnímu záznamu. Možné je i přeházení segmentů signálu ohraničeného mezi dvěma R peaky, což však u paroxysmů nedává moc smysl, protože by mohlo dojít ke ztrátě informace ohledně začátků paroxysmů. Bylo by však možné provést změnu pořadí celého segmentu s paroxysmem.

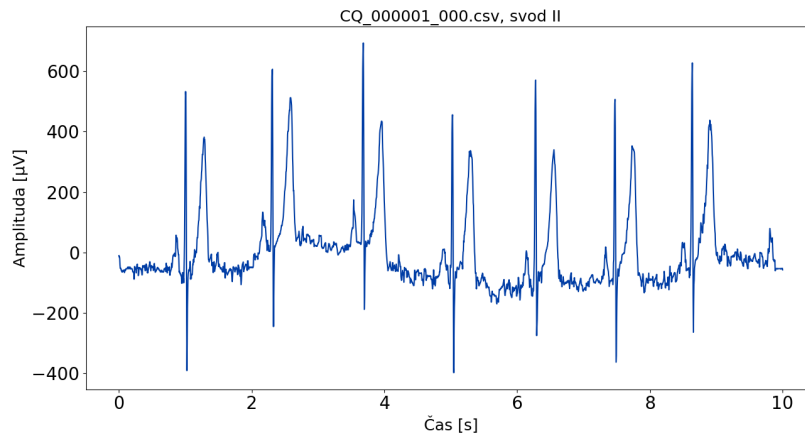
Nakonec byla zvolena metoda augmentace signálů z kategorie paroxysmů pomocí Gaussovského (4.3) a impulsního šumu (4.4). Signály byly zašuměny po filtraci nulové izolinie a síťového rušení, nicméně toto zašumění lze aplikovat na signál i před filtrací. Nakonec bylo získáno trojnásobné množství signálů pro tuto kategorii. Obě tyto funkce byly vytvořeny v balíčku `fs_signal.py`.

Vliv různých druhů augmentace na učení neuronové sítě bude popsán později v podsececi Ladění parametrů klasifikátorů 5.3.

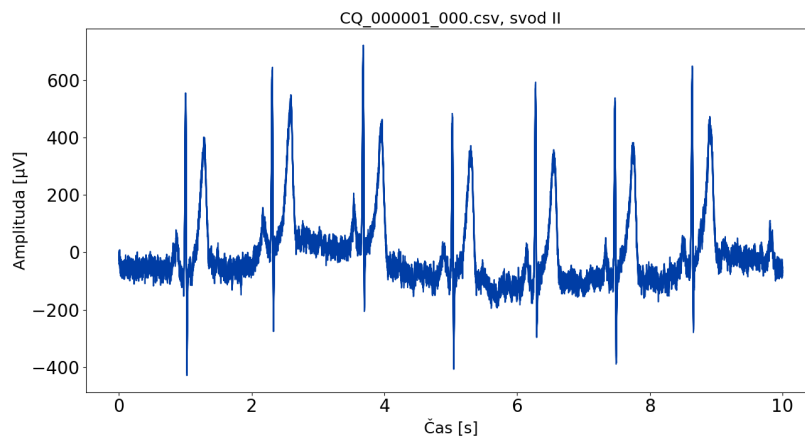
4.2 Předzpracování dat

Kvůli nutnosti jednotné délky pro vstupní data do CNN byla předběžně zvolena délka 10 s (2500 vzorků při vzorkovací frekvenci 250 Hz), kvůli největšímu zastoupení v datasetu. Záznamy kratší než 10 s (65 záznamů) byly prozatím vyřazeny kvůli nevhodnosti pro zadaný klasifikační problém (viz 4.2.2).

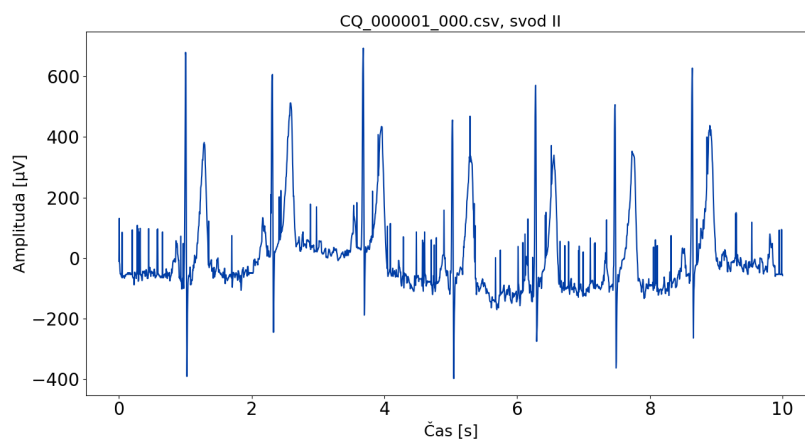
Pro co nejvyšší úspěšnost klasifikace je potřeba data vhodně předzpracovat a upravit. Funkce pro filtraci, standardizaci a úpravu délky signálu se nacházejí v balíčku `fs_signal.py`. Funkce pro vytvoření labelů je vytvořena v balíčku `train_test.py`.



Obr. 4.2: Signál bez umělého zašumění.



Obr. 4.3: Signál s Gaussovským šumem.



Obr. 4.4: Signál s impulsním šumem.

4.2.1 Vytvoření labelů

Pro korektní klasifikaci je nutné přiřadit dostupným datům správné označení, které je známé také jako label. Nejprve musí být určeno, do kolika kategorií budou data klasifikována. Poté jsou záznamy označeny odpovídajícím labelem příslušné kategorie. V této bakalářské práci budou implementovány klasifikátory do 3 kategorií a do 4 kategorií. Ke každému záznamu byl přiřazen právě jeden label z následujících kategorií:

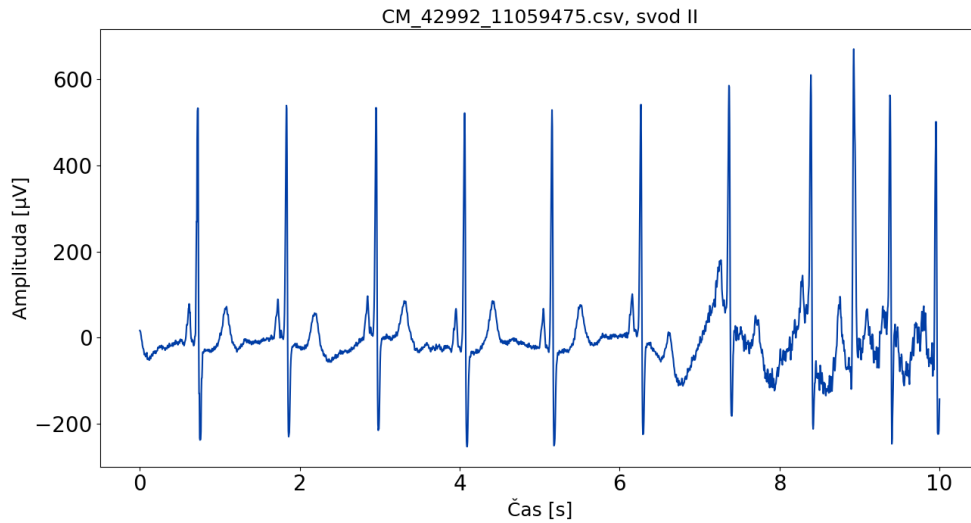
- SR: Obsahuje data s tagem SR v anotaci souboru, pokud to byl jediný uvedený tag a v záznamu nebyla označena žádná další patologie.
- PERM: Obsahuje data s tagem AFIB v anotaci souboru, kdy může být přítomno i více různých diagnóz. Záznamy v této kategorii mají přítomnu FS v celé své délce bez ohledu na délku záznamu.
- PAROX: Obsahuje data s tagem AFIB v anotaci souboru, kdy může být přítomno i více různých diagnóz. Záznamy v této kategorii mají přítomnu FS pouze v části záznamu. Můžou obsahovat pouze začátek, pouze konec nebo začátek i konec FS v záznamu bez ohledu na jeho délku.
- OTHER: Obsahuje všechna ostatní data nezařazená do výše zmíněných skupin.

Kategorie PERM a PAROX byly v datech z databáze *George B. Moody PhysioNet Challenge 2021*³ již anotovány, avšak v MDT databázi od sebe jednotlivé druhy FS nebyly nijak odlišeny. Proběhla proto ruční anotace záznamů na základě vizuálního posouzení signálu, kdy byla relevantní především pravidelnost RR intervalu a přítomnost fibrilačních vlnek. Ke všem dostupným záznamům byly tak byla přiřazeny odpovídající labely.

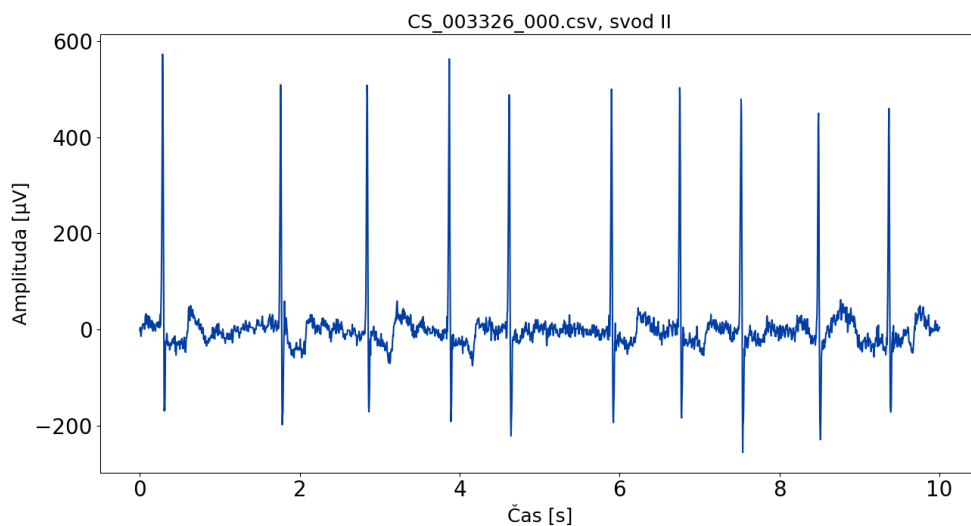
Kvůli zpracování neuronovou sítí byly labely zapsány pomocí *one hot encoding* vektoru. Při klasifikaci do 3 kategorií jsou kategorie PERM a PAROX brány jako jedna společná a při klasifikaci do 4 kategorií jsou všechny kategorie použity samostatně.

Na Obrázku 4.5 je příklad záznamu zařazeného do kategorie PAROX. Na záznam byla aplikována korekce kolísání nulové izolinie a síťového rušení. Je zobrazeno jen prvních 10 s záznamů s viditelným přechodem ze SR do FS. Do datasetu byl záznam zařazen v původní délce 40 s. Na Obrázku 4.6 je záznam zařazený do kategorie PERM, kdy jsou v signálu patrné fibrilační vlnky a nepravidelný RR interval v celé jeho délce. Na Obrázku 4.7 je zobrazen záznam zařazený do kategorie OTHER. Konkrétně se jedná o záznam se sinusovým rytmem, infarktem myokardu, bradykardií a depresí ST intervalu.

³<https://moody-challenge.physionet.org/2021/>



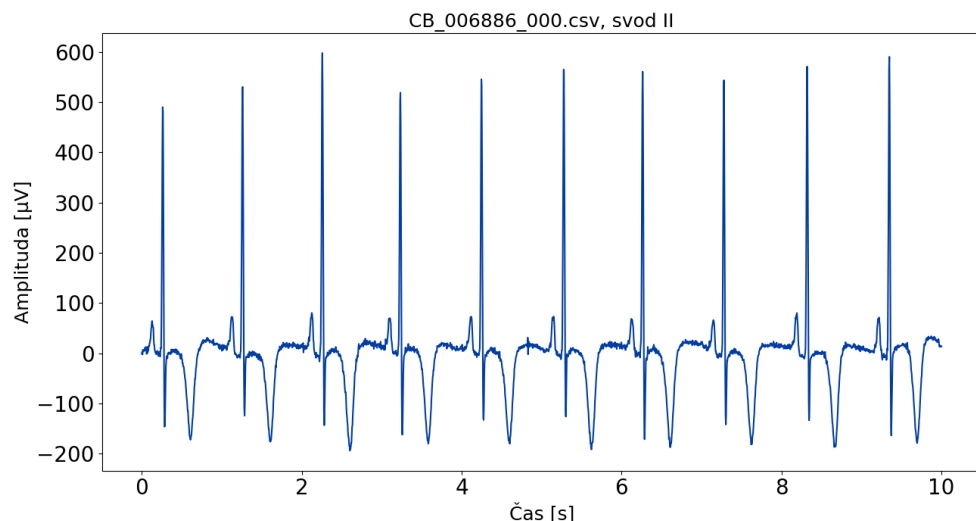
Obr. 4.5: Příklad záznamu zařazeného do kategorie PAROX.



Obr. 4.6: Příklad záznamu zařazeného do kategorie PERM.

4.2.2 Úprava délky signálu

CNN mají pevně danou velikost vstupu, která odpovídá rozměrům vstupních dat, proto je žádoucí, aby všechny záznamy měly stejné rozměry. Jak bylo již zmiňováno výše v této kapitole, tak záznamy se liší svou délkou, tudíž bylo potřeba vytvořit funkce pro prodloužení a zkrácení signálu na požadovanou délku. Funkce pro zkrácení byla testována především pro klasifikaci do tří kategorií, protože zde příliš nezáleží na konkrétní lokalizaci začátku nebo konce FS, nicméně u paroxysmů by mohlo dojít ke ztrátě důležitých informací.



Obr. 4.7: Příklad záznamu zařazeného do kategorie OTHER.

Funkce pro zkrácení signálu zkrátí signál na požadovanou délku ve vzorcích. Implementace je velmi jednoduchá. Je vybráno okno v dané délce ze středu signálu a signál je následně oříznut. Okno ze středu signálu bylo vybráno kvůli co největší pravděpodobnosti na zachycení případného konce nebo začátku FS při paroxysmu, případně jiných patologií.

Dále bylo nutné vyřešit prodlužování signálu. Podle dostupných zdrojů jsou nejběžnější dva postupy: prodloužit signál nulami (*zero-padding*), nebo signál rozkopírovat na požadovanou délku. Jako inspirace byly použity články [43] a [44]. Při prodlužování nulami je možné je přidávat na začátek, na konec nebo doporučně na oba konce signálu. Zbytek způsobů popsaných v článku [43] není pro EKG signál příliš vhodný, takže nebyly v rámci této práce realizovány.

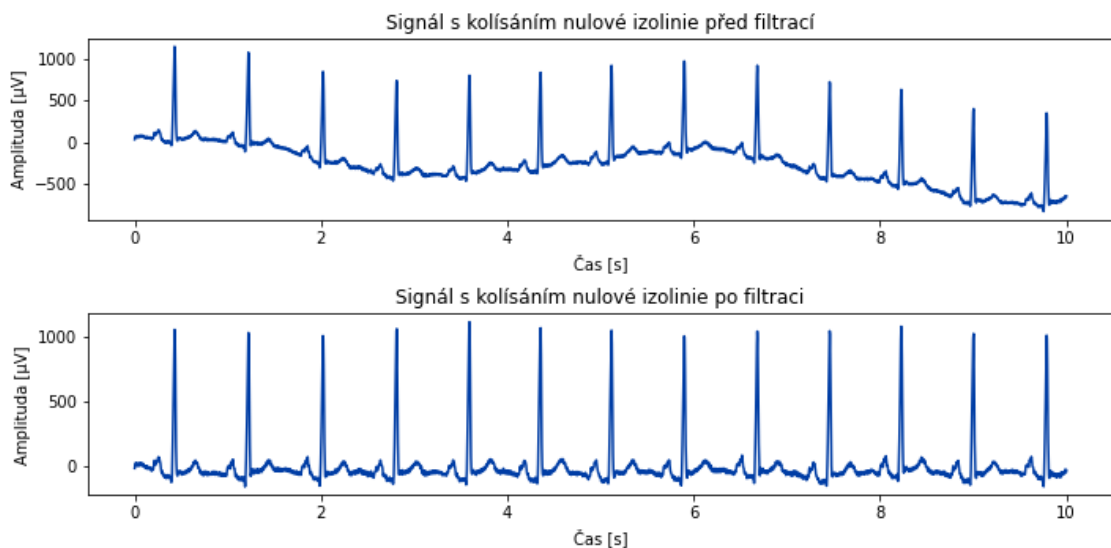
4.2.3 Filtrace dat

Nezákladnější úpravou EKG je jeho filtrace. Mezi nejčastější zdroje rušení patří síťové rušení (50 Hz v Evropě a 60 Hz v USA), kolísání nulové izoliny (drift) a myopotenciály.

Síťové rušení je možné odstranit pomocí úzkopásmového filtru typu pásmová zádrž s mezní frekvencí 50 Hz nebo 60 Hz. Zde byl použit Butterworthův filtr s mezní frekvencí 50 Hz a řádem filtru 2. Mezní frekvence 50 Hz byla zvolena s ohledem na to, že všechna dostupná data byla naměřena v Evropě. Jelikož se však jedná o IIR filtr, bylo nutné použít dopřednou a zpětnou filtraci, čímž byl získán nulový fázový posun a došlo ke zdvojnásobení řádu filtru na 4 [39]. Některé signály před filtrací síťového rušení byly převedeny do frekvenční oblasti a zobrazeny, nicméně nebyl

zaznamenán žádný výrazný peak v okolí 50 Hz. Možnost této filtrace byla však ponechána zejména kvůli dalšímu možnému doplňování záznamů.

