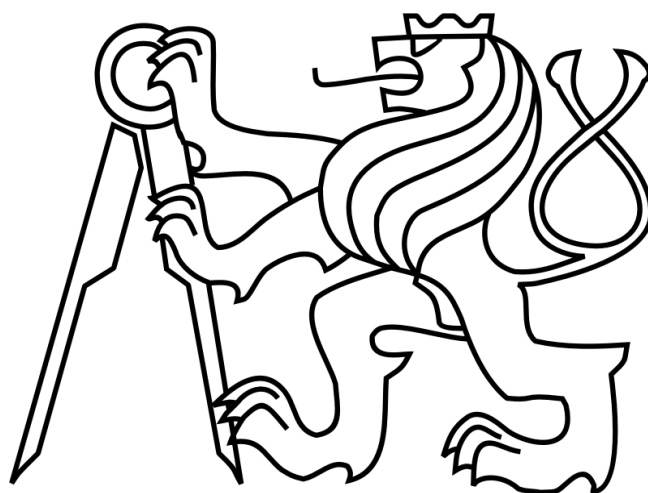


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ

V PRAZE

FAKULTA ELEKTROTECHNICKÁ

KATEDRA KYBERNETIKY



BAKALÁŘSKÁ PRÁCE

System pro selekci příznaků z EEG signálu

Vedoucí práce: Ing. Václav Gerla

2011

Ludmila Dohnalová

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Ludmila Dohnalová
Studijní program: Elektrotechnika a informatika (bakalářský), strukturovaný
Obor: Kybernetika a měření
Název tématu: Systém pro selekci příznaků z EEG signálu

Pokyny pro vypracování:

Cílem práce je tvorba systému, který bude schopen na základě klasifikovaného EEG signálu vybrat příznaky, které nejlépe korelují s danou klasifikací. Práce spočívá ve studiu dostupné literatury, týkající se selekce příznaků, na jejímž základě student zvolí algoritmy, které jsou nejlépe vhodné pro implementaci do navrhovaného systému. Je možno použít jak ověřené způsoby selekce, tak na základě vlastních úvah modifikovat či vytvořit algoritmy nové. Různé sady příznaků a odpovídající klasifikace budou studentovi poskytnuty. Práce bude realizována v prostředí programu Matlab.

Hlavní body zadání:

1. Seznámení se s problematikou extrakce příznaků z EEG signálu.
2. Studium literatury, týkající se selekce příznaků.
3. Implementace několika metod selekce.
4. Porovnání výsledků selekce nad reálnými EEG daty.

Seznam odborné literatury:

- [1] Liu, H.; Motoda, H.: Feature extraction, construction and selection: a data mining perspective. National University of Singapore, Singapore, 1998, ISBN 0-7923-8196-3.
- [2] Kotek, Z.; Mařík, V.; Hlaváč, V.; Psutka, J.; Zdráhal, Z.: Metody rozpoznávání a jejich aplikace. Academia, Praha, 1993.
- [3] Šnorek, M.; Jiřina, M.: Neuronové sítě a neuropočítače. Vydavatelství ČVUT, Praha, 1996.

Vedoucí bakalářské práce: Ing. Václav Gerla

Platnost zadání: do konce zimního semestru 2011/2012


prof. Ing. Vladimír Mařík, DrSc.
vedoucí katedry




prof. Ing. Boris Šimák, CSc.
děkan

V Praze dne 28. 2. 2011

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracovala samostatně a použila jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

V Praze dne 15.8.2011

Dolmalová

Podpis

Anotace

Tato práce se zabývá klasifikací spánku donošených novorozenců do dvou tříd – aktivního a klidného spánku. Vytvořený systém - soubor algoritmů a spustitelného kmenového souboru v MatLabu – pracuje s příznaky extrahovanými ze signálu EEG. Cílem práce bylo nalézt nejvhodnější algoritmus selekce příznaků, který by byl rychlý a přesný, a pokusit se sestavit sadu příznaků, která by byla obecně vhodná pro naši úlohu.

Použito bylo 6 algoritmů selekce příznaků (mRMR, SFS, SBS, SFFS, ReliefF a SBBS) a 4 různé klasifikátory (k-NN, SVM, Naive Bayes a C4.5), jejichž vzájemné kombinace jsme porovnávali. Klasifikační chyba byla určena porovnáním s hodnocením odborníka. Z těchto selekcí byl také určen optimální počet příznaků - 20. Jako nejlepší kombinace algoritmus selekce – klasifikátor nám vyšla kombinace algoritmu SFFS a klasifikátoru Naive Bayes.

Celkem jsme provedli 160 různých selekcí dvaceti příznaků, abychom určili četnost výběru jednotlivých příznaků. Dvacet nejčtetnějších příznaků, které tvořily především příznaky získané vlnkovou (Wavelet) transformací, jsme následně ověřili na nezávislém datasetu, který nebyl použit pro jejich výběr, jako sadu příznaků vhodnou pro klasifikaci novorozeneckého spánku.

Annotation

This thesis deals with neonatal sleep classification to 2 classes – active and quiet sleep. The developed system – a set of algorithms and the executable root file in MatLab – works with features extracted from the EEG signal. The goal was to find the best algorithm that would be quick and precise and to create a set of features sufficient for our task.

Combinations of 6 different feature selection algorithms (mRMR, SFS, SBS, SFFS, ReliefF and SBBS) and 4 classifiers (k-NN, SVM, Naive Bayes and C4.5) were compared. The optimal number of features based on these selections was determined. The best combination was found out as the combination of SFFS algorithm and Naive Bayes classifier. All classification results were compared with an expert's evaluation.

Altogether 160 different selections of 20 features were realized to determine their frequency of selection. The 20 most frequent features as a set of features proper to neonatal sleep classification, mostly the features computed with Wavelet transformation, were verified on an independent dataset that wasn't used for their selection.

Poděkování

Děkuji Ing. Václavu Gerlovi za rady a připomínky při zpracování teoretické i praktické části práce a za nalezení zajímavého tématu. Děkuji také Ústavu pro péči o matku a dítě za poskytnutí dat. Dále bych chtěla poděkovat Ing. Martinovi Macašovi za uvedení do problematiky klasifikace a validace a doc. Ing. Lence Lhotské, Csc. za uvedení do problematiky EEG. V poslední řadě bych chtěla poděkovat rodině, především své dceři Šárce, že mi umožnili práci dokončit.

Obsah

Seznam použitých zkratk	vii
1 Úvod	1
2 EEG	2
2.1 Úvod	2
2.2 Lidský mozek	2
2.3 EEG signál	3
2.3.1 Historie	3
2.3.2 Snímání EEG	4
2.3.3 Základní rytmy	5
2.3.4 Grafoelementy	6
2.3.5 Artefakty	7
2.3.6 Spánek	8
3 Zpracování signálu	12
3.1 Předzpracování	12
3.2 Segmentace	12
3.2.1 Konstantní segmentace	12
3.2.2 Adaptivní segmentace	13
3.3 Výpočet příznaků	14
3.4 Selektce příznaků	15
3.5 Klasifikace	15
3.5.1 Klasifikátory	16
3.5.2 Validace	20
4 Selektce příznaků	22
4.1 Metody snižování dimenze	22
4.2 Charakteristika selektce příznaků	23
4.2.1 Obecný algoritmus	24

4.3	Algoritmy selekce příznaků.....	25
4.3.1	SFS	25
4.3.2	SBS	26
4.3.3	FFS	26
4.3.4	B&BS	27
4.3.5	mRMR	28
4.3.6	Relief a ReliefF.....	29
5	Experimentální část.....	31
5.1	Použitý software	31
5.1.1	MatLab	31
5.1.2	WEKA	33
5.2	Data	34
5.3	Struktura experimentu	35
5.3.1	Výkonnostní křivky	35
5.3.2	Selekce zvoleného počtu příznaků	35
5.4	Výsledky a diskuze.....	36
5.4.1	Výkonnost algoritmů	36
5.4.2	Určení počtu příznaků	51
5.4.3	Porovnání selektovaných příznaků.....	54
5.4.4	Ověření na nezávislém datasetu	57
6	Závěr.....	61
7	Literatura	63
	Přílohy	67
	Příloha 1: Seznam příznaků.....	67
	Příloha 2: Ukázka kmenového souboru vytvořeného systému.....	73