Kolísání nulové izolinie je rušení o nízké frekvenci do 1 Hz způsobené převážně dýcháním, pohyby pacienta nebo elektrochemickými ději na rozhraní pokožky a elektrody [40]. Toto rušení může být filtrováno pomocí horní propusti s velmi nízkou mezní frekvencí okolo 0.5 Hz. Je to kvůli nelineární fázové charakteristice filtru, která způsobuje zkreslení signálu. Další možností je detekce určených bodů (PR interval, kvůli nulové izolinii v EKG zdravých jedinců) v EKG křivce a jejich interpolace pomocí např. kubického spline [39]. Byly vyzkoušeny obě metody s téměř totožnými výsledky, nicméně filtrace horní propustí byla o něco přesnější u signálů s velkým kolísáním nulové izolinie. Konkrétně byl použit Butterworthův filtr typu horní propust s mezní frekvencí 0.52 Hz a řádem filtru 3. Opět byla použita dopředná a zpětná filtrace k eliminaci nelineární fázové charakteristiky. Na Obrázku 4.8 je v horní části zobrazen signál před filtrací a v dolní části tentýž signál po filtraci.



Obr. 4.8: Příklad signálu před korekcí kolísání nulové izolinie (nahore) a signálu po filtraci kolísání nulové izolinie (dole).

V případě myopotenciálů se jedná o širokospektrální rušení, které zasahuje do užitečného pásma EKG. Jeho frekvenční rozsah se pohybuje od 10 až 500 Hz [40], takže se překrývá zejména s frekvencemi QRS komplexu. Z výše uvedených faktů vyplývá, že lineární filtrace není vhodná, protože by při ní došlo k nežádoucímu zkreslení signálu. Nejčastěji se k filtraci myopotenciálů používá vlnková filtrace nebo kumulační techniky. Kvůli možné ztrátě informace bylo rozhodnuto myopotenciály zatím nefiltrovat.

Dále byla přidána možnost rozhodnout se, zda má být vstupní signál filtrován či ne. Ve výchozím nastavení je zvolena možnost filtrace.

4.2.4 Standardizace

Dalším důležitým krokem je standardizace dat, kdy jsou data převedena na stejný rozsah hodnot, takže přestává záležet na skutečném rozměru jednotlivých proměnných. K nejčastější metodám patří standardizace směrodatnou odchylkou a standardizace rozpětím [41].

U standardizace směrodatnou odchylkou je od každé naměřené hodnoty odečtena průměrná hodnota všech měření a poté je výsledek vydělen směrodatnou odchylkou z průměru hodnot všech měření [41].

U standardizace rozpětím jsou data převedena na rozpětí od daného minima pro dané maximum. Nejčastěji od 0 do 1 [41].

Zde byla zvolena standardizace rozpětím pomocí funkce `MinMaxScaler()`⁴ z knihovny `sklearn`. Rozpětí je zvoleno od -1 do 1, kvůli většímu rozsahu, na který mohou být původní rozměry přepočteny.

4.3 Rozdělení na trénovací a testovací množinu

Kvůli jednoduššímu přístupu k datům byla vytvořena tabulka ve formátu JSON s následujícími údaji: ID signálu, label s názvem detekované patologie a signál ze svodu II. Svod II byl zvolen kvůli dostupnosti u všech signálů a také protože právě v tomto svodu jde dobře rozlišit FS od jiných patologií.

Data byla rozdělena do dvou nepřekrývajících se množin; trénovací a testovací. Byl zvolen poměr 80% pro trénovací množinu a 20% pro testovací množinu. Postupně byly zkoušeny i jiné poměry, například 85% a 90% pro trénovací množinu a zbylá procenta pro testovací množinu, ale nebylo tam znatelné zlepšení v přesnosti klasifikace.

Nejprve byl zvolen přístup s JSON formátem dat jako výchozím souborem, odkud budou záznamy zpracovávány. Byly vybrány záznamy v předem zvoleném poměru a rozděleny na trénovací a testovací množinu. Obě množiny byly promíchány pomocí funkce `shuffle()`⁵. Pak byla data rozdělena na množiny testovacích signálů a jim odpovídající labely. Obdobně byla zpracována i data z trénovací množiny. Jelikož však byla následná manipulace s těmito daty výpočetně náročná, bylo rozhodnuto o změně formátu na CSV. Oproti předchozímu návrhu byl zachován pouze label

⁴<https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html>

⁵<https://docs.python.org/3/library/random.html>

a k němu příslušný signál. Stejně jako v předchozím případě byl následně vybrán určitý počet záznamů pro každou kategorii. K tomu byla využita funkce `sample()`. Vybraná data jsou následně vložena do listu a načtena jako *pandas dataframe* pomocí funkce `DataFrame()`⁶. Jak label, tak samotný signál, jsou datového typu pole (`np.array`) s hodnotami `np.float64`, který umožňuje uložení čísel s dostatečnou přesností. Samotné rozdělení na trénovací a testovací množinu je provedeno funkcí `train_test_split()`⁷ a výstupem jsou jednotlivé množiny `X_train`, `X_test` (obsahující signály) a `y_train`, `y_test` (obsahující labely).

Jak je vidět z procentuálního zastoupení dat v Tabulce 4.3, je zřejmé, že se jedná o nevyvážený dataset, jelikož počet případů v kategorii PAROX a i obecně pro jakýkoliv detekovaný typ FS je výrazně nižší než u ostatních kategorií. Tento dataset uvedený v Tabulce 4.3 byl použit pro testování na jednoduché 1D CNN vycházející z několika článků pro množinu dat o délce 10 s a poté 40 s. Obsahovala 4 konvoluční vrstvy se stoupajícím počtem filtrů (32, 64, 64, 128) a velikost filtru 3, dále jednu dropout vrstvu, jednu pooling vrstvu s velikostí kroku 2 a plně propojenou vrstvu o velikosti 100. Pro 10 s záznamy byla přesnost 66,6% a pro 40 s záznamy byla přesnost 72,7%.

Patologie	SR	PERM/PAROX	OTHER	PREM	PAROX
Procentuální zastoupení	27,75	15,69	40,86	13,00	2,70

Tab. 4.3: Procentuální zastoupení po jednotlivé typy patologií v neaugmentovaném datasetu.

Po tomto experimentu bylo rozhodnuto o vytvoření vyváženého datasetu doplněním augmentovaných dat. V Tabulce 4.4 je vidět procentuální zastoupení augmentovaného datasetu. Do kategorie OTHER nebyly zahrnuty náhodně vybrané záznamy, nicméně v celkové CSV tabulce byly zanechány.

Patologie	SR	PERM/PAROX	OTHER	PREM	PAROX
Procentuální zastoupení	29,54	22,44	25,58	13,84	8,60

Tab. 4.4: Procentuální zastoupení po jednotlivé typy patologií v augmentovaném datasetu.

⁶<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html>

⁷https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

5 Návrh a implementace neuronové sítě

V této kapitole bude popsán návrh a implementace dvou neuronových sítí. První z nich klasifikuje vstupní data do 3 kategorií, druhá poté do 4 kategorií.

Implementace neuronové sítě bude realizována v jazyce Python (verze 3.10.1). Jedná se o vysokoúrovňový interpretovaný jazyk s dynamickou kontrolou datových typů. Podporuje také různá programovací paradigmaty, např. objektové nebo imperativní. Python je velmi univerzální v použití od datové analýzy, přes tvorbu webových stránek, až po automatizační problémy. V neposlední řadě je dnes velmi oblíbený v oblasti strojového učení, zejména kvůli nepřebornému množství rozsáhlých knihoven a frameworků, jako například PyTorch¹, Keras² nebo TensorFlow³.

Po prostudování dostupných článků zabývajících se klasifikací EKG byla zvolena 1D CNN založená na [42]. Původní CNN slouží jako klasifikátor srdečních arytmií do 16 různých kategorií. Jako vstupní data jsou použity 2 svody EKG signálu o délce 1 až 2 s, které se z 10 % překrývají.

V rámci semestrální práce byly vytvořeny dvě CNN - klasifikátor do 3 kategorií (model_3) a klasifikátor do 4 kategorií (model_4). V původní konfiguraci z článku [42] a s použitím dat pouze od MDT nedosáhla žádná z vytvořených CNN více jak 60% přesnosti, takže bylo rozhodnuto o upravení obou CNN pro dostupná data.

Byla vybrána 1D CNN architektura, kvůli největšímu zastoupení v článcích zabývajících se touto tematikou. Pro její vytvoření byly použity knihovny Keras a TensorFlow. Její uspořádání vrstev bylo převzato z článku [42], její hyperparametry byly zvoleny na základě shrnutí různých hodnot z odborné literatury. Pokud změna hyperparametru znamenala zlepšení sítě, tak byl parametr upraven, pokud následovalo zhoršení, tak byla hodnota vrácena na původní hodnotu. Postupně bylo upravováno i pořadí vrstev a hodnoty jejích argumentů.

Po prvních návrzích CNN a provádění výpočtů na CPU byla v rámci pokusů o zkrácení doby trénování při hledání vhodné kombinace hyperparametrů a uspořádání vrstev zapojena technologie CUDA⁴ (verze 11.6.2) a její knihovna cuDNN⁵. Jedná se o paralelní výpočetní platformu a programovací model vytvořený společností NVIDIA pro obecné výpočty na GPU. Je možné dosáhnout znatelného zrychlení při výpočtech oproti využití CPU.

CUDA je multiplatformní, tudíž podporuje velké množství jazyků (C/C++, Python, MATLAB, ...) a je aktivně využívána v mnoha odvětvích od vědeckých výpočtů přes medicínu až po zpracování videa. Zde došlo k výraznému zrychlení obou

¹<https://pytorch.org/>

²<https://keras.io/>

³<https://www.tensorflow.org/>

⁴<https://developer.nvidia.com/cuda-zone>

⁵<https://developer.nvidia.com/cudnn>

CNN při využití 10000 záznamů pro každou kategorii na celkovou dobu okolo 7 minut oproti původním 30 minutám, takže prototypování CNN mohlo být efektivnější a rychlejší.

Vzhledem k tomu, že použitá knihovna TensorFlow si standardně alokuje všechnu dostupnou paměť na začátku běhu, takže byla přidána podmínka zvýšení alokované paměti v závislosti na aktuální potřebě.

5.1 Model pro klasifikaci do 3 kategorií

Navržená neuronová síť se skládá z konvolučních vrstev, pooling vrstev, plně propojených vrstev a několika dropout vrstev. Jak již bylo popsáno v kapitole 2, nejdůležitější částí sítě jsou konvoluční vrstvy, které postupně vybírají příznaky ze vstupních dat. Zde jich bylo zvoleno celkem sedm. Počet jader filtru je nejvyšší v první konvoluční vrstvě a postupně se snižuje. Konkrétně filtr nabývá hodnot 100, 50, 25, 10, 5, 3 a 3. Počet filtrů stoupá s každou konvoluční vrstvou od hodnoty 2 ve vstupní vrstvě přes 32, 64, 128, 512, 128 v následujících vrstvách a nakonec 64 v poslední vrstvě. Pooling vrstvy měly původně velikost kroku 2, takže zmenšovaly velikost signálu na polovinu. Později bylo testována radikálnější varianta s velikostí kroku 4 v první vrstvě, ve zbylých byly ponechány výchozí hodnoty, tedy 2. Vybrána byla metoda max pooling, takže jsou na výstup vybrány jen nejvýraznější příznaky. Původní model obsahoval pouze jednu dropout vrstvu, ale ukázalo se, že to nestačí k zabránění přeučení, takže byly přidány další dropout vrstvy. První byla umístěna mezi poslední pooling vrstvou a *flatten* vrstvou. Další je těsně před výstupní vrstvou a poslední mezi plně propojenými vrstvami. Všechny mají shodný parametr 0.5, což udává 50% pravděpodobnost pro deaktivaci neuronů.

V této CNN jsou použity dva druhy aktivačních funkcí (viz Příloha A). První je ReLU (*Rectified Linear Unit*) a je použita zejména v konvolučních vrstvách a v prvních dvou plně propojených vrstvách. Jedná se o částečně lineární funkci která, pokud je hodnota vyšší než nula, tak předává výstupy z předchozích vrstev přímo na výstup aktivační funkce a ve zbylých případech převede výstupy na nulu (rovnice 5.1). Druhá je softmax (rovnice 5.2), kde K značí počet kategorií. Jedná se o nejčastěji používanější funkci pro klasifikaci do více kategorií. Použitá je pouze ve výstupní vrstvě, kde určuje pravděpodobnost zařazení výstupu do předem dané kategorie. Proto vyžaduje převedení labelu na *one hot encoding* vektor.

$$f(x) = \max(0, x) \quad (5.1)$$

$$f(x) = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}} \quad (5.2)$$

Pak bylo potřeba najít správné hyperparametry, které musely být stanoveny před začátkem učení. Jejich hodnoty výrazně ovlivňují výsledek učení a pokud jsou nevhodně nastaveny, může dojít například k přeučení sítě. Mezi základní hyperparametry patří počet epoch, dávka (*batch size*) a krok učení. Počet epoch udává počet úplných průchodů trénovací množinou. Pokud je počet epoch nízký, síť se pak nebude schopná dostatečně naučit správné vzory, nicméně pokud bude epoch příliš mnoho, tak se síť přeučí a nebude správně generalizovat na testovací množině. Po prostudování dostupné literatury bylo předběžně zvoleno 12 učících epoch. Dávka (*batch size*) je počet záznamů, které jsou zpracovány před výpočtem ztrátové funkce a úpravou parametrů sítě. Nejvíce oblíbené jsou hodnoty 32, 64 a 128. Zde však musel být brán ohled na paměťovou kapacitu použitého GPU, takže byla zvolena hodnota 60.

Jako ztrátová funkce zde byla zvolena křížová entropie⁶ (*categorical cross entropy loss function*), která je jednou z nejpoužívanějších pro klasifikační problémy do více kategorií. Pro použití vyžaduje labely ve formátu *one hot encoding*. Na základě jejích výpočtů je určena chyba mezi vstupem a výstupem. Pro úpravu vah modelu je použit optimalizační algoritmus ADAM⁷, který přidává do výpočtu stochastického gradientního sestupu i hybnost. Hledání minima ztrátové funkce je tak ovlivňováno i předchozími kroky, které jsou postupně exponenciálně zapomínané. Algoritmus má 4 parametry: *alpha* (velikost kroku s kterým probíhá učení), *beta1* a *beta2* (zapomínací parametry) a *epsilon* (velmi malé číslo, které má zabránit dělení nulou). Tyto parametry byly ponechány výchozí, tak jak byly implementovány v knihovně Keras (*alpha* = 0.001, *beta1* = 0.9, *beta2* = 0.999 a *epsilon* = $1 \cdot 10^{-7}$).

Dále bylo potřeba vybrat správný učící dataset. Po nízké přesnosti na nevyváženém datasetu od MDT v semestrální práci bylo vytvořeno několik datasetů s daty získanými z jiných databází. Byla k tomu použita funkce `create_dataset()`, která umožňuje vybrat si požadované množství záznamů, zvolit si jaké záznamy budou použity (původní data, augmentovaná data, pouze záznamy se SR nebo pouze záznamy s PERM/PAROX) a také zvolit si možnost standardizace nebo filtrování, které byly popsány v kapitole 4. V základním nastavení jsou všechny signály filtrované a standardizované. Nakonec bylo předběžně vybráno 8000 záznamů pro každou kategorii.

⁶https://keras.io/api/losses/probabilistic_losses/

⁷<https://keras.io/api/optimizers/adam/>

Kromě jiného bylo dále třeba zvolit správnou délku signálu vzhledem k detekci FS. Nejvyšší zastoupení v datasetu mají záznamy dlouhé 2500 vzorků (10 s), nicméně tato délka není vhodná vzhledem k nutnosti zkracování delších signálů, kdy by mohlo dojít ke ztrátě informace. Další nejvíce zastoupená délka signálu je 10000 vzorků (40 s), která byla taktéž testována jako vhodná délka pro trénovací data, přičemž se nakonec ukázala být nejvhodnější.

5.2 Model pro klasifikaci do 4 kategorií

Stejně jako model_3 se i tato CNN skládá z konvolučních vrstev, pooling vrstev, plně propojených vrstev a dropout vrstev. Uspořádání vrstev této neuronové sítě vychází z článku [42] stejně jako u předchozího modelu_3. Byla taktéž provedena změna počtu neuronů ve výstupní vrstvě na 4. V semestrální práci byly model_3 a model_4 spojeny a volba klasifikátoru tak probíhala pouze změnou počtu výstupních neuronů skrze podmínku. Toto uspořádání se však ukázalo jako nepraktické, takže nakonec byly oba klasifikátory rozděleny a pro model_4 byly hledány optimální hyperparametry samostatně. Bylo zvoleno 7 konvolučních vrstev se stoupajícím počtem jader filtrů (50, 50, 25, 10, 5, 3, 3). Počet filtrů stoupá v každé vrstvě s hodnotami 2, 32, 64, 128, 512, 128 a 64. Po každé konvoluční vrstvě následuje max pooling vrstva s hodnotou kroku 2, vyjma první pooling vrstvy, která má krok 4. Další následují 4 plně propojené vrstvy s různým počtem neuronů (128, 128, 64 a 4) a celkem 3 dropout vrstvy. Jako aktivační funkce jsou použity ReLU a softmax. Ztrátová funkce a optimalizační algoritmus jsou totožné jako u modelu_3.