Seznam použitých zkratek

ARFF	Attribute-Relation File Format – datový formát
ASU	Arizona State University – Arizonská univerzita
B&BS	Branch and Bound Search - algoritmus
C 4.5	Algoritmus rozhodovacího stromu
CNS	Centrální nervová soustava
CV	Cross-Validation
EEG	Elektroencefalografie, elektroencefalogram, elektroencefalograf
EKG	Elektrokardiografie, elektrokardiogram
EMG	Elektromyografie, elektromyogram
EOG	Elektrookulografie, elektrookulogram
FE	Feature Extraction – extrakce příznaků
FFT	Fast Fourier Transform – rychlá Fourierova transformace
FIR	Finite Impulse Response – filtr s konečnou impulsní odezvou
FS	Feature Selection – selekce příznaků
IBL	Instance Based Learning – učení založené na instancích
k-NN	k-Nearest Neighbour – k nejbližších sousedů
MatLab	Matrix Laboratory – software
mRMR	minimum Redundancy, Maximum Relevancy - algoritmus
NREM	Non Rapid Eyes Movement – viz REM
PCA	Principal Component Analysis – analýza hlavních komponent
PNG	Pneumografie, pneumograf
PRTools	Pattern Recognition Tools – toolbox MatLabu
PSG	Polysomnografie, polysomnograf
REM	Rapid Eyes Movement – spánková fáze (podle rychlých pohybů očí)
RMS	Root Mean Square – efektivní hodnota
SBS	Sequential Backward Search – zpětné hledání
SFFS	Sequential Forward Floating Search - algoritmus
SFS	Sequential Forward Search – dopřední hledání
SVM	Support Vector Machine – podpůrné vektorové stroje
WEKA	Waikato Environment for Knowledge Analysis - software

1 Úvod

Počítače hrají stále důležitější úlohu v mnoha odvětvích dnešního světa. Výjimkou není ani oblast zpracování signálu. Správné vyhodnocení biologických signálů je stejně důležité jako bezchybné provedení operace. Ačkoli se lékaři odborníci stále spoléhají hlavně na sebe, stále častěji sahají po počítači jako pomocníkovi.

Elektroencefalogram – záznam elektrické aktivity mozku – nalézá široké uplatnění v diagnostice. Spánkové záznamy dospělých osob jsou dnes s vysokou přesností automaticky zpracovávány počítači, které dokážou rozpoznat jednotlivé spánkové fáze. [1] Větší obtíž představuje zpracování spánkových záznamů dětí a novorozenců. Znalost poměru zastoupení jednotlivých fází spánku novorozence napomáhá neonatologům určit, zda se správně vyvíjí jeho mozek. [2] Systém, který by dokázal s přiměřenou přesností určit, v jaké fázi spánku se pacient právě nachází, by pomohl zrychlit a usnadnit analýzu těchto záznamů.

Tato práce se zabývá klasifikací spánku novorozenců na základě příznaků extrahovaných ze signálu EEG. Cílem práce je nalézt nejvhodnější algoritmus selekce příznaků a ověřit, zda existuje sada příznaků, která by byla obecně vhodná pro tuto úlohu. Záznamy EEG pocházejí z Ústavu pro péči o matku a dítě a jsou ohodnoceny odborníkem.

Z celkových 547 příznaků získaných ze signálu byly provedeny selekce různého počtu příznaků 6 algoritmy – mRMR, SFS, SBS, SFFS, ReliefF a SBBS. Ke vzájemnému porovnání výsledků jednotlivých selekcí nám posloužily 4 různé klasifikátory – k-NN, SVM, Naive Bayes a C4.5. Jako kritérium výběru nejlepší kombinace algoritmus selekce - klasifikátor byla uvažována časová náročnost algoritmu a klasifikační chyba, které každá kombinace dosáhla. Následně byl zvolen optimální počet příznaků, který byl selektován celkem 160x s použitím různých algoritmů a datasetů. Nakonec byla vhodnost použití nejčastěji volených příznaků jako sady ověřena na nezávislém datasetu.

Celý experiment byl prováděn a zpracován v prostředí MatLab. Byly použity některé již naimplementované algoritmy, které jsou sjednoceny a spolu s dalšími kódy jednoduše spustitelné z jednoho kmenového souboru pro další možné experimenty.

2 EEG

2.1 Úvod

EEG je zkratka anglického slova „Electroencephalography“ (česky elektroencefalografie) – měření elektrické aktivity mozku. [3] Ta je zaznamenávána elektrodami umístěnými na lebce přístrojem nazývaným elektroencefalograf a takto pořízený záznam se nazývá elektroencefalogram. EEG nalézá široké uplatnění v diagnostice, počínaje diagnostikou nádoru či epilepsie až po rozpoznávání spánkových poruch. [1]

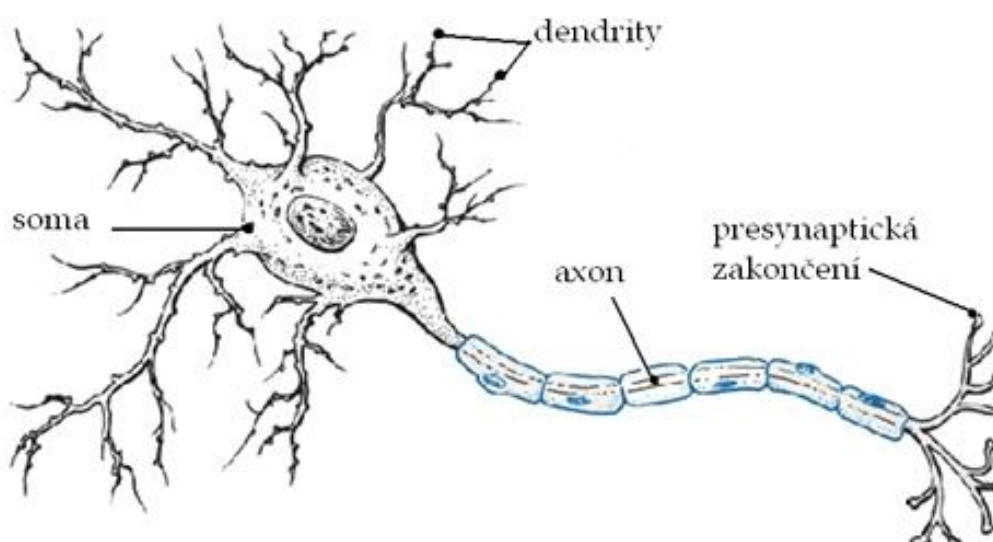
2.2 Lidský mozek

Trvalo velmi dlouho, než lidé přišli na to, že to, co je dělá lidmi, je vázáno na mozek. [4] Zpočátku byl mozek považován za chladicí systém, u starověkých Egypťanů dokonce za bezpodstatnou kašovitou hmotu, zatímco za hybatele života bylo považováno srdce. Již ve starém Řecku se ovšem našli vzbouřenci, kteří tvrdili, že právě mozek je sídlem inteligence člověka. Základy moderní neuroanatomie leží v polovině 19. století, kdy byl objeven neuron – nervová buňka. Rychlost výzkumu mozku akcelerovala zejména ve 2. polovině 20. století, kdy moderní elektronové mikroskopy umožnily nahlédnout do dějů v neuronových spojích (synapsích). [5]

Dnes již víme, že mozek je řídicí orgán nervové soustavy, který řídí a kontroluje většinu tělesných funkcí. Spolu s míchou tvoří centrální nervový systém (CNS). Je bezpečně uložen v lebeční dutině naplněné mozkomíšním mokem. [3]

Základním stavebním prvkem mozku je neuron. Neurony vytvářejí složitou, mnohostranně propojenou síť a spolu s podpůrnými buňkami formují jednotlivé části mozku. Pracují s informacemi, které prostřednictvím elektrických potenciálů umějí rozvádět, vytvářet i přeměňovat.

Ačkoli je dnes popsáno mnoho různých typů nervových buněk, jejich základní stavba je obdobná (viz Obrázek 1). Tělo (sóma) je centrální částí neuronu, obsahuje jádro a další buněčné organely. Stromečkovitě se větvící výběžky, vedoucí směrem k buněčnému tělu se nazývají dendrity. Naproti tomu axon je jediný výběžek neuronu, který vede směrem od těla. Koncové části axonu (tzv. presynaptická knoflíková zakončení) vstupují do synapsí a propojují se tak s dendrity ostatních buněk. [5] V synapsích se mění elektrický signál na chemický a zpět na elektrický, tak dochází k přenosu vzruchů mezi buňkami. Elektrické signály v neuronech jsou zdrojem pro měření EEG.



Obrázek 1: neuron [6]

2.3 EEG signál

2.3.1 Historie

Elektrické signály v těle savců byly poprvé objeveny v 19. století u neanestezovaných zvířat, čímž byly položeny základy neurofyzologie, vědy, která se zabývá studiem fyzikálních a chemických procesů v nervové soustavě. Koncem téhož století objevil Angličan Richard Caton přítomnost elektrického signálu v mozku. Zjistil také, že tyto signály vypadají jinak při spánku než v bdělém stavu.

Ve dvacátých letech 20. století byla poprvé zaznamenána elektrická aktivita mozku z povrchu neporušené lebky zdravého člověka. Zasloužil se o to německý lékař Hans Berger. V následujících letech díky zlepšujícím se přístrojům byly pořízeny záznamy pacientů s nádory mozku a s epilepsií. Následoval popis signálu v hypnóze a popis spánkových fází dospělého člověka. V šedesátých letech se začala zkoumat také mozková aktivita novorozenců, která je velmi odlišná od aktivity dospělého člověka. Od sedmdesátých let se v EEG začíná výrazně uplatňovat výpočetní technika. Vznikají první systémy pro automatické zpracování signálu využívající záznamová zařízení s mikroprocesory a nově vyvinuté numerické postupy zpracování dat.

V dnešní době se EEG využívá zcela běžně při standardních vyšetřeních. Často je však nahrazováno modernějšími metodami jako je počítačová tomografie nebo magnetická

rezonance, avšak oproti nim má výhodu vyšší rychlosti a nižších nákladů. Na poli zpracování signálu EEG se dnes dostávají ke slovu postupy umělé inteligence. [1; 3]

2.3.2 Snímání EEG

Důsledkem činnosti neuronů jsou elektrická napětí, která můžeme měřit pomocí elektrod. Podle jejich umístění rozlišujeme 2 způsoby – neinvazivní, kdy jsou elektrody umístěny na povrchu hlavy (většinou pomocí vodivého gelu), a invazivní, kdy se elektrody implantují pod lebeční kosti. Rozdíly v intenzitě jsou značné, invazivní způsob dosahuje zhruba stokrát vyšších hodnot napětí. Přesto se častěji využívá neinvazivní měření, protože není třeba chirurgický zákrok.

Signály z elektrod se vyhodnocují jedním ze dvou způsobů. Spojení mezi dvěma elektrodami je buď bipolární (měříme rozdíl signálu mezi dvěma sousedními elektrodami) nebo unipolární (měříme zvlášť rozdíl mezi každou elektrodou a referenční). Referenční elektroda bývá umístěna na uchu, bradě nebo se jedná o tzv. Goldmannovu elektrodu, která představuje spojení všech elektrod se zemí.

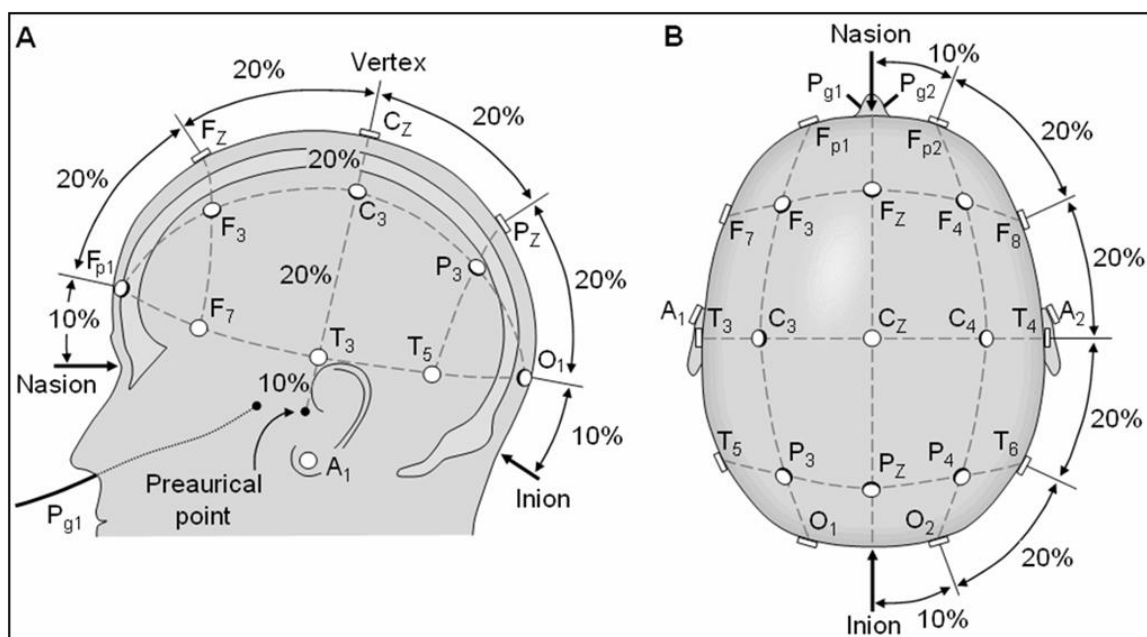
Samotný průběh signálu se zaznamenává pomocí jehel na papír nebo (dnes častěji) prochází digitalizací a zapisuje se do počítače. Objem digitalizovaných dat závisí na nastavených parametrech, jako je počet segmentů za sekundu či počet bitů na vzorek a také na délce záznamu. Tento objem se tedy může vyšplhat i na poměrně vysoké hodnoty, jeho nezměrnou výhodou je ovšem možnost zpracování nebo alespoň předzpracování pomocí PC. [3]

2.3.2.1 Rozložení elektrod

Rozložení elektrod na hlavě není náhodné. Počet elektrod se pohybuje od jedné až k několika desítkám. Nejčastěji se řídí podle jednoduchého měření, které navrhl roku 1957 Kanadčan Herbert Henri Jasper. Jde o tzv. systém „10–20“, který je dnes mezinárodním standardem (viz Obrázek 2). Vzájemná vzdálenost 19 elektrod je 10 % nebo 20 % v obou rovinách – příčné (mezi oběma zvukovody) i podélné (mezi kořenem nosu a zevním týlním hrbolkem). Pro zjednodušení se často používá speciální čepice, která již má elektrody na sobě umístěné. Tyto čepice se vyrábějí v různých velikostech, aby co nejlépe seděly každému pacientovi.