Dále bylo potřeba najít vhodnou délku signálu tak, aby nedošlo k případným ztrátám přechodů mezi FS a SR, které jsou typické pro záznamy označené jako PAROX. Byly zjištěny délky ve vzorcích v této kategorii, které se nejčastěji pohybují okolo 12000 vzorků (odpovídá 48 s).

Podobně jako u předchozího modelu bylo vytvořeno několik datasetů, u nichž byla testována ideální délka ve vzorcích a počet záznamů pro každou kategorii. Předběžně bylo zvoleno 12000 vzorků.

5.3 Ladění parametrů klasifikátorů

Postupně byly navrženy hodnoty různých parametrů s cílem vyzkoušet jejich vliv na přesnost obou CNN. Vždy je k testování parametru přistupováno tak, že je měněna hodnota jednoho vybraného parametru, zatímco ostatní jsou ponechány beze změny. Pokud se výsledky mezi modelem_3 a modelem_4 výrazně liší, je tato skutečnost uvedena v příslušné podkapitole. Pro obě CNN jsou v Přílohách C až M uvedeny

grafy se změnou závislosti jednotlivých parametrů na přesnost sítí. Parametr přesnost byl vybrán na základě vyváženého datasetu použitého pro trénování obou CNN.

5.3.1 Aktivační funkce

Ve skrytých vrstvách byla ponechána ReLU, která je standardně používána pro skryté vrstvy v CNN [55]. Byly však vyzkoušeny i jiné známé aktivační funkce: leaky ReLU, SELU, ELU, PReLU, hyperbolický tangens a sigmoida. Všechny použité funkce byly již naimplementovány v knihovně Keras. Jako nejpřesnější se jeví ReLU s přesností 81.75 %, nicméně hodnota leaky ReLU byla velmi podobná, rozdíl činil zhruba 1 %. Funkce ELU, PReLU a SELU všechny dosahovaly podobné přesnosti (okolo 78 %), přičemž nejvíce dosáhla ELU s 79.15 %. U SELU bylo zjištěno rychlé přeučení v porovnání s ostatními aktivačními funkcemi při zachování ostatních parametrů. Sigmoida dosáhla v průměru 32.96% přesnosti. Dle dostupných materiálů [54] je spíše vhodná do skrytých vrstev pro RNN a LSTM a potřebuje více epoch pro dosažení akceptovatelné přesnosti. Nízké přesnosti okolo 50 % dosáhl i hyperbolický tangens. Tato funkce navíc vykazovala velmi vysokou variabilitu v přesnosti mezi jednotlivými běhy, hodnoty se pohybovaly mezi 30 a 70 %. Konečný výsledek do tabulky musel být zprůměrován. U ostatních funkcí tento problém nebyl zaznamenán. Do výstupní vrstvy byla vybrána funkce softmax a nebyly testovány jiné alternativy. Více v Příloze C.

5.3.2 Dropout vrstvy

Ve všech třech dropout vrstvách byl nastaven parametr 0.5, tedy 50% pravděpodobnost deaktivace každého z neuronů. Byly otestovány dvě možnosti zlepšení přesnosti: změnit počet dropout vrstev nebo jejich parametr. Při snižování dropout parametru byl vidět velký rozdíl mezi přesností na trénovací množině, který však při vyšších hodnotách skoro vymizel, avšak výsledná přesnost na testovací množině byla celkově nižší pro nižší hodnoty parametru. Dále byl otestován i různý počet dropout vrstev, kdy se s přidáváním více vrstev snižoval rozdíl mezi trénovací a testovací množinou. U tří dropout vrstev bylo zjištěno, že je rozdíl v přesnosti mezi trénovací a testovací množinou nejmenší, ale za cenu snížení výsledné přesnosti oproti jiným počtům dropout vrstev. Pro model_4 byly výsledky velmi podobné. Při použití parametru 0.5 pro dropout vrstvu je nejmenší rozdíl mezi testovací a trénovací množinou, aniž by došlo ke snížení celkové přesnosti. Při testování počtu dropout vrstev jsou výsledky taktéž podobné. Více v Přílohách D a E.

5.3.3 Pooling vrstvy

Celkový počet pooling vrstev byl nastaven na 6 a jako pooling metoda byl zvolen max pooling. V první vrstvě je velikost kernelu nastavena 4 a ve všech ostatních 2. Byly vyzkoušeny i jiné hodnoty jako například nastavení velikosti kroku na 4 ve všech pooling vrstvách nebo různé kombinace, ale žádné z nich nevedly ke znatelnému zlepšení. Dále byla otestována metoda average pooling jako případná náhrada za max pooling, ale výsledky byly téměř totožné. Byla tedy ponechána metoda max pooling z důvodu její menší výpočetní náročnosti. Pro model_4 byly vyzkoušeny stejné varianty a výsledky byly téměř totožné s modelem_3. Více v Příloze F.

5.3.4 Konvoluční a plně propojené vrstvy

Bylo použito 7 konvolučních vrstev oproti původní CNN, která jich obsahovala pouze 5. Bylo také zjištěno, že s narůstajícím počtem skrytých vrstev je síť náchylnější k přeučení, které je pak nutné regulovat.

Jak již bylo řečeno, tak jedny z nejdůležitějších parametrů konvolučních vrstev jsou parametr konvolučního jádra a počet filtrů.

Nejprve byl počet jader filtru byl zvolen shodně ve všech vrstvách stejný, a to 3. Postupně byl ovšem měněn počet jader pro každou konvoluční vrstvu.

Velikost konvolučního jádra v první vrstvě byla zkoušena v závislosti na charakteristikách EKG signálu (délka periody a jejich částí ve vzorcích). Byla vyzkoušena velikost konvolučního jádra o hodnotě 50, což odpovídá maximální povolené délce PQ intervalu, hodnota 25, což odpovídá délce P vlny, a 100, což by mělo zahrnovat vlnu P a QRS komplex. Nejvyšší přesnost byla dosažena na délce 100. Všechny délky byly počítány ve vzorcích při vzorkovací frekvenci 250 Hz.

Kromě proměnlivé velikosti konvolučního jádra byla testována i situace, kdy ve všech vrstvách byla použita stejná hodnota. K tomu byla zvolena hodnota 25 a bylo zaznamenáno zlepšení do 1 % oproti ostatním testovaným hodnotám. Dále byla testována hodnota 10, ale byly zaznamenány téměř totožné výsledky. Různé změny v parametrech konvolučního jádra mají variabilitu přesnosti od 81 % až po 84 % (výsledky byly průměrovány pro více běhů). Určit správnou kombinaci je však nesmírně časově náročné kvůli obrovskému množství parametrů. Nejlepší výsledek byl dosažen s velikostí jádra 100 v první konvoluční vrstvě a ve zbylých vrstvách postupně se snižující se hodnotou (50, 25, 10, 5, 3).

Počet filtrů je nastaven na hodnotu 2 ve vstupní vrstvě, 32, 64, 128, 512, 128 v následujících vrstvách a nakonec 64 v poslední vrstvě. Další testované hodnoty byly voleny víceméně náhodně a bylo sledováno, zda dojde ke zlepšení výstupu. Tento proces se ovšem ukázal jako časově náročný a výsledná změna v přesnosti sítě nebyla natolik významná, aby se tyto hodnoty dále optimalizovaly.

Taktéž byly různě měněny počty plně propojených vrstev a jejich neuronů, ale nebyly znamenány lepší výsledky, takže jejich počet a parametry ve výsledné síti byly ponechány na původních hodnotách. Pro model_3 jsou to 3 plně propojené vrstvy s počty neuronů 128, 64 a 3 a pro model_4 jsou to 4 plně propojené vrstvy s hodnotami 128, 128, 64 a 4.

5.3.5 Epochy a batch size

Nejprve bylo zvoleno 12 učicích epoch, postupně byl jejich počet zvyšován. Dosažená přesnost při 20 učicích epochách byla v průměru 83.23 %. Od počtu 20 učicích epoch docházelo k přeučování, proto byla nakonec zvolena hodnota 15 jako kompromis mezi přesností a tendencí sítě k přeučování. Výsledky pro model_4 byly velmi podobné, nicméně se zvyšujícím se počtem epoch byla zaznamenána tendence přeučovat se. Nakonec byl pro model_4 změněn počet epoch na 18. Více v Příloze G.

Jako výchozí hodnota *batch size* byla zvolena hodnota 60, testovány byly vybrané hodnoty v rozmezí 20 až 80. Pro velikost 20 byla přesnost velmi nízká (okolo 30 %), od velikosti 40 bylo zaznamenáno znatelné přeučení s přesností okolo 82 %, která pro dále rostoucí hodnotu *batch size* postupně mírně klesala až na 81.52 % pro model_3 a 81.02 % pro model_4. U modelu_4 je vyšší přesnost pro velikost *batch size* 20, a to 79.32 %. Nakonec byla pro model_3 zvolena hodnota *batch size* 80. Ukázalo se, že se vzrůstající velikostí *batch size* roste i celková přesnost klasifikátoru. Oproti modelu_3 bylo u modelu_4 potřeba zmenšit velikost dávky na 70, kvůli omezení paměti GPU. Více v Příloze H.

5.3.6 Počet vzorků a optimalizace nestejně dlouhých signálů

Jak již bylo popsáno v kapitole 4, původní dataset obsahuje signály o délce 2500 až 14000 vzorků a po rozšíření dat je nejdelší signál s 36000 vzorky při vzorkovací frekvenci 250 Hz.

Volba správné délky signálu byla složitým úkolem, kdy bylo třeba zohlednit mnoho faktorů, např. nutnost signály ořezat/doplnit, množství ořezané/doplněné informace u jednotlivých signálů aj. Například signály označené jako PAROX měly minimální délku okolo 10000 vzorků, pro jakoukoli kratší délku by se tedy musely ořezat, čímž by hrozila ztráta části důležité informace. Postupně byly otestovány hodnoty 2500, 5000, 8000, 10000, 12000 a 15000 vzorků při vzorkovací frekvenci 250 Hz, kdy první a čtvrtá hodnota jsou nejpočetnější délky signálů v datasetu. Z hlediska zachytu částí FS by bylo vhodné volit co nejvíce signálů o délce zhruba 12000 vzorků pro oba druhy CNN. Pro hodnoty 10000 vzorků je přesnost 81.62 %, nejvyšší dosažená přesnost je 82.17 % pro délku 8000 vzorků u modelu_3. Pro model_3 se ukázal jako nejvhodnější počet vzorků 10000 a počet 10000 signálů pro každou

z kategorií, kdy byly signály kategorie PAROX/PERM rozděleny proporčně, tedy 5000 záznamů od každé subkategorie. Po několika dalších bžích modelu_3 a vykreslení jeho matice na Obrázku 6.3 byly zvýšeny počty záznamů pro SR o 2000, ale přesnost pro rozeznání SR a OTHER se nijak výrazně nezlepšila. Taktéž bylo vyzkoušeno snížit počty FS a stejně proporčně snížit i počty ostatních kategorií s tím, že SR by byla nejvíce zastoupená, ale znamenalo to snížení celkové přesnosti o 2 %, proto bylo rozhodnuto o navrácení se k hodnotám používaným před touto změnou. Pro model_4 byla nejvyšší přesnost pro 12000 vzorků, a to 82.50 %. Ještě byly testovány datasety složené z různého počtu vzorků od dostupných kategorií. Nakonec bylo dosaženo nejlepších výsledků se 7000 vzorky pro kategorie SR a OTHER, 8000 vzorků pro kategorii PERM a 7200 vzorků pro PAROX. Pro poslední kategorii se jedná o maximální počet vzorků. Zvolená délka signálu ve vzorcích bylo 1200. Více v Příloze I. Proces doplnění příliš krátkých signálů byl popsán v podkapitole 4.2.2.

5.3.7 Rozdělení testovací a trénovací množiny a rozšíření databáze

V této podkapitole bude popsáno pouze testování zvolených parametrů rozdělení dat na testovací a trénovací množinu. Více informací ohledně samotného rozdělení na testovací a trénovací množinu je uvedeno v kapitole 4.

Původně byla data rozdělena v poměru 80 % trénovací a 20 % testovací. V průběhu hledání nejvhodnějších parametrů bylo vyzkoušeno několik variant, ale nejlepších výsledků bylo dosaženo pro poměr rozdělení 90 % trénovací a 10 % testovací množina. Jednalo se o zlepšení do 1 %. Také byl testován vliv filtrů na přesnost klasifikace, signál bez filtrování dosáhl hodnoty 82,45 % a výsledky pro zbylé varianty byly víceméně shodné okolo 80 %. Pokud není kategorie PERM/PAROX vybírána náhodně, ale skládá se ze stejného množství PAROX a PERM záznamů, pak je celková přesnost lehce vyšší. Pokud jsou obě kategorie mnohonásobně zastoupenější než ty zbylé, pak je celková přesnost okolo 86 %. Byl použit signál bez filtrace, s filtrace jak síťového rušení, tak nulové izolinie a pak každý filtr zvlášť. Přesnost pro model_4 pro různá rozdělení na testovací a trénovací množinu byla pro stejné parametry lepší o méně než 1 % a poměr rozdělení 90 % trénovací a 10 % se ukázal jako nejlepší. Pro různé druhy filtrace byl model_4 lepší pro všechny možnosti a opět se jednalo o zlepšení okolo 1 %. Více v Přílohách K a L.

Taktéž bylo testováno konkrétní množství záznamů v datasetu. Jelikož je však kategorie PAROX velmi málo zastoupená vzhledem k ostatním kategoriím, bylo potřeba jednotlivé kategorie vyrovnat. Jedna z nejjednodušších metod je *undersampling*, kdy byly počty více zastoupených kategorií náhodně zredukovány na počet vzorků nejméně zastoupené kategorie. Další variantou je *oversampling*, kdy jsou všechny

kategorie dat doplněny na množství nejvíce zastoupené kategorie. Zde se objevil problém s celkovou pamětí GPU, kdy byla nedostačující pro rozšíření dat na počty nejvíce zastoupené kategorie, takže byla nejméně zastoupená kategorie rozkopírována v trénovací množině třikrát až čtyřikrát, avšak nebylo dosaženo uspokojivé přesnosti. Jako další možnost rozšíření je augmentace dat popsaná v podkapitole 4.1. Taktéž byly implementovány dvě jednoduché funkce pro přidání impulsního a gaussovského šumu. Obě metody mají vyšší přesnost než předchozí zmiňované a při přidávání pouze gaussovského šumu je přesnost lehce větší než při přidání pouze impulsního. Přidání signálů augmentovaných oběma metodami má ze všech metod nejvyšší přesnost a u modelu_3 je o něco málo přesnější než u modelu_4. Jedná se o přesnosti 81.75 % pro model_3 a 82.50 % pro model_4. Více v Příloze M.

5.3.8 Doplnování signálu na požadovanou délku

V rámci semestrální práce byly vytvořeny dvě možnosti použití funkce na rozkopírování signálu a doplnění nul na konec signálu, později byly doplněny i možnosti přidání nul na začátek signálu a proporčně na začátek a na konec. Bylo zjištěno, že doplnování nul na konec je přesnější než rozkopírování signálu asi o 1 %. Jelikož prodlužování signálu probíhá nakopírováním začátku signálu na konec a doplněním na potřebnou délku, je možné, že tento přechod může způsobit nevhodné navázání signálu a dojde tak k vyhledávání nežádoucích příznaků v prodlužovaných signálech. Jedním z řešení by mohlo být rozkopírování signálu s ohledem na výskyt R kmitů a spojovat signál vždy pouze v tomto místě. Nově přidané možnosti přidávání nul (na začátek a proporčně) nicméně nebyly moc úspěšné, protože se jejich přesnost pohybovala okolo 81 %, nedošlo tedy k žádnému zlepšení. Hodnoty přesnosti všech vyzkoušených možností pro klasifikátor do 4 kategorií byly vyšší než pro klasifikátor do 3 kategorií, a to do 1 %. Nakonec byla zvolena jako výchozí možnost prodlužování nulami na konci pro obě CNN, nicméně metoda prodlužování lze měnit jedním z parametrů ve funkci `lengthen_signal()`. Více v Příloze J.

5.4 Výsledné modely

Na základě pozorování změn při různých hyperparametrech byly vytvořeny 2 sítě: model_3 a model_4. Konkrétní hodnoty parametrů a topologie těchto sítí jsou vidět na příložených Obrázcích 5.1 a 5.2. Výpisy vrstev obou CNN obsahují 3 sloupce. První sloupec určuje název vrstvy a její typ (např. název `dense_1` a typ `Dense`), druhý sloupec určuje výstupní rozměry matice a třetí je počet parametrů jednotlivých vrstev. V dolní části na výpisu vrstev je vypsán konečný počet parametrů. Počty parametrů vyšší než nula jsou pouze u vrstev, kde probíhá učení CNN.