Jednotlivé roviny, ve kterých jsou elektrody umístěny, jsou označeny písmeny: Fp – prefrontální, F – frontální, C – centrální, P – parietální, O – okcipitální, T – temporální. Kolmo k nim jsou roviny laterální (na okraji hlavy), paramediální a mediální (mezi kořenem

nosu a týlem). Liché indexy označují levou hemisféru, sudá čísla pravou hemisféru a písmeno „z“ značí střed mezi oběma hemisférami. [7]



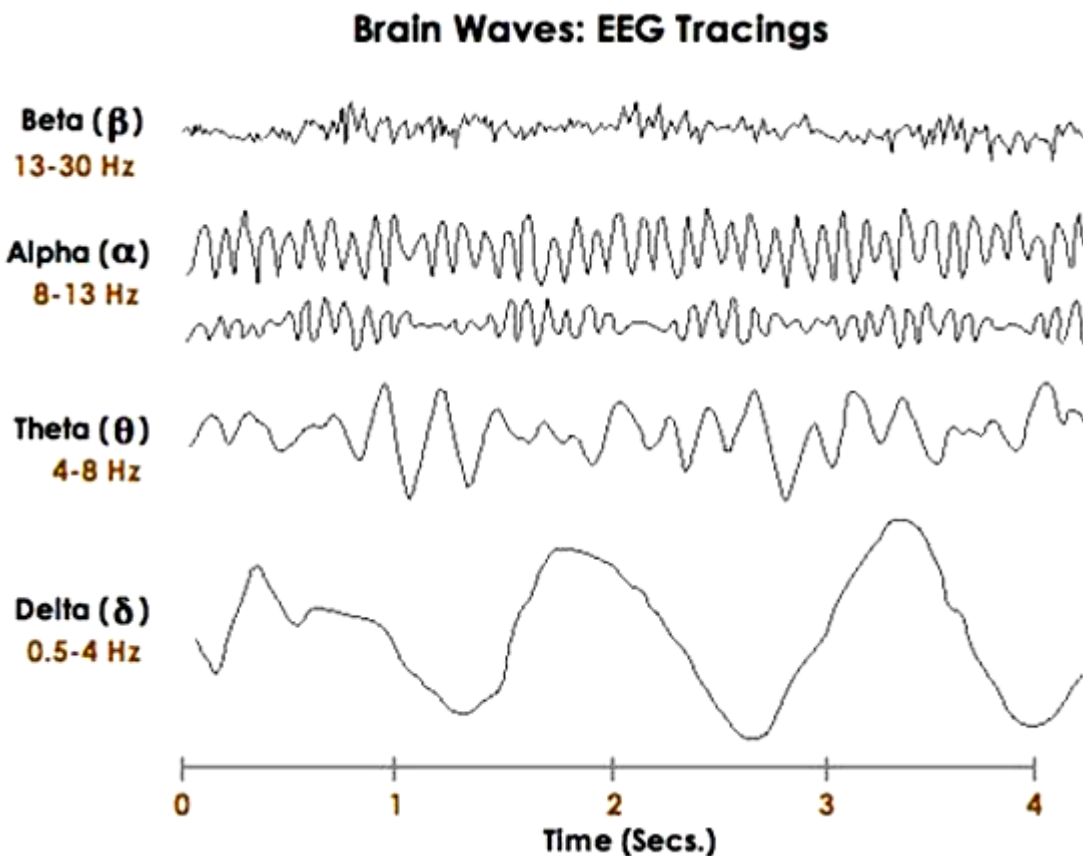
Obrázek 2: Poloha elektrod v systému „10–20“; zvýrazněn nasion (kořen nosu) a inion (zevní týlní hrbolky) [8]

2.3.3 Základní rytmy

Nejčastěji vychází popis EEG záznamu z frekvenčních spekter získaných Fourierovou transformací. Tato metoda se rozvinula především díky výpočetně nenáročné rychlé Fourierově transformaci (FFT – Fast Fourier Transform). Ve frekvenčním pásmu rozeznáváme 4 základní rytmy (delta, théta, alfa a beta), které se liší frekvencí i amplitudou a odpovídají spontánní aktivitě mozku (viz Obrázek 3). [9]

Delta rytmus s frekvencí mezi 0,5 a 4 Hz se objevuje u novorozenců do 1. roku života obvykle s vysokou amplitudou až 150 μV . U dospělých se vyskytuje ve III. a IV. stádiu NREM spánku (viz kapitola 2.3.6), při hyperventilaci a provází bezvědomí.

Théta rytmus má frekvenci 4 – 8 Hz a amplitudu do 70 μV , v bdělosti se u zdravého člověka téměř nevyskytuje. Provází stavy těsně před usnutím, při probuzení a během meditačních technik. Bývá spojován s fantazií, představivostí a kreativním myšlením. U dětí je běžný kolem prvního roku života, ale lze jej zachytit i u starších dětí. Bývá výraznější při emočním vzrušení.



Obrázek 3:Základní rytmy a jim odpovídající frekvence a typické průběhy [10]

Alfa rytmus je u dospělého člověka nejvýznamnější složkou EEG. Jeho frekvence je 8 – 13 Hz a amplituda 30 – 80 μV . Objevuje se při relaxované bdělosti, tlumí se pozorností, soustředěním a hlavně zrakovým vjemem – často hovoříme o blokadě alfa rytmu při otevření očí. Tento rytmus se jen zřídka vyskytuje u dětí mladších 6 let.

Beta rytmus je dominantním rytmem při stavu bdělosti s otevřenými očima. Má frekvenci 13 – 30 Hz a amplitudu obvykle 10 – 30 μV . Je velmi nepravidelný s nízkou amplitudou. Vyskytuje se především nad frontálními krajinami. Je pozorován u všech věkových skupin. [7]

Frekvenční pásmo nad 30 Hz se nazývá gama. Vlny o takto vysoké frekvenci se objevují při řešení náročných problémů a při zpracovávání velkého množství informací. Lidé, kteří efektně využívají gama pásmo, se vyznačují dobrou pamětí. [3]

2.3.4 Grafoelementy

Mimo základních rytmů se v EEG záznamu vyskytují další charakteristické vlny, hroty a jejich složitější komplexy významné pro diagnostiku, tzv. grafoelementy. Patří mezi ně

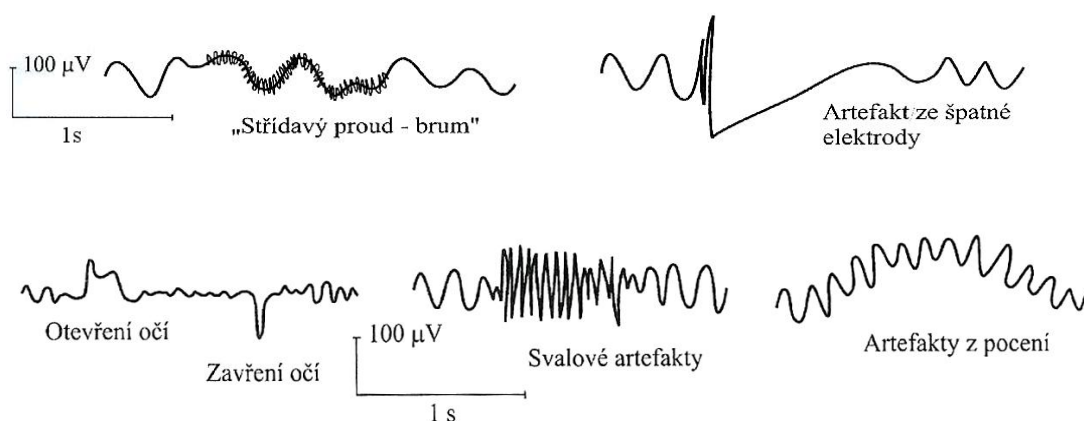
například k-komplex objevující se těsně před usnutím, μ -vlny související s pohybem či úmyslem se pohnout nebo třeba lambda vlna spjatá s upoutáním zraku. Typickým grafoelementem jsou také komplexy vyskytující se při epileptických záchvatech (viz Obrázek 4). [11]



Obrázek 4: Epileptický grafoelement (petit mal variant) [7]

2.3.5 Artefakty

Protože má elektrický potenciál mozku na povrchu lebky napětí jen několik desítek μ V, nalezneme v EEG záznamu kromě signálu vhodného k analýze tzv. artefakty – potenciály, které nemají vztah k měřenému signálu. Artefakty dělíme na dva druhy – artefakty technické a artefakty biologické (viz Obrázek 5). Typickými technickými artefakty jsou síťové rušení 50 Hz a artefakt ze špatné elektrody (kolísání izolinie či ztráta kontaktu). Jejich zdrojem je buď samotné záznamové zařízení, nebo jiné technické zařízení, které nepatří do měřicího řetězce, ale vyskytuje se poblíž. Zdrojem biologických artefaktů je samotný pacient, jedná se často o interference mezi jednotlivými biosignály. Řadíme mezi ně pohyby očí a očních víček, pocení, svalovou aktivitu a další. [12]