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 9901, 2)	202
conv1d_1 (Conv1D)	(None, 9852, 32)	3232
max_pooling1d (MaxPooling1D)	(None, 2463, 32)	0
conv1d_2 (Conv1D)	(None, 2439, 64)	51264
max_pooling1d_1 (MaxPooling1D)	(None, 1219, 64)	0
conv1d_3 (Conv1D)	(None, 1210, 128)	82048
max_pooling1d_2 (MaxPooling1D)	(None, 605, 128)	0
conv1d_4 (Conv1D)	(None, 601, 512)	328192
max_pooling1d_3 (MaxPooling1D)	(None, 300, 512)	0
conv1d_5 (Conv1D)	(None, 298, 128)	196736
max_pooling1d_4 (MaxPooling1D)	(None, 149, 128)	0
conv1d_6 (Conv1D)	(None, 147, 64)	24640
max_pooling1d_5 (MaxPooling1D)	(None, 73, 64)	0
dropout (Dropout)	(None, 73, 64)	0
flatten (Flatten)	(None, 4672)	0
dense (Dense)	(None, 128)	598144
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 64)	8256
dropout_2 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 3)	195
=====		
Total params: 1,292,909		
Trainable params: 1,292,909		
Non-trainable params: 0		

Obr. 5.1: Model pro klasifikaci do 3 kategorií (model_3).

Layer (type)	Output Shape	Param #
conv1d_14 (Conv1D)	(None, 11951, 2)	102
conv1d_15 (Conv1D)	(None, 11902, 32)	3232
max_pooling1d_12 (MaxPooling1D)	(None, 2975, 32)	0
conv1d_16 (Conv1D)	(None, 2951, 64)	51264
max_pooling1d_13 (MaxPooling1D)	(None, 1475, 64)	0
conv1d_17 (Conv1D)	(None, 1466, 128)	82048
max_pooling1d_14 (MaxPooling1D)	(None, 733, 128)	0
conv1d_18 (Conv1D)	(None, 729, 512)	328192
max_pooling1d_15 (MaxPooling1D)	(None, 364, 512)	0
conv1d_19 (Conv1D)	(None, 362, 128)	196736
max_pooling1d_16 (MaxPooling1D)	(None, 181, 128)	0
conv1d_20 (Conv1D)	(None, 179, 64)	24640
max_pooling1d_17 (MaxPooling1D)	(None, 89, 64)	0
dropout_6 (Dropout)	(None, 89, 64)	0
flatten_2 (Flatten)	(None, 5696)	0
dense_7 (Dense)	(None, 128)	729216
dense_8 (Dense)	(None, 128)	16512
dropout_7 (Dropout)	(None, 128)	0
dense_9 (Dense)	(None, 64)	8256
dropout_8 (Dropout)	(None, 64)	0
dense_10 (Dense)	(None, 4)	260
=====		
Total params: 1,440,458		
Trainable params: 1,440,458		
Non-trainable params: 0		

Obr. 5.2: Model pro klasifikaci do 4 kategorií (model_4).

6 Výsledky

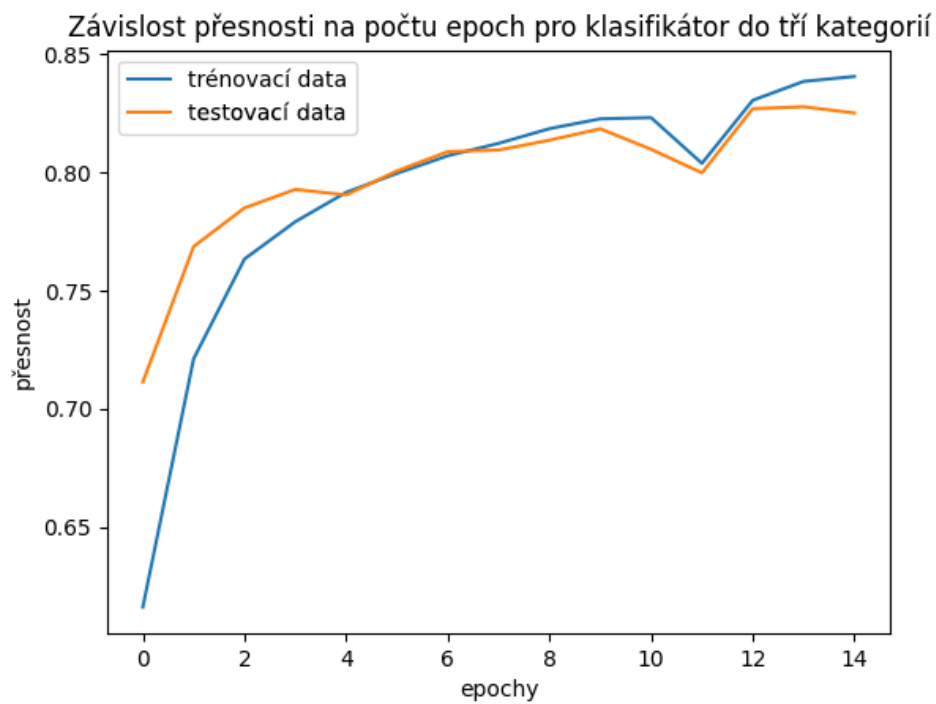
V této kapitole budou popsány výsledky obou vytvořených modelů, tedy model_3 a model_4. Jako metrika pro měření výsledků byla primárně použita přesnost, nicméně jsou zmíněny i ostatní obvykle používané metriky pro vyhodnocování úspěšnosti u klasifikátorů např. matice záměn a F1-skore.

Srovnání dvou odlišných modelů s různými parametry je velmi obtížné, proto bylo původně rozhodnuto o použití stejného datasetu a stejného CNN modelu. Taktéž byly pro oba modely použity stejné hyperparametry a počty jednotlivých vrstev. Následně proběhlo 10 běhů pro model_3 a model_4 a výsledky byly zprůměrovány. Ukázalo se, že nejsou skoro žádné rozdíly v přesnosti a oba modely dosáhly okolo 82 % a rozdílem maximálně do 1 %. Později byly hledány optimální parametry pro každý z modelů, kdy bylo zjištěno, že každému modelu vyhovuje pro nejlepší výsledky přesnosti jiná množina parametrů.

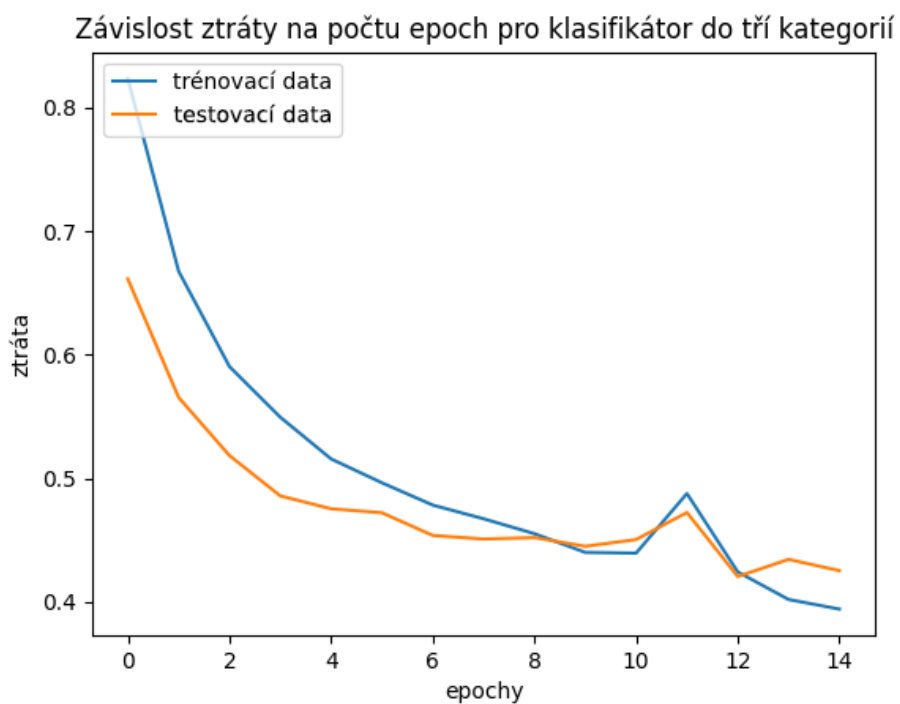
Pro každý z modelů bylo vyzkoušeno mnoho parametrů (viz 5.3) a hledání bylo zastaveno v případě nejlepší průměrné přesnosti. Dále byla uložena historie učebního procesu a ztráty (*loss*), která byla vykreslena do grafu. Taktéž byly uloženy hodnoty pro Sp, Se, PVV, přesnost a F1-skore. Nakonec byla vykreslena matice záměn. Prodloužení signálů kratších než zvolená délka bylo realizováno pomocí prodloužení signálu nulami na konec do požadované délky. Po deseti opakováních byla konečná celková přesnost vyhodnocena na 82.12 %. Byl zde použit model z Obrázku 5.1.

Jak je patrné z Obrázků 6.1 a 6.2, tak je pro trénovací data přesnost i ztráta vyšší než pro testovací data, protože je obvyklé, že neuronová síť hodnotí o něco hůře data, která jí nebyla ještě nikdy předložena. Nejedná se ovšem ještě o přetrénování, v takovém případě by byl rozdíl přesnosti a ztráty výrazně vyšší.

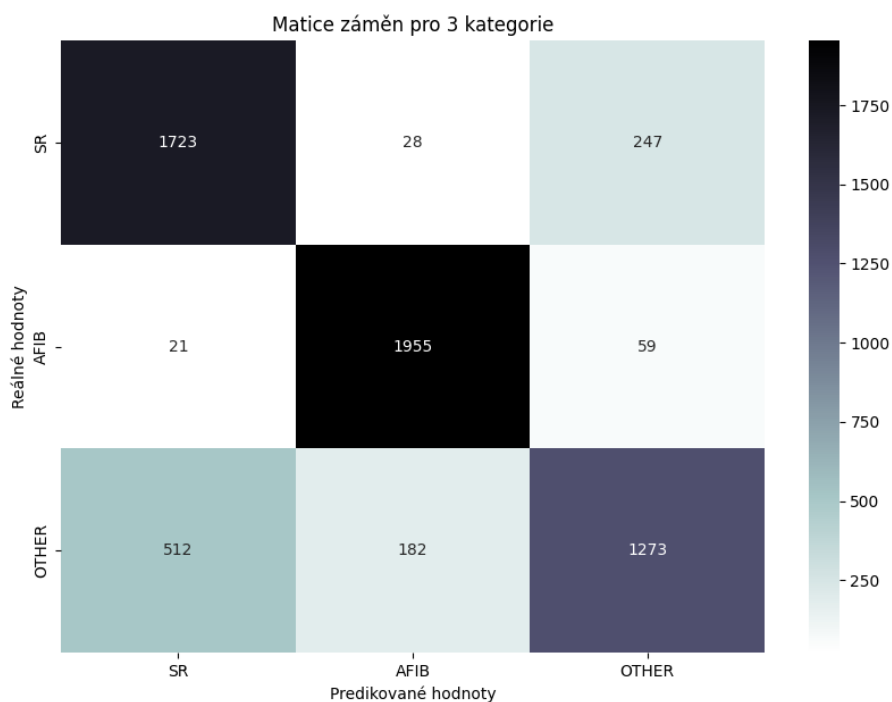
Dále bylo zjištěno, že neuronová síť velmi dobře klasifikuje záznamy s FS a záměna s kategorií SR je jen v desítkách případů, a to 21. Jen o něco horší výsledek s hodnotou 59 jsou pro záznamy s FS, které ale byly označeny jako OTHER. Tento model je poměrně dobře schopný ohodnotit i záznamy se SR a záměna s FS je opět pouze v desítkách případů, konkrétně 28. Oproti tomu 247 záznamů z kategorie SR bylo ohodnoceno jako ostatní patologie. Nejhorší výsledky jsou pro ostatní patologie, kdy jich bylo nejvíce chybně označeno jako SR, a to 512 záznamů. Dále bylo 182 záznamů predikováno jako FS.



Obr. 6.1: Graf závislosti přesnosti na počtu epoch pro model_3.



Obr. 6.2: Graf závislosti ztráty na počtu epoch pro model_3.



Obr. 6.3: Matice záměn pro tři kategorie.

V Tabulce 6.1 jsou zaznamenány hodnoty přesnosti, F1-skóre, senzitivity, specificity a PPV pro vyhodnocení úspěšnosti klasifikátoru. Jednotlivé hodnoty byly vypočteny z matice záměn (Obrázek 6.3). Přesnost je nejvyšší pro FS, a to 95.17 %, nejnižší pro ostatní patologie s 83.34 %. Dále je vidět velmi dobrá hodnota Se pro FS, což je patologie, která má být touto sítí primárně detekována. Nejnižší hodnota pro FS je u Sp a to 84.70 %. Obecně je u klasifikátorů detekujících nějaké onemocnění lepší, když je označen zdravý jedinec jako FP než jako FN, protože to hodnotitele obvykle donutí tento konkrétní případ více prozkoumat.

Hlavním zdrojem chyb se zdá být záměna ostatních patologií za SR, což může být způsobeno tím, že se v databázi nacházejí i záznamy s více labely, jako například SR a MI (záznam se sinusovým rytmem a infarktem myokardu). Právě tyto záznamy se zdají být příčinou špatného zařazení, jelikož se vizuálně zdají být velmi podobné záznamům se SR a navíc R reálně obsahují. Dále se SR vyskytuje v databázi velmi často i s bradykardií, tachykardií nebo atrioventrikulární bloádou prvního stupně, což může být dalším zdrojem FP záznamů. Tyto patologie bývají velmi často hodnoceny hodnotitelem subjektivně v rámci širšího kontextu, např. z více dostupných svodů nebo lékařské historie pacienta.

	SR	PERM/PAROX	OSTATNÍ
ACC	86,54	95,17	83,34
F1	81,01	93,06	71,80
SE	86,24	96,07	64,72
SP	86,68	84,70	92,41
PPV	76,37	90,30	80,62

Tab. 6.1: Výsledky pro klasifikátor do 3 kategorií.

Data u kategorie OTHER zaměněná za FS by mohla obsahovat flutter síní, který je vizuálně podobný s FS zejména kvůli fibrilačním vlnkám. V datasetu je nicméně zastoupeno jen málo dat s tímto labellem, takže se nezdá, že by tyto záznamy mohly nějak výrazně ovlivňovat výslednou preferenci pro kategorii PAROX/AFIB.

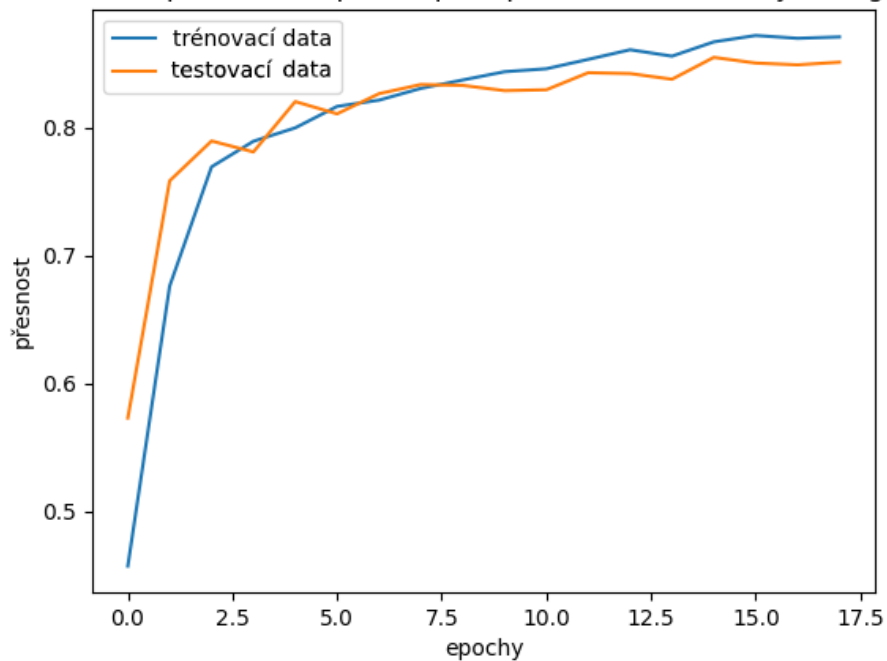
Stejně jako u modelu_3, i zde bylo opakováno 10 běhů a jejich výsledek zprůměrován. Bylo dosaženo přesnosti 85.14 %. Opět i zde bylo použito prodloužení signálů nulami na konci u příliš krátkých signálů. Byl zde použit původní model z Obrázku 5.2.

Stejně jako u modelu_3, tak je z Obrázků 6.4 a 6.5 patrné, že trénovací množina dosahuje lehce vyšší přesnosti než testovací data a opět zde není vidět výrazné přetrénování sítě.

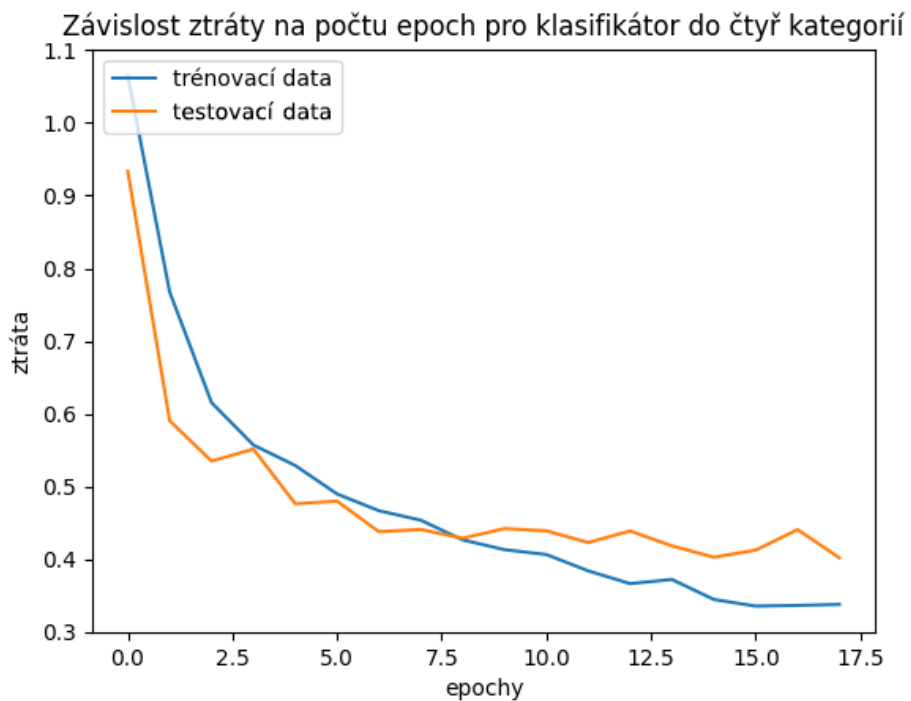
Model_4 zaměňoval kategorii SR nejčastěji s ostatními patologiemi stejně tak jako model_3, jednalo se o 131 případů. Nejméně FP záznamů pro kategorii SR je u kategorií PERM a PAROX, kdy podle očekávání byla záměna za PAROX poněkud častější, nicméně rozdíl jsou pouze 3 záznamy, takže se nezdá příliš relevantní. Kategorie PERM byla nejčastěji klasifikována správně a záměna za ostatní kategorie je v jednotkách vzorků. Konkrétně 1 záznam pro SR, 4 záznamy pro PAROX a 0 záznamů pro ostatní patologie.

Kategorie PAROX byla nejčastěji zaměněna za PERM a také za ostatní patologie, přičemž obojí bylo s velmi podobnou četností (24 a 22 záznamů). Zde se může jednat o chybu v anotaci dat, kdy ne vždy byly úplně zřejmé přechody mezi SR a paroxismem, ale i přesto byl záznam označen jako PAROX. Navzdory očekávání, že kategorie PAROX, zejména u záznamů s kratšími úseky FS, bude označena jako SR, byla označena jako SR pouze v jednotkách případů.

Závislost přesnosti na počtu epoch pro klasifikátor do čtyř kategorií

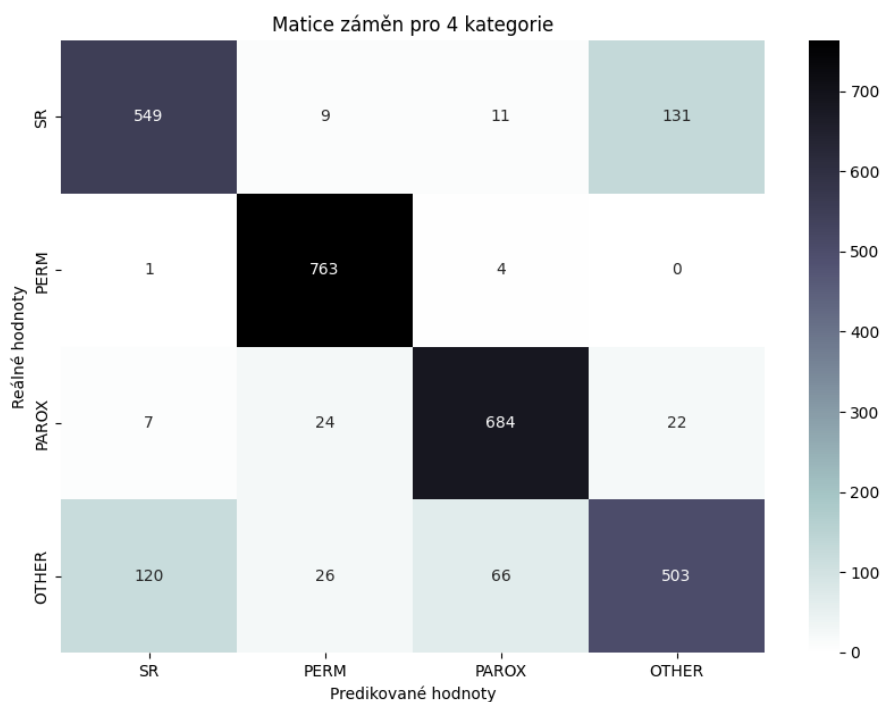


Obr. 6.4: Graf závislosti přesnosti na počtu epoch pro model_4.



Obr. 6.5: Graf závislosti ztráty na počtu epoch pro model_4.

Kategorie OTHER byla nejčastěji zaměňována se SR, a to ve 120 případech, dále byla poměrně často zaměňována i s kategorií PAROX a PERM. Důvod záměny u ostatních patologií za FS je pravděpodobně stejný jako u modelu_3 a jedná se o flutter síní. U FP záznamů ostatních patologií označených jako SR se pravděpodobně jedná opět stejnou záměnu jako u modelu_3 za záznamy s infarktem myokardu nebo s bradykardií. Pro třídy PERM a PAROX jsou záměny za ostatní patologie opět jen v desítkách případů (26 a 66), přičemž třída PAROX je zaměňována častěji než PERM. Pravděpodobně by se mohlo jednat o záznamy s obtížně filtrovatelným šumem zasahujícím do užitečného pásma signálu, který by mohl vypadat podobně jako fibrilační vlnky.



Obr. 6.6: Matice záměn pro 4 kategorie.

	SR	PERM	PAROX	OSTATNÍ
ACC	90,21	97,34	94,93	87,74
F1	79,54	94,74	91,25	72,55
SE	80,46	99,13	90,40	69,15
SP	93,23	96,86	96,81	93,43
PPV	78,64	90,73	92,12	76,29

Tab. 6.2: Výsledky pro klasifikátor do 3 kategorií.

V Tabulce 6.2 jde vidět hodnoty přesnosti, F1-skóre, senzitivity, specifity a PPV pro vyhodnocení úspěšnosti klasifikátoru. Jednotlivé hodnoty byly vypočteny z matice záměn 6.6. Nejvyšší výsledky v přesnosti dosáhly kategorie PERM s 97.34 % a PAROX s 94.93 % a nejhorší OTHER s 87.74 %. Kategorie SR i OTHER dosáhly vyšších hodnot zhruba o 4 % než při použití modelu_3. Nejvyšší Se i Sp měla třída PERM a to 99.13 % a 96.86 %. Hodnota pro Se u SR se v porovnání s modelem_3 zhoršila o asi 3 % a naopak Sp se zlepšila o asi 6 %. U třídy OTHER je Se lepší o zhruba 6 % a Sp se téměř nezměnila.

7 Závěr

Cílem této bakalářské práce bylo navrhnout, vytvořit a analyzovat učební proces dvou klasifikátorů, a to klasifikátoru pro rozpoznávání SR, AFIB a OTHER (v práci označovaný jako model_3) a klasifikátoru pro rozpoznávání SR, PERM, PAROX a OTHER (označovaný jako model_4).

V teoretické části práce bylo vytvořeno shrnutí lékařské podstaty fibrilace síně a dále také principy detekce fibrilace síně. Byly popsány jak metody bez využití umělé inteligence, tak s využitím neuronových sítí, především CNN. Je zde také uveden popis všech dat použitých ke klasifikaci včetně jejich základních statistických charakteristik.

V praktické části byly navrženy dvě neuronové sítě využívající 1D konvoluci (1D CNN). Jedna pro klasifikaci do 3 kategorií (PAROX/PERM, SR, OTHER) a druhá pro klasifikaci do 4 kategorií (PAROX, PERM, SR, OTHER). Bylo provedeno předzpracování dat, jejich filtrace a standardizace. Následně byla data rozdělena na testovací a trénovací množinu.

Dále byly analyzovány různé parametry ovlivňující proces učení pro oba modely. Jednalo se například o použití různých aktivačních funkcí, parametry dropout vrstev, druhy poolingů, počet epoch, počet *batch size*, délku vzorků, způsob prodlužování signálu, rozdělení testovací a trénovací množiny, použití filtrů, způsoby rozšíření datasetu aj. Pro každý z těchto parametrů byly testovány různé hodnoty a vybrány byly ty nejvhodnější pro konkrétní neuronovou síť. Bylo zjištěno, že výslednou přesností se pro většinu hodnot parametrů liší oba vytvořené modely jen minimálně.

Jako vyhodnocovací metrika byla pro obě CNN zvolena přesnost, jelikož byly v obou případech použity vyvážené datasety. Byly však také použity i jiné hodnotící metriky, např. matice záměn nebo F1-skore. Nejlepší dosažená přesnost pro model_3 je 82.12 % a pro model_4 je 85.14 %.

Výsledná přesnost, zejména v případě klasifikátoru do 4 kategorií, již může být dostatečná pro praktické využití. Klasifikátor nyní může dát alespoň přibližnou informaci, kterou patologii v záznamu hledat, například v situaci, kdy je třeba ručně anotovat dataset pro další práci. Na druhou stranu, dosažená přesnost je pro nasazení ve zdravotnictví (byť jen při využití klasifikátoru jako doplňkového nástroje) pravděpodobně nedostatečná.

Bylo by možné provést další úpravy obou CNN a odstranění několika hlavních problémů. Jedním z nich je záměna záznamů ze skupiny SR na OTHER a naopak. Možným řešením by mohlo být vybalancování výběru dat pro OTHER, kdy by se nejednalo o náhodný výběr z celé množiny všech ostatních patologií, ale o roztrídění jednotlivých záznamů podle patologie a poměrné vybírání z těchto roztríděných

záznamů. Taktéž by mohlo být přínosné prohlédnout záznamy s více patologiemi a záznam zařadit do skupiny podle nejvíce výrazné patologie.

Další možností by mohlo být vytvoření více kategorií, do kterých by byl záznam zařazen s určitou pravděpodobností, takže by se nejednalo o striktní zařazení záznamu do jedné z kategorií, ale záznam by měl různou míru příslušnosti k několika kategoriím (tzv. *multilabel* přístup). Také by bylo možné použít jinou architekturu sítě, například RNN nebo hybridní síť kombinující CNN a RNN. Dalším zajímavým řešením by bylo použití 2D CNN, kdy by byl jako vstup využit spektrogram jednotlivých záznamů.

Jako poměrně perspektivní se může jevit i využití dalších oblastí strojového učení a umělé inteligence. Pro hledání parametrů neuronových sítí by například šel využít genetický algoritmus. Parametry sítě by se zakódovaly do genomu a jako metrika (fitness funkce) by byla použita dosažená přesnost výsledné neuronové sítě s danými parametry. Tento přístup by ovšem byl časově náročný, i když by nevyžadoval tolik manuální práce.

Literatura

- [1] ČIHÁK, Robert, Petr HEINC, Luděk HAMAN, Martin FIALA, Petr NEUŽIL a Ondřej TOMAN. Fibrilace síní. *Cor et Vasa* [online]. 2011, 53(Suppl.), 27-52 [cit. 2023-05-26]. ISSN 00108650. Dostupné z: doi:10.33678/cor.2011.193
- [2] ZONI-BERISSO, Massimo, Fabrizio LERCARI, Tiziana CARAZZA a Stefano DOMENICUCCI. Epidemiology of atrial fibrillation: European perspective. *Clinical Epidemiology* [online]. [cit. 2023-05-26]. ISSN 1179-1349. Dostupné z: doi:10.2147/CLEP.S47385
- [3] RIENSTRA, Michiel, Steven A. LUBITZ, Saagar MAHIDA, et al. Symptoms and Functional Status of Patients With Atrial Fibrillation. *Circulation* [online]. 2012, 125(23), 2933-2943 [cit. 2023-05-26]. ISSN 0009-7322. Dostupné z: doi:10.1161/CIRCULATIONAHA.111.069450
- [4] ACLS MEDICAL TRAINING. Atrial fibrillation - ACLS medical training. In: *ACLS Medical Training* [online]. 22. 9. 2021. Available at: <https://www.aclsmedicaltraining.com/atrial-fibrillation/>
- [5] HAMPTON, John R. a Joanna HAMPTON. *EKG stručně, jasně, přehledně*. Přeložil Leoš LANDA. Praha: Grada Publishing, 2022. ISBN 978-80-271-1317-0.
- [6] HABERL, Ralph. *EKG do kapsy*. Praha: Grada, 2012. ISBN 978-80-247-4192-5.
- [7] PAN, Xian-Ming. Multiple linear regression for protein secondary structure prediction. *Proteins: Structure, Function, and Genetics* [online]. 2001, 2001-05-15, 43(3), 256-259 [cit. 2023-08-02]. ISSN 0887-3585. Dostupné z: doi:10.1002/prot.1036
- [8] TACCHINO, Francesco, Chiara MACCHIAVELLO, Dario GERACE a Daniele BAJONI. An artificial neuron implemented on an actual quantum processor. *Npj Quantum Information* [online]. 2019, 5(1) [cit. 2023-08-02]. ISSN 2056-6387. Dostupné z: doi:10.1038/s41534-019-0140-4
- [9] MÜLLER, Berndt a J. REINHARDT. *Neural networks: an introduction*. Corr. 2nd print. New York: Springer-Verlag, 1991. ISBN 0387523804.
- [10] WIDROW, B. a M.A. LEHR. 30 years of adaptive neural networks: perceptron, Madaline, and backpropagation. *Proceedings of the IEEE* [online]. 78(9), 1415-1442 [cit. 2022-12-8]. ISSN 00189219. Dostupné z: doi:10.1109/5.58323

- [11] NAUCK, D. et al. *Neuro-fuzzy Systems*. Chichester, England: John Wiley & Sons, 1997. ISBN 9780471971511.
- [12] KOHONEN, Teuvo. *Self-organizing maps*. 3rd ed. Berlin: Springer-Verlag, 2001, xx, 501 s. : il. ISBN 3-540-67921-9.
- [13] ROZENBERG, Grzegorz, Thomas BÄCK a Joost N. KOK, ed. *Handbook of Natural Computing* [online]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012 [cit. 2023-08-02]. ISBN 978-3-540-92909-3. Dostupné z: doi:10.1007/978-3-540-92910-9
- [14] MURAT, Fatma, Ferhat SADAK, Ozal YILDIRIM, et al. Review of Deep Learning-Based Atrial Fibrillation Detection Studies. *International Journal of Environmental Research and Public Health* [online]. 2021, 18(21) [cit. 2023-05-26]. ISSN 1660-4601. Dostupné z: doi:10.3390/ijerph182111302
- [15] GUPTA, Deepak, et al. (ed.). *Deep Learning for Medical Applications with Unique Data*. Academic Press, 2022, ISBN 978-0128241455.
- [16] TUČKOVÁ, Jana. *Vybrané aplikace umělých neuronových sítí při zpracování signálů*. Praha: České vysoké učení technické v Praze, 2009. ISBN 978-80-01-04229-8.
- [17] GRAUPE, Daniel. *Principles of artificial neural networks (3rd edition)*. 3. ed. Singapore, Singapore: World Scientific Publishing, 2013. ISBN 9789814522731.
- [18] GHOSH, Bhaskar et al. *An empirical analysis of generative adversarial network training times with varying batch sizes*. IEEE, 2020. ISBN 9781728196565.
- [19] BURKOV, Andriy. *The hundred-page machine learning book*. [Quebec, Canada]: Andriy Burkov, [2019]. ISBN 978-1999579500.
- [20] HU, Jie, Li SHEN a Gang SUN. Squeeze-and-Excitation Networks. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* [online]. IEEE, 2018, 2018, s. 7132-7141 [cit. 2023-05-26]. ISBN 978-1-5386-6420-9. Dostupné z: doi:10.1109/CVPR.2018.00745
- [21] MAEDA-GUTIÉRREZ, Valeria et al. Comparison of convolutional neural network architectures for classification of tomato plant diseases. *Applied sciences (Basel, Switzerland)*. 2020, vol. 10, no. 4, p. 1245. ISSN 2076-3417. DOI: 10.3390/app10041245
- [22] VENKATESAN, Ragav; LI, Baoxin. *Convolutional neural networks in visual computing: a concise guide*. CRC Press, 2017.