Obrázek 5: Technické a biologické artefakty [13]

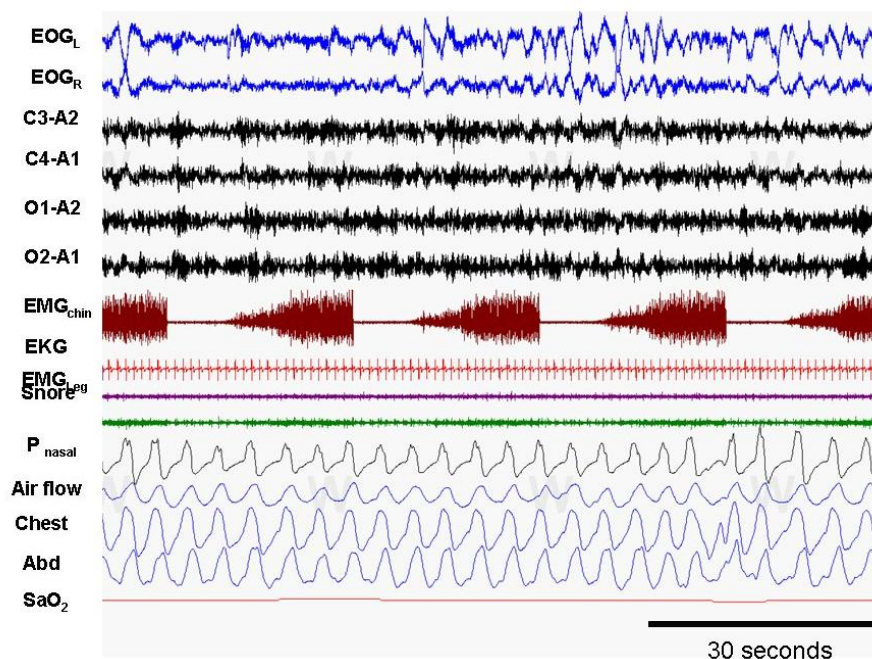
Některé artefakty se odstraní již při měření pomocí filtrů, jiné až při zpracování digitalizovaného signálu. Typicky se odstraňují filtrací frekvence nižší než 0,5 Hz jako šum a síťové rušení správně nastavenou pásmovou zádrží. [1]

2.3.6 Spánek

Spánek je základní fyziologická potřeba podobně jako příjem potravy a vody. Při spánku se snižuje funkčnost smyslů, klesá tělesná teplota, dýchání se zpomaluje a krevní tlak se snižuje. Spánek představuje regeneraci sil fyzických i psychických. Výzkum spánku byl zahájen ve 30. letech 20. století, v padesátých letech byl objeven a popsán tzv. REM spánek, který je doprovázen rychlými pohyby očí pod očními víčky, na jehož objev navázali roku 1968 Allan Rechtschaffen a Anthony Kales, kteří vytvořili dodnes používaný manuál k vyhodnocování fází lidského spánku. Měření EEG je důležitým prostředkem ke zkoumání a analýze spánku. [7]

2.3.6.1 Polysomnografie

Měří-li se během spánku současně s EEG ještě další signály, hovoříme o tzv. polysomnografii (PSG). Ta poskytuje doplňující informace ve formě záznamu dalších fyziologických hodnot a usnadňuje tak analýzu a ohodnocování záznamu. Používá-li se této metody při analýze spánku, hovoříme o tzv. spánkové polysomnografii, která je pro správné určení fází spánku nezbytná. Současně se měří: EEG, EKG (elektrokardiogram – záznam srdeční aktivity), EMG (elektromyogram – záznam svalové aktivity), EOG (elektrookulogram – záznam pohybu očí), PNG (pneumograf – záznam dechu) a často i některé další hodnoty jako například nasycení krve kyslíkem - SaO₂ (viz Obrázek 6). [14]



Obrázek 6: Ukázka polysomnografického záznamu [15]

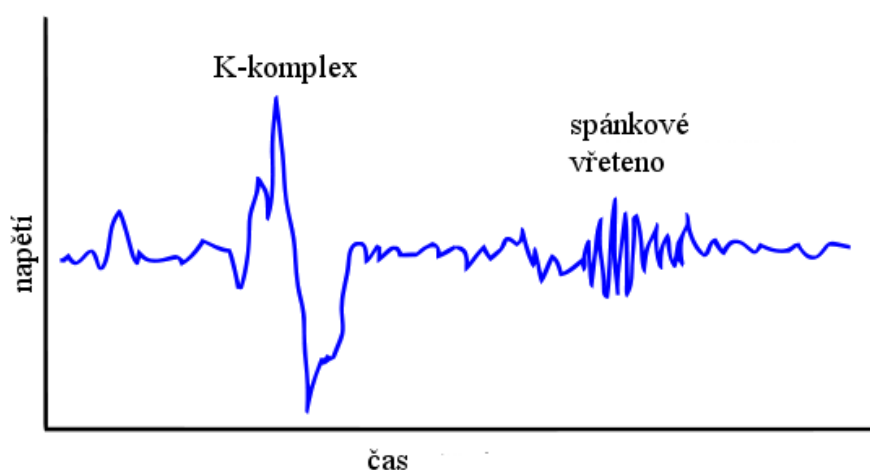
2.3.6.2 Spánkové fáze

Jak již bylo zmíněno, Rechtschaffen a Kales vytvořili manuál k vyhodnocování spánkových fází na základě rozdělení spánku na REM (paradoxní spánek) a NREM (synchronní spánek). Během nočního spánku se tyto fáze několikrát vystřídají.

Zkratka REM z anglického rapid eye movement vystihuje charakteristiku této fáze. REM fáze je doprovázena rychlými pohyby očí pod víčky, je charakteristická nízkým svalovým napětím a rychlým nízkonapěťovým EEG podobným průběhu v bdělém stavu, ačkoli se jedná o spánek hluboký. Proto také bývá označován za spánek paradoxní. REM spánek pokrývá asi 10–25 % celkového času spánku dospělého člověka a představuje tu část noci, ve které se zdají sny.

Opakem je NREM (non-REM) spánek, který se vyznačuje jen slabými pohyby očí. Někdy používané označení – synchronní – je odvozeno z EEG obrazu tohoto spánku. V EEG se objevuje aktivita vysokých pomalých vln, způsobená synchronizací elektrických potenciálů. Každá ze 4 fází, na které můžeme NREM dále dělit, se vyznačuje rozlišitelnou charakteristikou.

NREM 1 zahrnuje okolo 5 % spánku a objevuje se při usínání. V této fázi ustupuje alfa aktivita, která je nahrazována aktivitou théta. Klesá amplituda EMG a EOG značí pomalé krouživé pohyby očí. NREM 2 začíná obvykle po 10 minutách spánku a představuje asi polovinu celkového času. Křivka EEG se základní aktivitou théta je charakteristická tzv. spánkovými vřeteny a K-komplexy. [7]



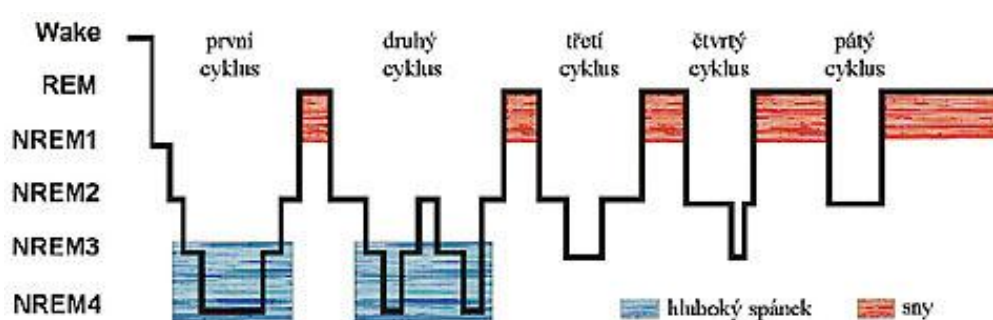
Obrázek 7: EEG křivka s K-komplexem a spánkovým vřetenem [16]