- [23] DERTAT, Arden. *Applied Deep Learning - Part 4: Convolutional neural networks*. In: Towards Data Science [online]. 8. 11. 2017 [cit. 08.12.2022].
- [24] PAULS, Alexander and Josiah A. YODER. *Determining Optimum Drop-out Rate for Neural Networks*. In: Micsymposium.org [online] [cit. 08.12.2022].
- [25] WEI, Wenkang. *DeepLearning -2 DropOut* [online]. 2020 [cit. 08.12.2022]. 38
- [26] GOODFELLOW, Ian, Yoshua BENGIO a Aaron COURVILLE. *Deep learning*. Cambridge, MA: MIT press, [2016]. Adaptive computation and machine learning series. ISBN 978-0262035613.
- [27] THAKUR, Ayush. *Intuitive understanding of 1D, 2D, and 3D convolutions in convolutional neural networks*. Weights & Biases. [online]. 2020 [viewed 10 November 2022]. Available from: <https://wandb.ai/ayush-thakur/dl-question-bank/reports/Intuitive-understanding-of-1D-2D-and-3D-convolutions-in-convolutional-neural-networks—VmlldzoxOTk2MDA>
- [28] KRISHNAMURTHY, Kamesh; CAN, Tankut; SCHWAB, David J. Theory of gating in recurrent neural networks. *Physical Review X*, 2022, 12.1: 011011.
- [29] AGRAWAL, Rakshit, et al. Attention in recurrent neural networks for ransomware detection. *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2019. p. 3222-3226.
- [30] WANG, Lixiong, et al. Long short-term memory neural network with transfer learning and ensemble learning for remaining useful life prediction. *Sensors*, 2022, 22.15: 5744.
- [31] KENNEDY, Alan, et al. The accuracy of beat-interval based algorithms for detecting atrial fibrillation. *2015 Computing in Cardiology Conference (CinC)*. IEEE, 2015. p. 893-896.
- [32] RUAN, Xiuhua, et al. Automatic detection of atrial fibrillation using RR interval signal. *2011 4th International Conference on Biomedical Engineering and Informatics (BMEI)*. IEEE, 2011. p. 644-647.
- [33] ALAMATSAZ, Negin, et al. A lightweight hybrid CNN-LSTM model for ECG-based arrhythmia detection. arXiv preprint arXiv:2209.00988, 2022.
- [34] NAMDAR, Khashayar, Masoom A. HAIDER and Farzad KHALVATI. A modified AUC for training Convolutional neural networks: Taking confidence into account. *Frontiers in artificial intelligence*. 2021, vol. 4, p. 582928. ISSN 2624-8212. DOI: 10.3389/frai.2021.582928

- [35] ALI, Najat, Daniel NEAGU and Paul TRUNDLE. Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets. *SN Applied Sciences*. 2019, vol. 1, no. 12. ISSN 2523-3963. DOI: 10.1007/s42452-019-1356-9
- [36] WIKIPEDIA CONTRIBUTORS. Receiver operating characteristic. *Wikipedia, The Free Encyclopedia* [online]. 24. 10. 2022.
- [37] George B. moody PhysioNet Challenge. *George B. Moody PhysioNet Challenge*. [online] [cit. 08.12.2022].
- [38] TERESHCHENKO, Larisa G. and Mark E. JOSEPHSON. Frequency content and characteristics of ventricular conduction. *Journal of electrocardiology*. 2015, vol. 48, no. 6, pp. 933–937. ISSN 0022-0736. DOI: 10.1016/j.jelectrocard.2015.08.034
- [39] LENIS, Gustavo, et al. Comparison of baseline wander removal techniques considering the preservation of ST changes in the ischemic ECG: a simulation study. *Computational and mathematical methods in medicine*, 2017, 2017.
- [40] SÖRNMO, Leif; LAGUNA, Pablo. Bioelectrical signal processing in cardiac and neurological applications. Academic press, 2005. 10.1016/B978-0-12-437552-9.X5000-4.
- [41] ŠNÁBL, Ivo. Standardizace dat. *Matematickabiologie.cz* [online] [cit. 08.12.2022].
- [42] FERRETTI, Jacopo, et al. 1-D convolutional neural network for ECG arrhythmia classification. *Progresses in artificial intelligence and neural systems*, 2021, 269-279.
- [43] LOPEZ-DEL RIO, Angela, Maria MARTIN, Alexandre PERERA-LLUNA a Rabie SAIDI. *Effect of sequence padding on the performance of deep learning models in archaeal protein functional prediction*. *Scientific Reports* [online]. 2020, 10(1) [cit. 2023-04-28]. ISSN 2045-2322. Dostupné z: doi:10.1038/s41598-020-71450-8
- [44] DWARAMPUDI, Mahidhar; REDDY, N. V. Effects of padding on LSTMs and CNNs. arXiv preprint arXiv:1903.07288, 2019.
- [45] CAO, Ping, Xinyi LI, Kedong MAO, Fei LU, Gangmin NING, Luping FANG a Qing PAN. A novel data augmentation method to enhance deep neural networks for detection of atrial fibrillation. *Biomedical Signal Processing and Control* [online]. 2020, 56 [cit. 2023-08-02]. ISSN 17468094. Dostupné z: doi:10.1016/j.bspc.2019.101675

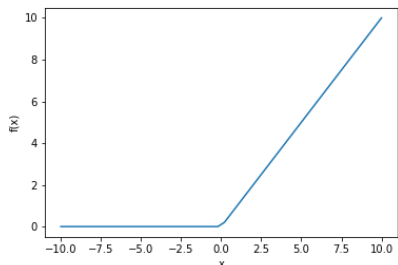
- [46] MOUSAVI, Sajad, Fatemeh AFGHAH a U. Rajendra ACHARYA. HAN-ECG: An interpretable atrial fibrillation detection model using hierarchical attention networks. *Computers in Biology and Medicine* [online]. 2020, 127 [cit. 2023-08-02]. ISSN 00104825. Dostupné z: doi:10.1016/j.combiomed.2020.104057
- [47] NGUYEN, Quang H., Binh P. NGUYEN, Trung B. NGUYEN, Trang T.T. DO, James F. MBINTA a Colin R. SIMPSON. Stacking segment-based CNN with SVM for recognition of atrial fibrillation from single-lead ECG recordings. *Biomedical Signal Processing and Control* [online]. 2021, 68 [cit. 2023-08-02]. ISSN 17468094. Dostupné z: doi:10.1016/j.bspc.2021.102672
- [48] WANG, Jibin. Automated detection of atrial fibrillation and atrial flutter in ECG signals based on convolutional and improved Elman neural network. *Knowledge-Based Systems* [online]. 2020, 193 [cit. 2023-08-02]. ISSN 09507051. Dostupné z: doi:10.1016/j.knosys.2019.105446
- [49] FAN, Xiaomao, Zhejing HU, Ruxin WANG, Liyan YIN, Ye LI a Yunpeng CAI. A novel hybrid network of fusing rhythmic and morphological features for atrial fibrillation detection on mobile ECG signals. *Neural Computing and Applications* [online]. 2020, 32(12), 8101-8113 [cit. 2023-08-02]. ISSN 0941-0643. Dostupné z: doi:10.1007/s00521-019-04318-2
- [50] FAN, Xiaomao, Qihang YAO, Yunpeng CAI, Fen MIAO, Fangmin SUN a Ye LI. Multiscaled Fusion of Deep Convolutional Neural Networks for Screening Atrial Fibrillation From Single Lead Short ECG Recordings. *IEEE Journal of Biomedical and Health Informatics* [online]. 2018, 22(6), 1744-1753 [cit. 2023-08-02]. ISSN 2168-2194. Dostupné z: doi:10.1109/JBHI.2018.2858789
- [51] ELIZAR, Elizar, Mohd Asyraf ZULKIFLEY, Rusdha MUHARAR, Mohd Hairi Mohd ZAMAN a Seri Mastura MUSTAZA. A Review on Multiscale-Deep-Learning Applications. *Sensors* [online]. 2022, 22(19) [cit. 2023-08-02]. ISSN 1424-8220. Dostupné z: doi:10.3390/s22197384
- [52] CAI, Wenjuan, Yundai CHEN, Jun GUO, et al. Accurate detection of atrial fibrillation from 12-lead ECG using deep neural network. *Computers in Biology and Medicine* [online]. 2020, 116 [cit. 2023-08-02]. ISSN 00104825. Dostupné z: doi:10.1016/j.combiomed.2019.103378
- [53] CRESWELL, Antonia, Tom WHITE, Vincent DUMOULIN, Kai ARULKUMARAN, Biswa SENGUPTA a Anil A. BHARATH. Generative Adversarial Networks: An Overview. *IEEE Signal Processing Magazine* [online]. 2018, 35(1), 53-65 [cit. 2023-08-02]. ISSN 1053-5888. Dostupné z: doi:10.1109/MSP.2017.2765202

- [54] SZANDAŁA, T. Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks. In: Bhoi, A. K., Mallick, P. K., Liu, C.-M. a Balas, V. E., ed. *Bio-inspired Neurocomputing*. Singapore: Springer Singapore, 2021, s. 203–224. DOI: 10.1007/978-981-15-5495-7_11. ISBN 978-981-15-5495-7. Dostupné z: https://doi.org/10.1007/978-981-15-5495-7_11.
- [55] NWANKPA, Chigozie, et al. Activation functions: Comparison of trends in practice and research for deep learning. arXiv preprint arXiv:1811.03378, 2018.

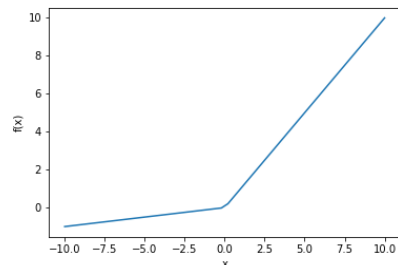
Seznam příloh

A Grafy vybraných aktivačních funkcí a jejich rovnice	65
B Věkové rozložení datasetu z MDT	67
C Přesnost pro vybrané aktivační funkce	69
D Rozdíl přesnosti pro různé počty dropout vrstev	71
E Rozdíl přesnosti pro různé parametry dropout vrstev	73
F Přesnost pro pooling metody	75
G Přesnost pro různé počty učících epoch	77
H Přesnost pro různou batch size	79
I Přesnost pro různé délky signálu	81
J Přesnost pro různé způsoby prodloužení signálu	83
K Přesnost pro různé rozdělení testovací a trénovací množiny	85
L Přesnost pro různé druhy filtrace	87
M Přesnost pro různé druhy rozšíření datasetu	89

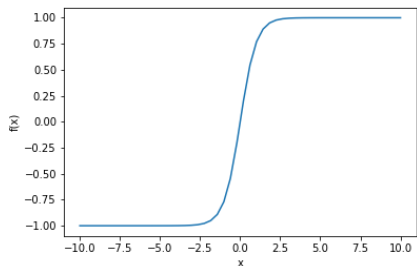
A Grafy vybraných aktivačních funkcí a jejich rovnice