NREM 3 a 4 pokrývají společně asi 15 % spánku a hovoříme o nich jako o hlubokém spánku. V EEG se objevuje stále častější delta aktivita, ve 3. fázi pokrývá 20–50 % a má nepravidelný tvar. Ve 4. fázi delta aktivita přesahuje 50 %. V této fázi se již vůbec nevyskytují spánková vřetena, která ještě můžeme občas zachytit v NREM 3. Svalová aktivita i oční pohyby jsou vzhledem k bdělosti zanedbatelné. [7]

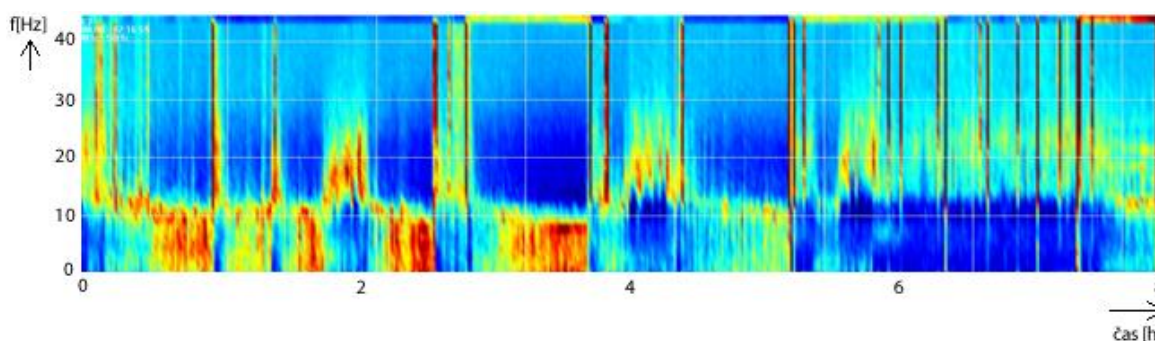
2.3.6.3 Hypnogram a spektrogram

Sled jednotlivých spánkových fází během noci se zobrazuje pomocí grafu tzv. hypnogramu, jehož název souvisí s řeckým bohem spánku – Hypnem. Jednotlivé fáze se určují pomocí PSG. Hypnogram slouží jako pomůcka k diagnóze spánkových poruch. [17]

Oproti hypnogramu má spektrogram o rozměr víc – je to trojrozměrný graf s frekvenční osou, časovou osou a složkou energie. Podobně jako hypnogram jej lze použít ke sledování spánkových fází během noci. Většinou se na vodorovnou osu vynášejí čas, na svislou osu frekvence a barva zastupuje energetickou složku – červená odpovídá největší četnosti, modrá naopak minimální. [19]



Obrázek 8: Hypnogram dospělého člověka během celé noci [18]



Obrázek 9: Spektrogram spánkového cca 8 hodinového záznamu

2.3.6.4 Spánek novorozenců

Zatímco dospělí lidé tráví spánkem okolo 7 hodin denně, u novorozenců a batolat zabírá spánek hlavní část dne. I jeho architektura se během prvních tří let vyvíjí a mozek postupně dozrává. U novorozenců rozlišujeme dva druhy spánku: aktivní a klidný spánek.

Aktivní spánek se podobá REM fázi spánku dospělého člověka a u novorozenců představuje asi 70 % spánku. V této fázi se často objevují sací pohyby, grimasy a občasné hlasové projevy (např. krátké zaplakání), novorozenec dýchá nepravidelně. Naproti tomu klidný spánek je podobný NREM spánku, ačkoli delta aktivita není tak výrazná. Je typický minimálními pohyby a klidným dechem. Jak dítě vyžívá, doba spánku se zkracuje a mění se poměr mezi aktivním a klidným spánkem.

Pro snímání se používá pouze 8 elektrod, protože hlavička novorozenců je malá. Je velký problém s artefakty, protože se děti hodně pohybují a elektrody často odpadávají. I proto je analýza spánku dětí o tolik obtížnější než u dospělých. [12; 20; 21]



Obrázek 10: Měření EEG na novorozenci [21]

3 Zpracování signálu

Vyhodnocení zaznamenaného signálu EEG je náročný úkol – jak časově, tak kvalifikačně. Odborníků není mnoho a vyhodnocení záznamu trvá nezanedbatelně dlouho, například analýza 24 hodinového záznamu trvá zhruba 4 hodiny. [17] V dnešní době je nenahraditelnou pomocí počítačové zpracování, které vyhodnocování značně usnadňuje, ačkoli zatím počítače nedokážou práci odborníka plnohodnotně nahradit. V oblasti výzkumu se však stále hledají postupy, které by dokázaly signál dostatečně přesně ohodnotit. Celý takovýto postup sestává z několika kroků – předzpracování, segmentace, výpočet (extrakce) příznaků, selekce příznaků a konečně klasifikace.

3.1 Předzpracování

Předzpracování signálu obnáší převzorkování signálu, odfiltrování síťového rušení a vysokofrekvenčního šumu, odstranění izoliny nebo třeba úprava amplitudy. [22]

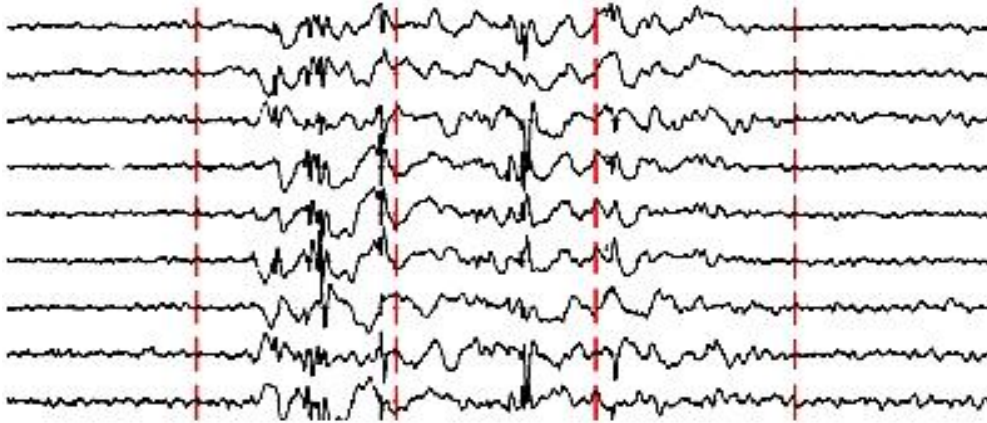
K minimalizování vlivu rozvodné elektrické sítě lze využít speciálně stíněných prostor, ale nebývá to běžnou praxí. Protože je frekvence rozvodné sítě téměř konstantní, lze ji odstranit pomocí jednoduchého číslicového filtru typu pásmová zadrž na frekvenci 50 Hz. Podobně se odstraňuje i vysokofrekvenční šum filtrem typu dolní propust a také nízké frekvence (např. dýchání) horní propustí se zlomovou frekvencí okolo 0,1 Hz. Většinou využíváme filtry typu FIR, které jsou lineární a nemění fázi signálu. [17]

3.2 Segmentace

Segmentací signálu rozumíme jeho rozdělení na kratší úseky. Protože signál EEG je nestacionární, slouží segmentace především k tomu, abychom našli úseky, ve kterých se námí požadovaná charakteristika nemění, nebo vykazuje pouze malé rozdíly. Tyto úseky nazýváme „lokálně stacionární segmenty“. [1] Rozlišujeme dva základní typy segmentace: segmentaci konstantní a segmentaci adaptivní.

3.2.1 Konstantní segmentace

Signál je konstantní segmentací rozdělen na úseky o stejné délce – tedy stejném počtu segmentů, který předem určíme. Výhodou tohoto postupu je jednoduchá implementace a nízká výpočetní náročnost. Na druhou stranu velkou nevýhodou je ta skutečnost, že hranice takto vytvořených úseků nemá přímý vztah k charakteru signálu, a nemůžeme tedy s jistotou tvrdit, že charakteristika signálu je v daném úseku téměř neměnná. [1]



Obrázek 11: Výsledek konstantní segmentace. Vidíme, že hranice segmentů nejsou optimální.