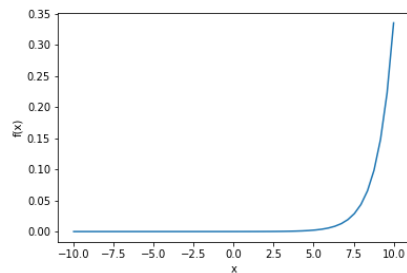
(a) $f(x) = \max(0, x)$



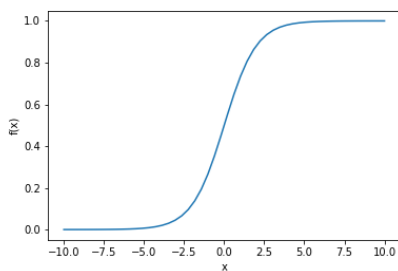
(b) $f(x) = \max(0.01, x)$



(a) $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

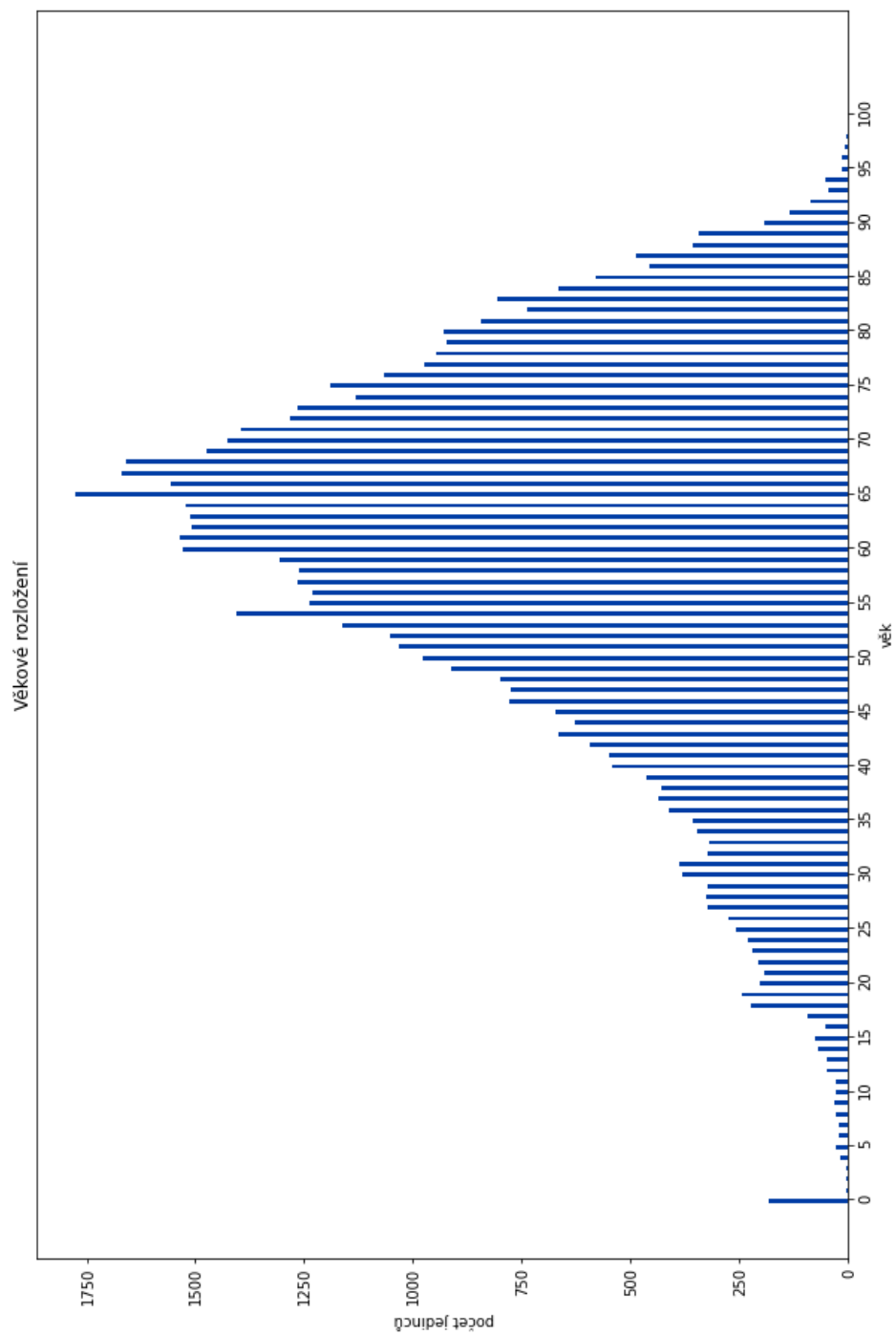


(b) $f(x) = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}}$

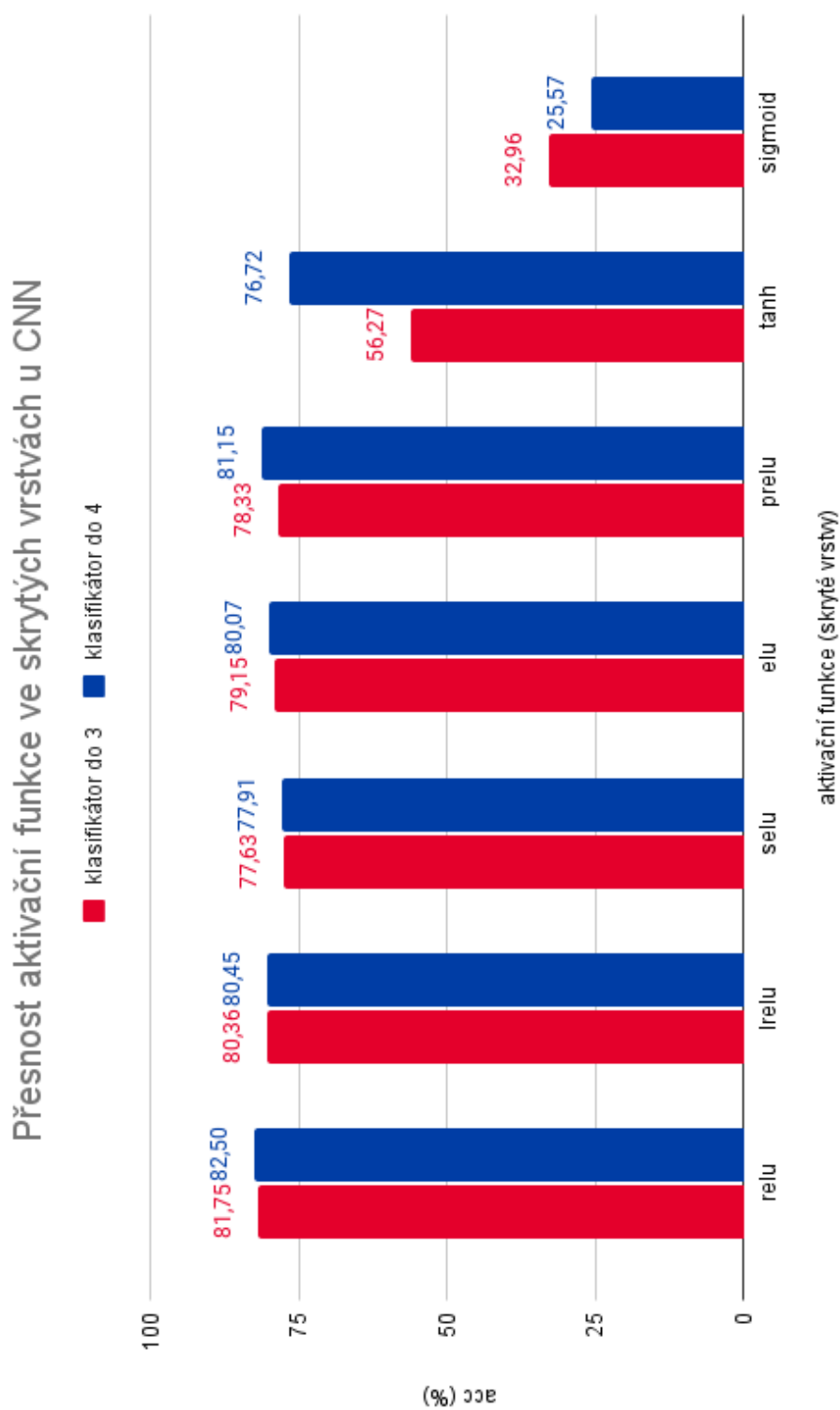


(a) $f(x) = \frac{1}{1 - e^{-x}}$

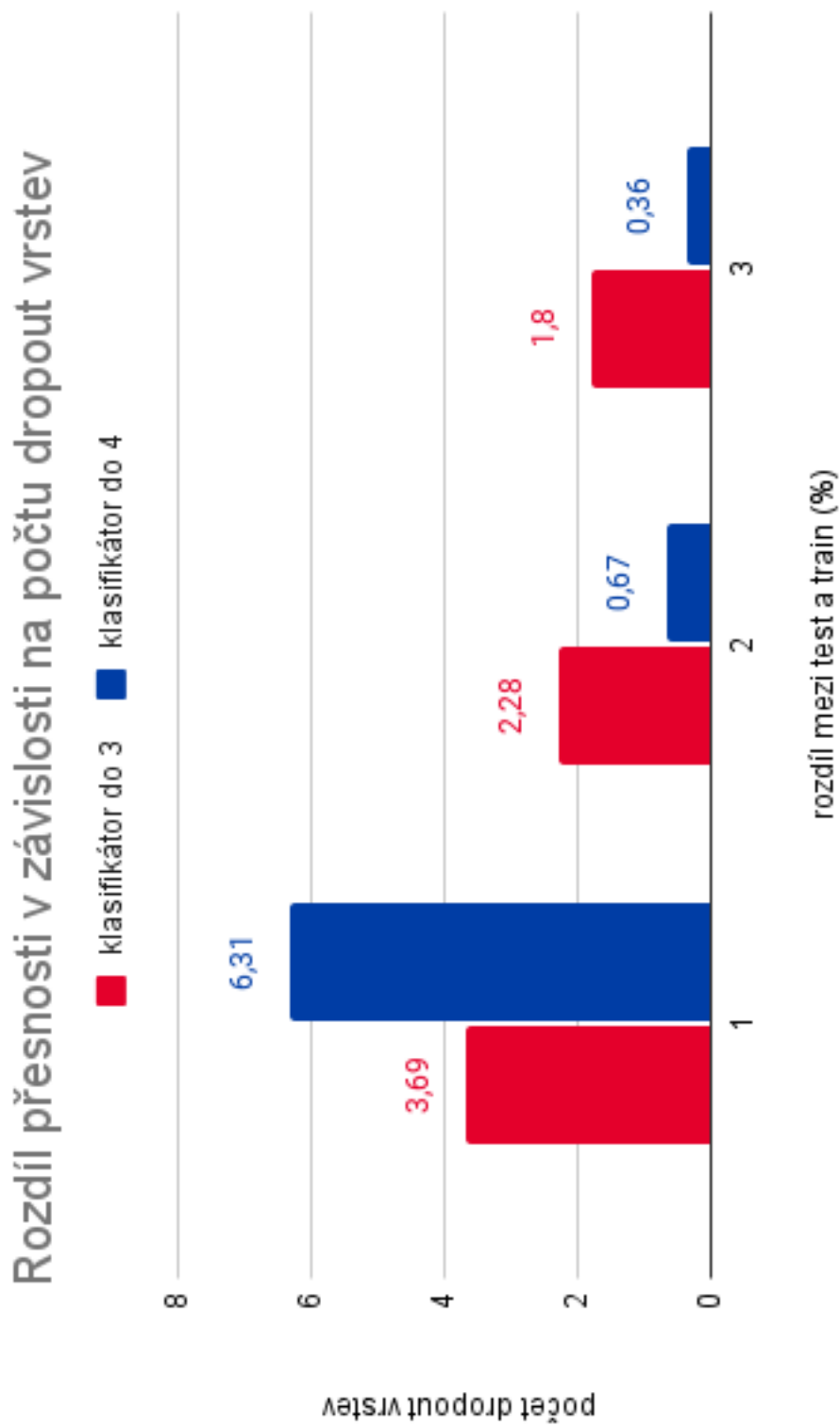
B Věkové rozložení datasetu z MDT



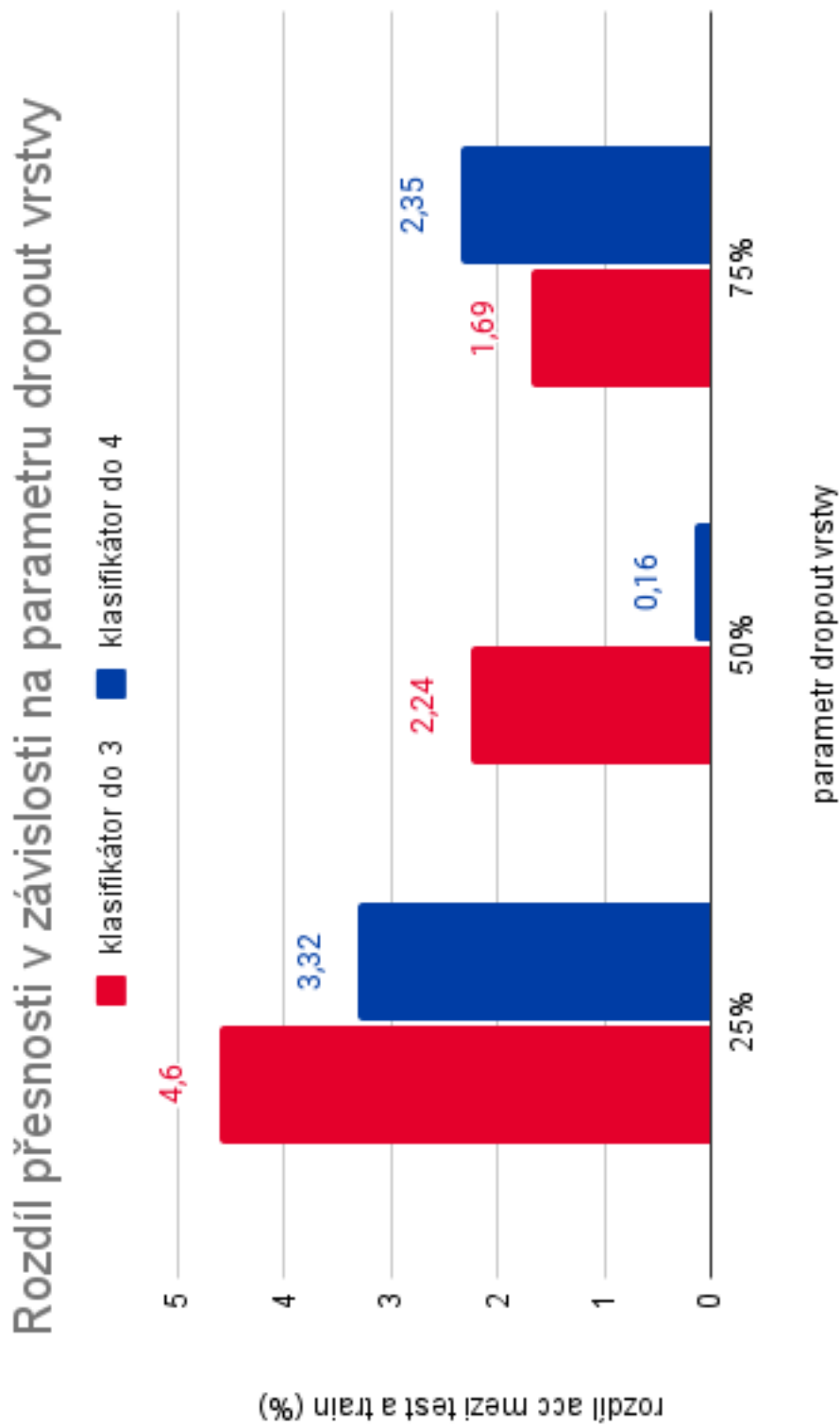
C Přesnost pro vybrané aktivační funkce



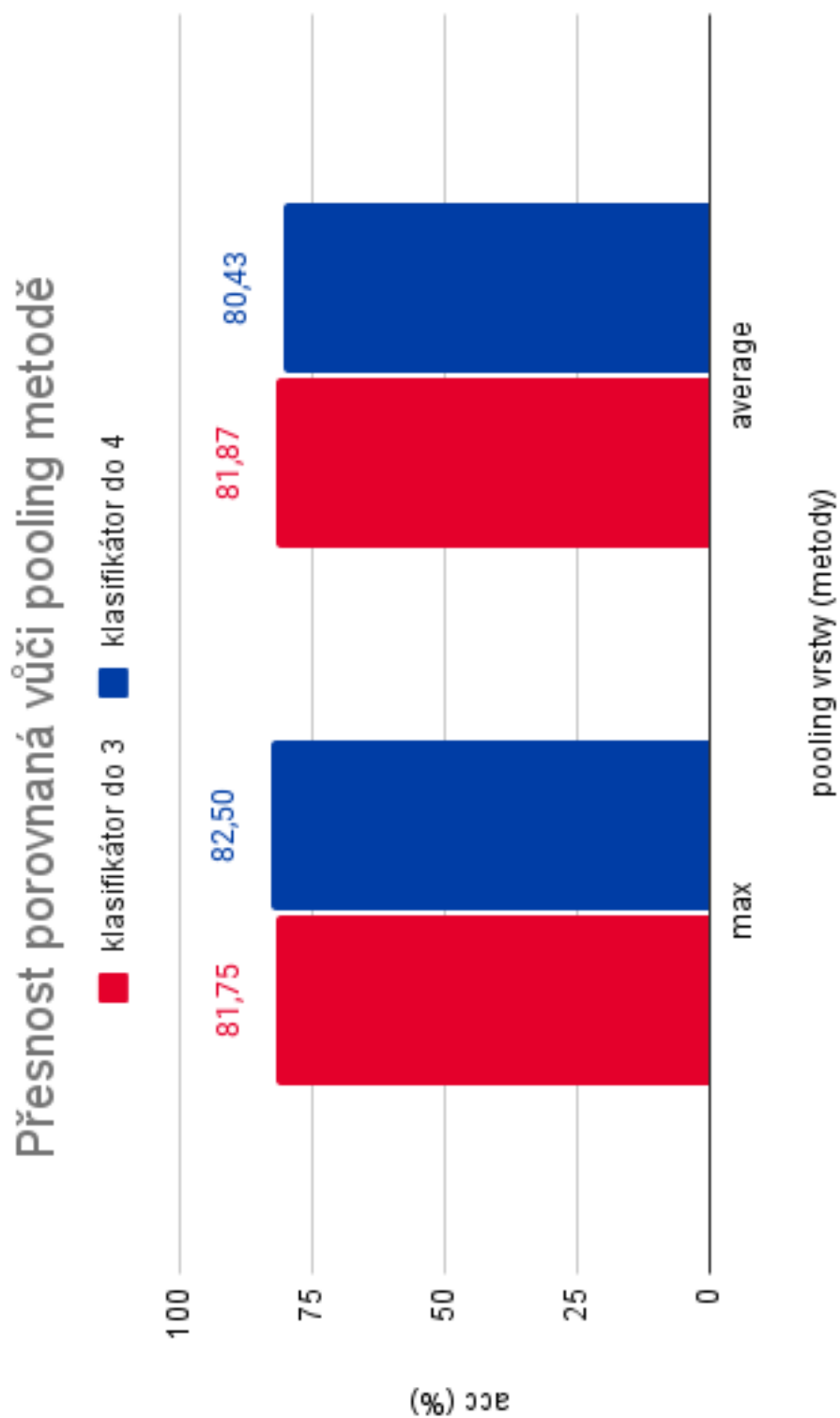
D Rozdíl přesnosti pro různé počty dropout vrstev



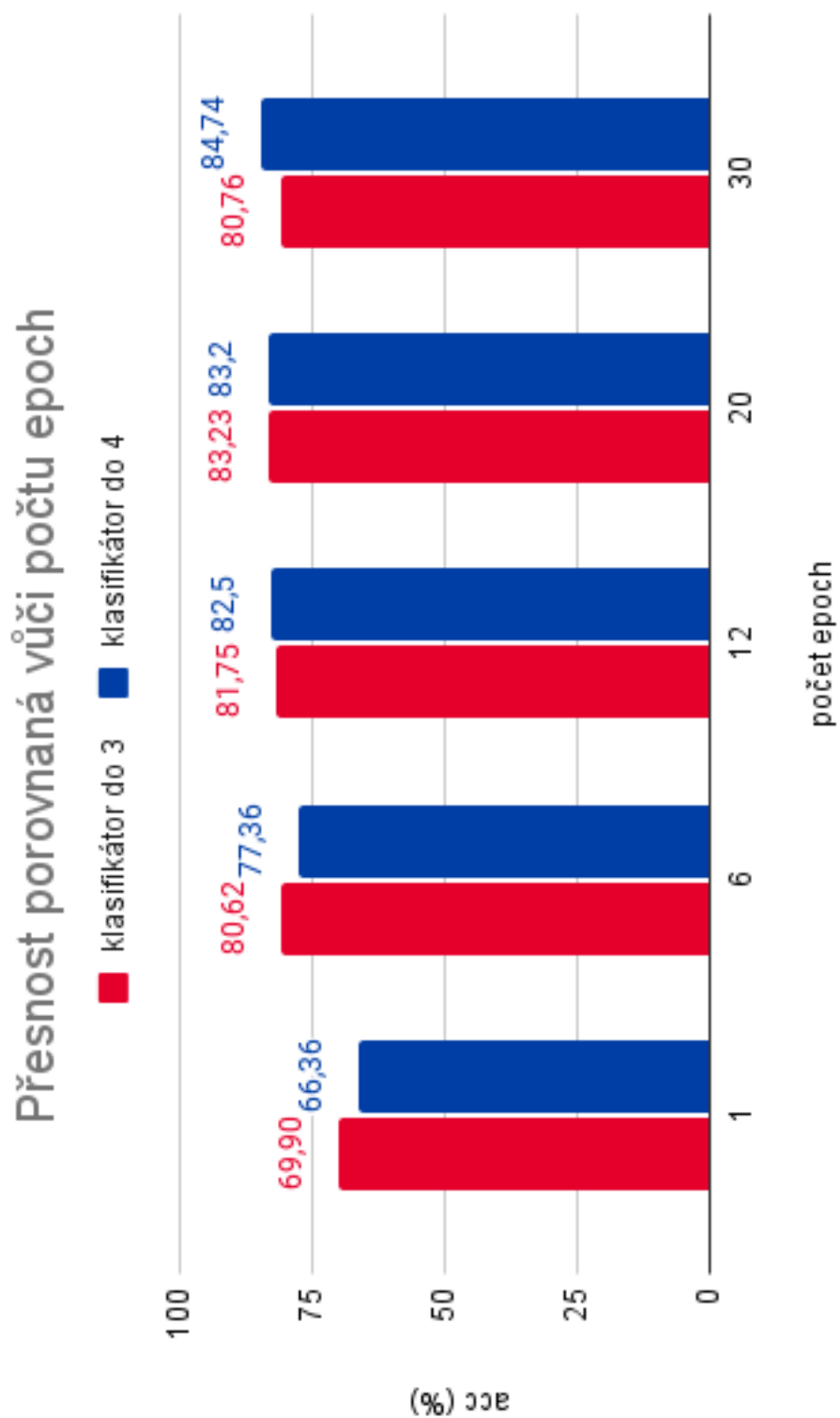
E Rozdíl přesnosti pro různé parametry dropout vrstev