3.2.2 Adaptivní segmentace

Při tomto způsobu segmentace je signál rozdělen na úseky proměnné délky. Ta se mění se změnou námi požadované charakteristiky a přizpůsobuje se okamžitému stavu signálu (odtud název adaptivní). Výsledkem je signál rozdělený na segmenty podobných vlastností. [19]

Často používanou metodou adaptivní segmentace je tzv. metoda dvou spojených oken, která je obměnou algoritmu navrženého roku 1977 Bodenstemem a Praetoriem. Algoritmus využívá dvou spojených oken, které se posouvají po signálu a počítají stejné charakteristiky. Z rozdílu charakteristik signálu v obou oknech se určí míra diference. Hranice segmentu je potom umístěna v lokálních maximech diference (viz Obrázek 12). [11]

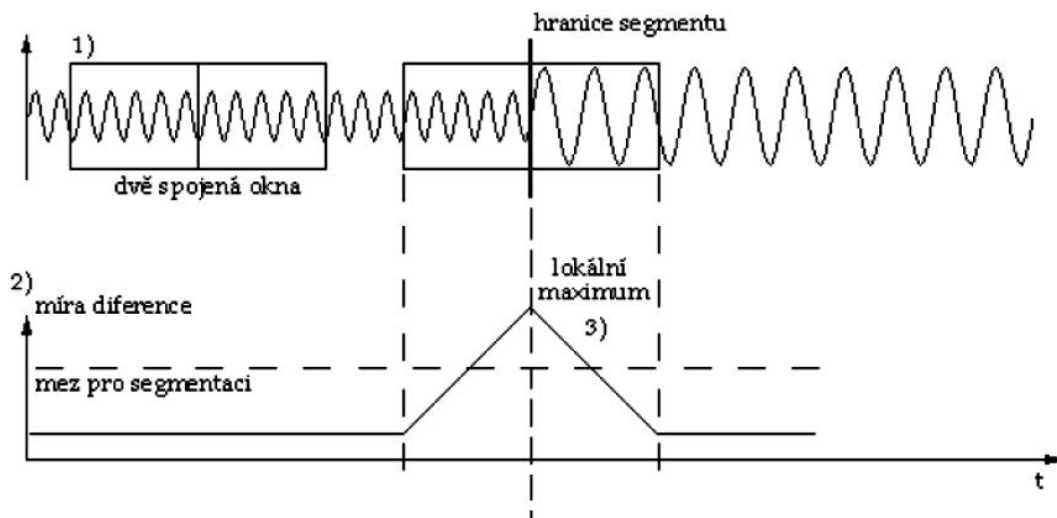
Výpočet celkové míry diference vychází z jednoduchého a rychlého výpočtu střední amplitudy a střední frekvence.

$$A_w = \sum_{n=1}^{WL} |x_n| \quad (1)$$

$$F_w = \sum_{n=1}^{WL} |x_n - x_{n-1}| \quad (2)$$

Střední amplitudu v okně w (A_w) spočteme dle (1), kde WL je délka okna a x_n hodnota n -tého vzorku. Odhad střední frekvence (F_w) vychází z poznatku, že průměrná diference je přímo úměrná střední frekvenci signálu (2). Protože jsou obě okna stejně dlouhá, není nutné počítat průměrnou hodnotu na vzorek. Celková míra diference G tedy bude (3), přičemž k_a a k_f jsou vhodně zvolené váhovací konstanty. [23; 24]

$$G = k_a |A_1 - A_2| - k_f |F_1 - F_2| \quad (3)$$



Obrázek 12: Princip adaptivní segmentace [11]

3.3 Výpočet příznaků

Výsledné segmenty získané některým způsobem segmentace chceme následně rozřadit do tříd podle vlastností tak, aby každá třída obsahovala vzájemně podobné segmenty signálu. Pro každý segment lze vypočítat velké množství různých příznaků, které se ke klasifikaci využívají. Proces, při kterém vytváříme množinu charakteristických atributů (příznaků), nazýváme extrakcí příznaků. [17]

Příznaků, které můžeme spočítat, je velké množství. K jejich výpočtu se používá znalostí ze statistiky i některých transformací – např. FFT. Následuje přehled některých příznaků, které jsou často využívány a vyskytují se i u dat v našem experimentu:

- minimální a maximální hodnota
- střední hodnota a směrodatná odchylka
- koeficient špičatosti – porovnává rozdělení signálu s normálním rozdělením
- koeficient šikmosti – kladná nebo záporná hodnota, podle toho, na jakou stranu od průměru jsou odchylky větší
- průměr a maximum hodnot první a druhé derivace
- absolutní a relativní výkon základních rytmů EEG a jejich poměrné časové zastoupení
- hodnoty vlnkových koeficientů – získané tzv. Waveletovou (vlnkovou) transformací
- entropie

- efektivní hodnota (RMS)
- počet průchodů nulou a délka křivky
- koherence a koeficienty vzájemné korelace jednotlivých elektrod

3.4 Selekcce příznaků

Příznaky, kterými popisujeme segmenty, můžeme uspořádat do n -rozměrného vektoru, kde n je počet extrahovaných příznaků. Jednotlivé segmenty jsou poté reprezentovány body v n -rozměrném prostoru. [11] Klasifikace následně provádí zobrazení z tohoto prostoru do prostoru tříd, který bývá obvykle jednorozměrný.

Výpočetní náročnost tohoto zobrazení stoupá s počtem příznaků. Některé příznaky obsahují shodné nebo velmi podobné informace a jiné mohou být pro danou úlohu bezvýznamné. Cílem selekcce příznaků je odstranit redundantní a irelevantní informace a snížit tak dimenzi prostoru (více viz kapitola 4). [25] Tento krok zpracování signálu se někdy vynechává.

3.5 Klasifikace

Posledním krokem počítačového zpracování EEG je samotná klasifikace. Jak již bylo zmíněno, klasifikace se snaží provést zobrazení z prostoru příznaků do prostoru tříd. Máme-li klasifikátor q a vektory dat x_i , označíme \tilde{y}_i jako třídu, kterou přiřadí klasifikátor datům, a snažíme se o to, aby toto přiřazení bylo co nejpřesnější, tedy aby rozdíl mezi přiřazenou a reálnou třídou byl co nejmenší. Klasifikátor k tomu potřebuje ještě dodatečnou znalost (označíme Θ), takže získáme funkci:

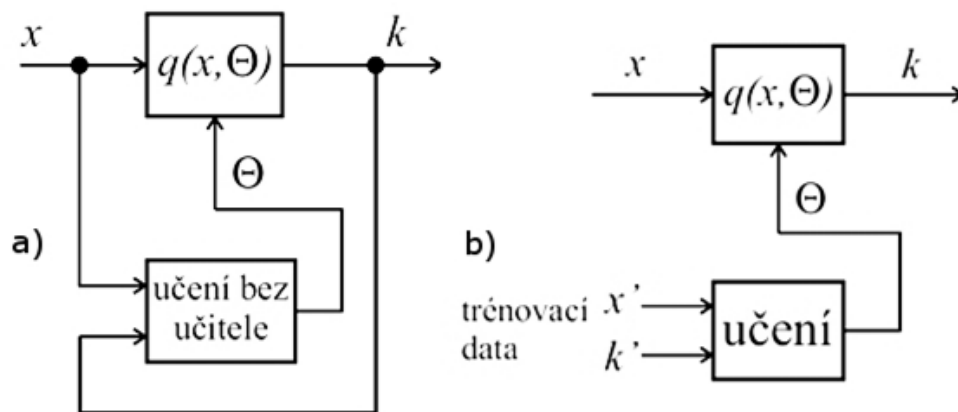
$$\tilde{y}_i = q(x_i, \Theta) \quad (4)$$

Podle toho, zda jsou třídy známy předem, nebo je teprve vytváříme, rozlišujeme dva základní způsoby: učení bez učitele (Unsupervised Learning) a učení s učitelem (Supervised Learning). Tyto metody se liší množinou, ze které získáváme znalost Θ (viz Obrázek 13). [26]