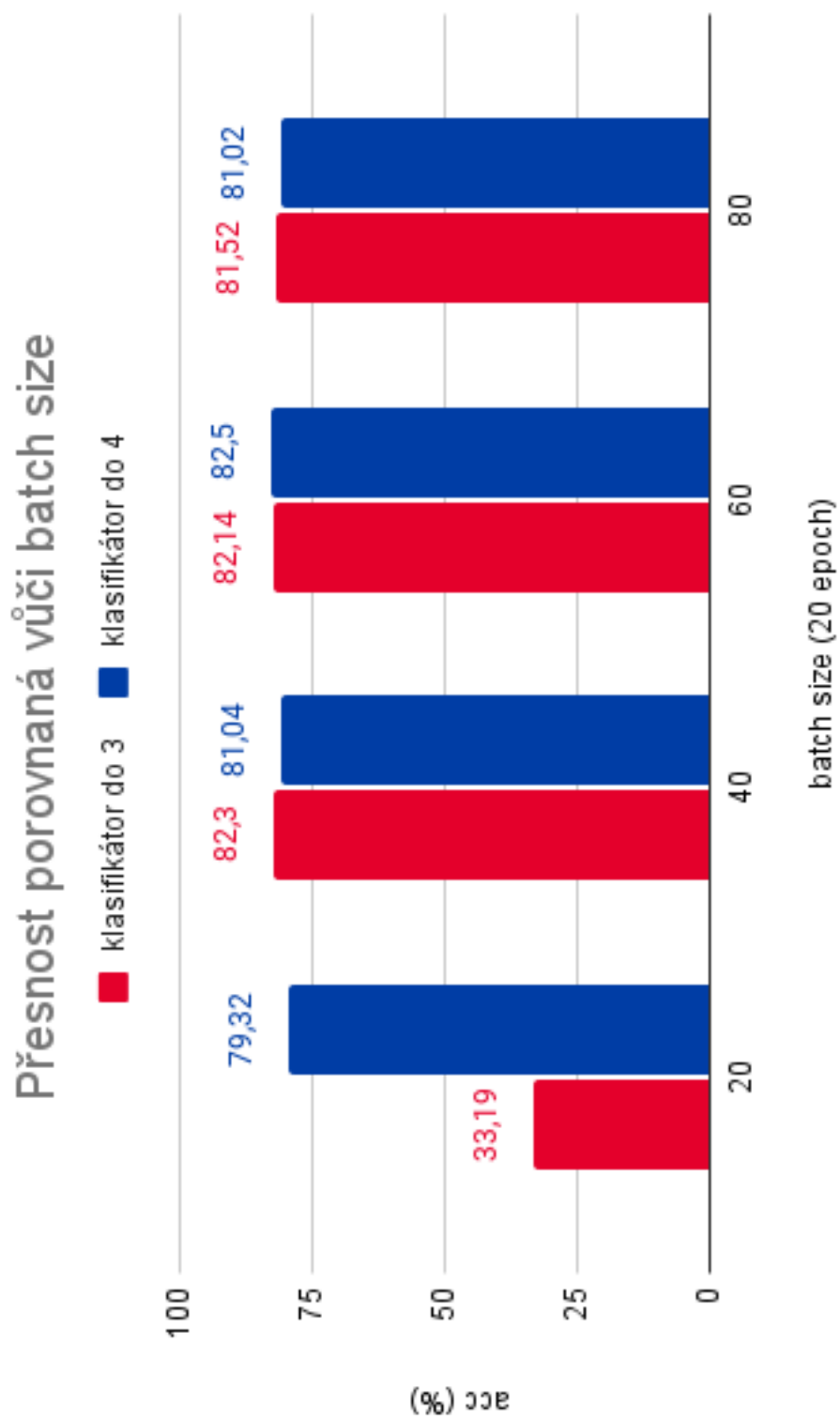
F Přesnost pro pooling metody



G Přesnost pro různé počty učících epoch

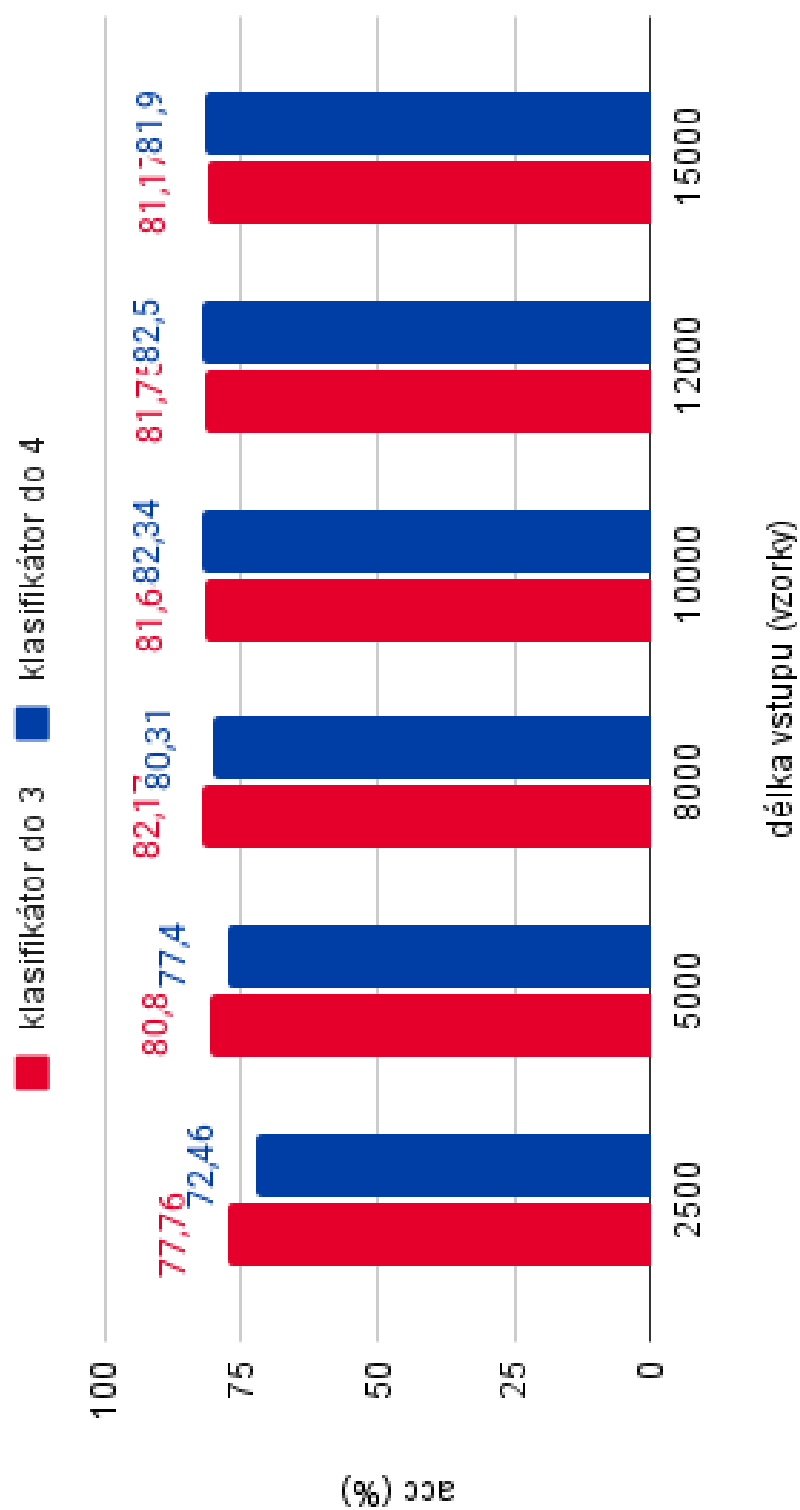


H Přesnost pro různou batch size

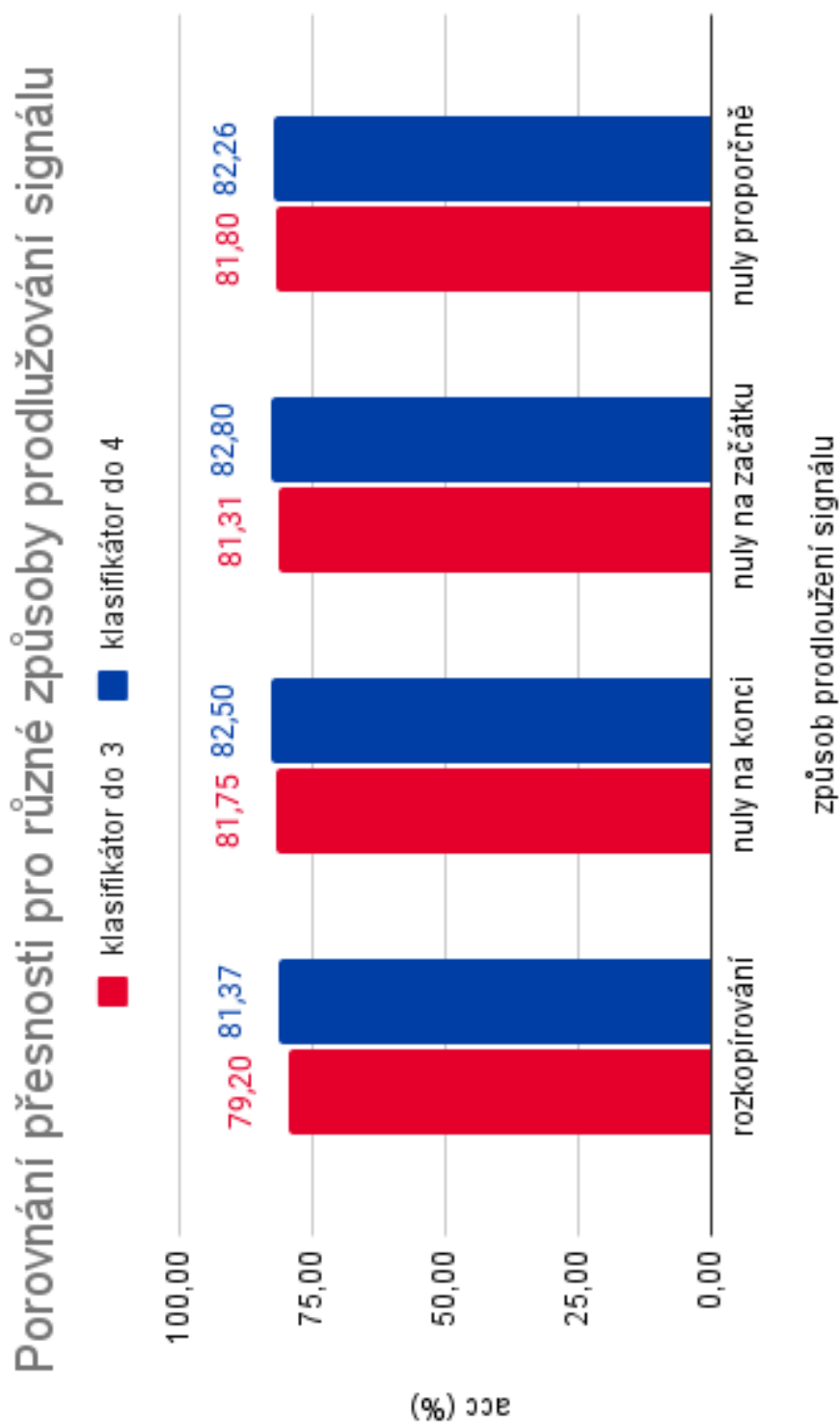


I Přesnost pro různé délky signálu

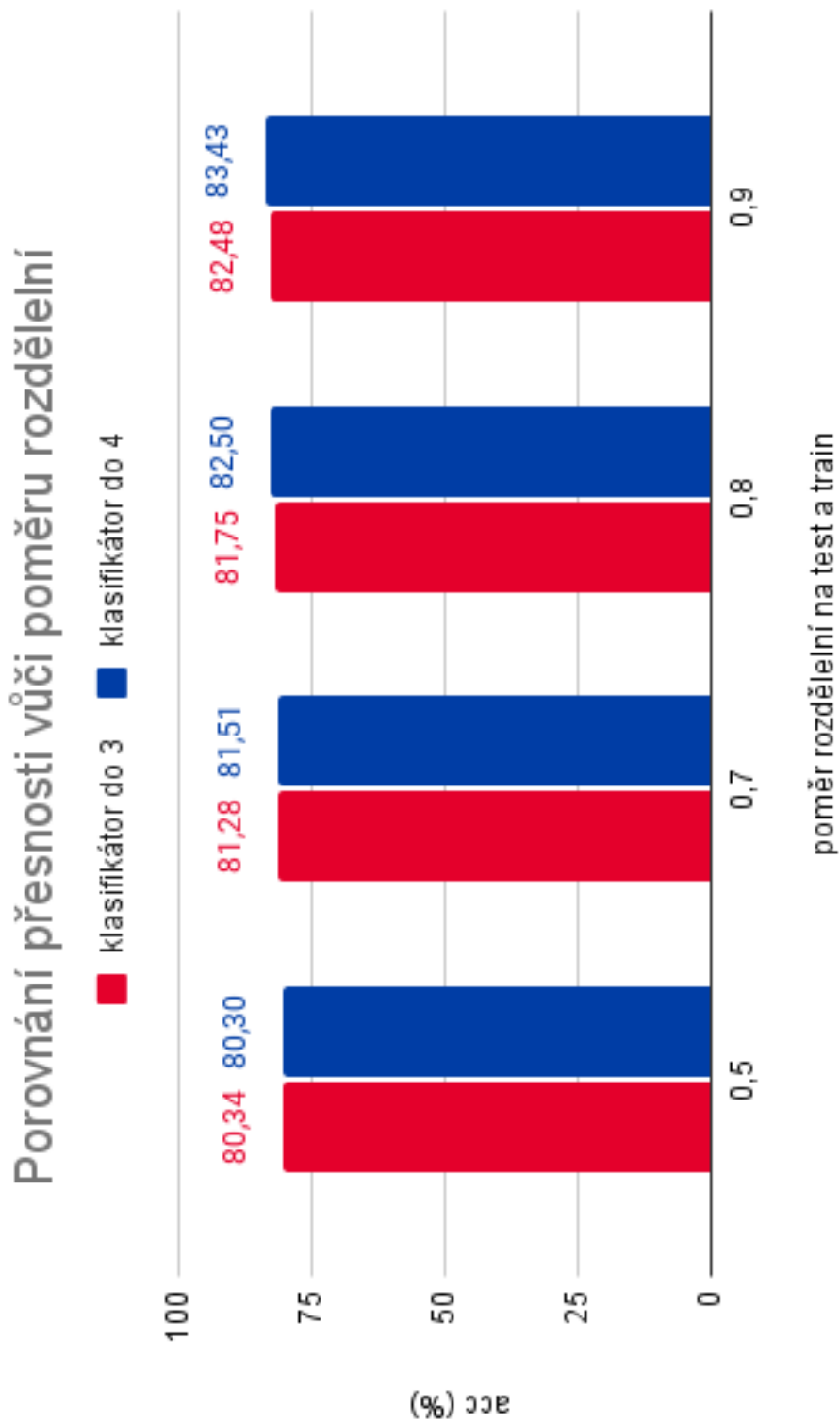
Porovnání přesnosti pro délku vstupu



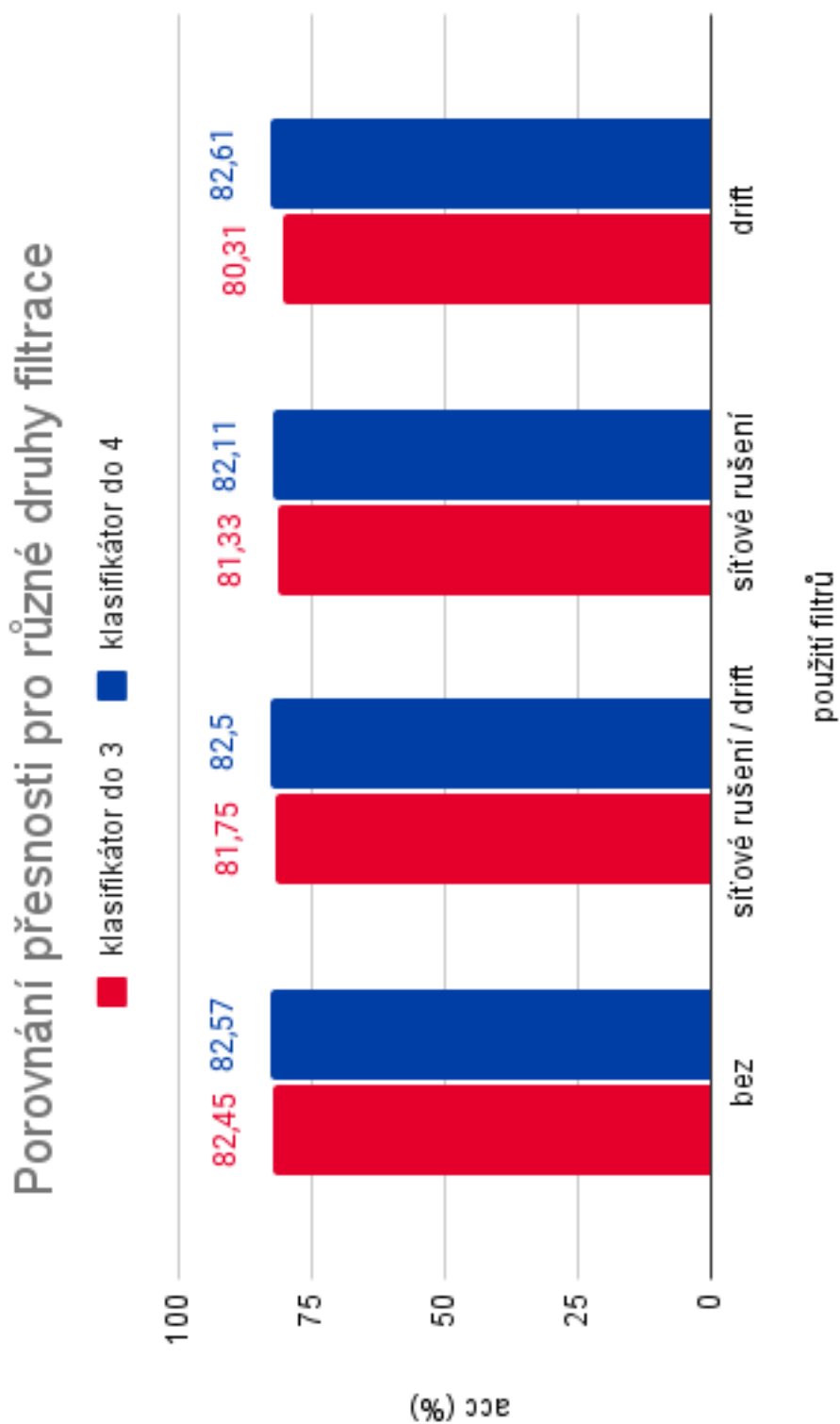
J Přesnost pro různé způsoby prodloužení signálu



K Přesnost pro různé rozdělení testovací a trénovací množiny



L Přesnost pro různé druhy filtrace



M Přesnost pro různé druhy rozšíření data- setu