Učení bez učitele pracuje s daty, u nichž není správné přiřazení do tříd známo, často se jedná o data z experimentů, kde rozřazení do tříd teprve hledáme. Znalost tedy musíme získat ze samotných neohodnocených dat. Existují dva základní přístupy: statistický a deterministický. Statistický přístup chápe pozorování jako náhodné veličiny. Výsledkem učení je statistický model dovolující přiřadit pozorování ke třídě podle modelované sdružené hustoty pravděpodobnosti. Když se pozorovaná data snažíme vysvětlit pomocí matematického modelu, na základě myšlenky, že data patřící do jedné třídy si jsou podobná podle jiné míry

(např. vzdálenosti), mluvíme o deterministickém přístupu – někdy se také používá pojem shluková analýza. [27]

Naproti tomu učení s učitelem dostává data, u kterých je přiřazení do tříd známo. Část dat použijeme jako data trénovací a část jako data testovací. Z trénovacích dat získá algoritmus potřebnou znalost, jinými slovy „natrénuje se“, a na testovacích datech následně ověří její správnost. (více viz 3.5.2). Nebezpečím této metody je přeučení (Overfitting) – stav, kdy je systém příliš přizpůsoben množině trénovacích dat, ale selhává na množině testovacích dat. K tomu může dojít, je-li trénovací množina příliš malá, nebo pokud je systém příliš komplexní (nebezpečí neuronových sítí). [28]



Obrázek 13: Princip získávání znalosti potřebné k rozřazení do tříd u učení bez učitele (a) a učení s učitelem (b)

Při klasifikaci EEG se většinou využívá učení s učitelem. Jako trénovací data využíváme záznamy ohodnocené expertem. Na jejich základě jsme potom schopni ohodnotit i jiná data představující stejnou úlohu (např. spánkové nebo epileptické záznamy). Učení bez učitele se využívá zřídka, například snažíme-li se nalézt souvislosti mezi mozkovou aktivitou a projevem některé nemoci, abychom ji byli schopni předvídat.

3.5.1 Klasifikátory

Klasifikátory, které často používáme při zpracování signálu, dělíme podle principu na:

- lineární klasifikátory – např. perceptronový, SVM (Support Vector Machine)
- IBL (Instance Based Learning) – učení založené na instancích, např. k-NN
- Bayesovské klasifikátory – založené na statistickém modelování
- rozhodovací stromy a tabulky
- neuronové sítě
- genetické algoritmy [29; 30]

Popis principu každého samotného klasifikátoru je obsáhlý, proto zde budou uvedeny pouze ty, které jsou použity v experimentální části: SVM, k-NN, naivní Bayesův klasifikátor (Naive Bayes) a C4.5 (rozhodovací strom). Tyto klasifikátory jsou již implementovány v použitém software a, protože představují různé přístupy, jsou vhodné ke vzájemnému porovnání.

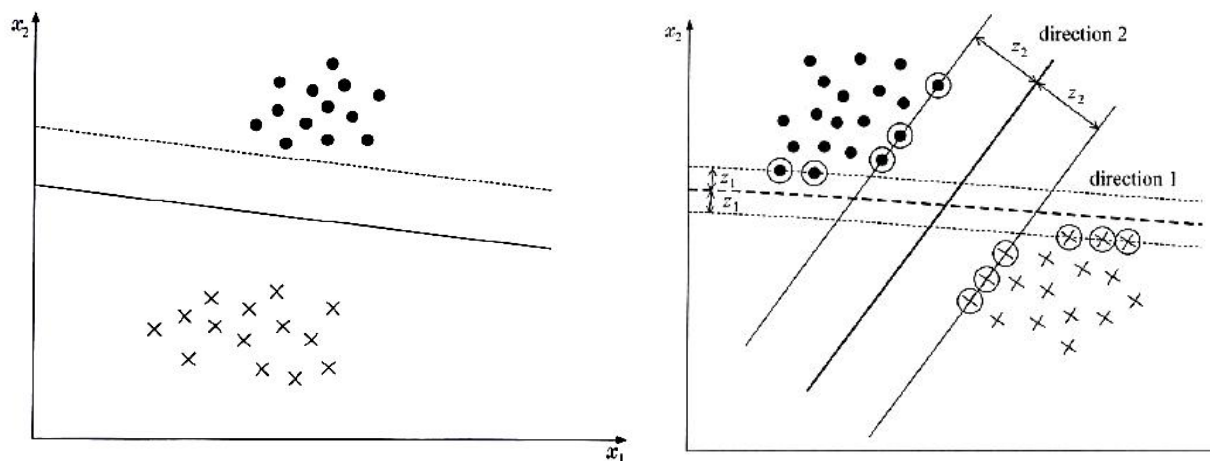
3.5.1.1 SVM

Lineární SVM (lineární podpůrné vektorové stroje) se snaží nalézt optimální nadrovinu, která od sebe odděluje prvky tříd. Je to klasifikátor z principu binární, lze jej tedy použít pouze pro data, která mají dvě třídy.

Mějme separabilní třídy $y_1=1$ a $y_2=-1$ a trénovací vzorky x_1 až x_n , potom označíme množinu bodů \mathbf{x} , které splňují:

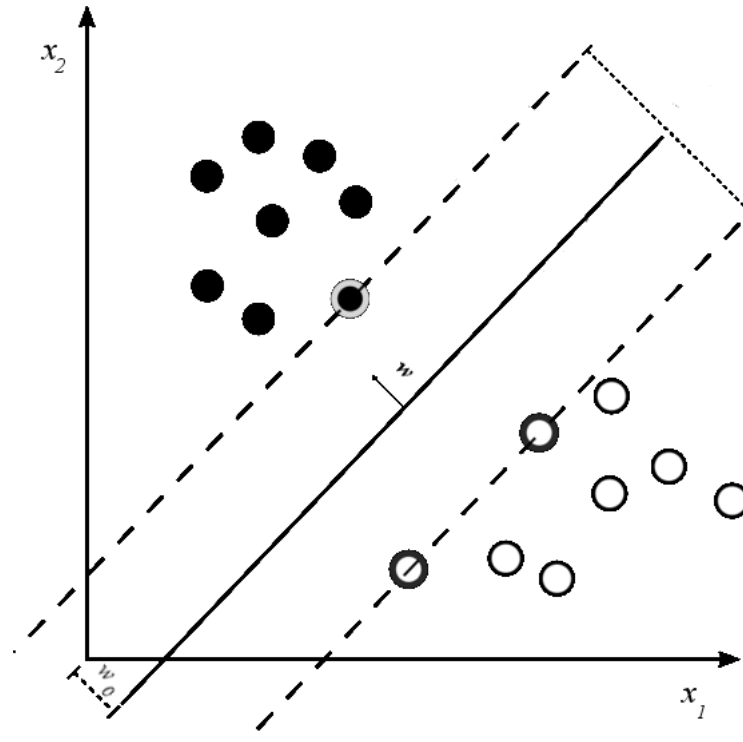
$$\mathbf{w} \cdot \mathbf{x} - w_0 = 0 \quad (5)$$

jako klasifikační nadrovinu, která správně klasifikuje dané vzory (\mathbf{w} je tedy normálový vektor k dané nadrovině). Tato rovina ovšem není určena jednoznačně (viz Obrázek 14 vlevo). Hledáme proto parametry \mathbf{w} a w_0 takové, aby vzdálenost nejbližších segmentů obou tříd od roviny \mathbf{x} byla co nejvyšší (tedy aby ležela „uprostřed“).



Obrázek 14: Příklad nejednoznačnosti klasifikační nadroviny důvodu maximalizace vzdálenosti [29]

Zároveň požadujeme, aby okolí této nadroviny bylo maximální (viz Obrázek 14 vpravo). Můžeme nalézt dvě mezní nadroviny \mathbf{x}_1 a \mathbf{x}_{-1} (meze), které jsou co nejvíce vzdálené od sebe, ale obě ještě separují data – jsou to ty, které obsahují „krajní“ vzorky tříd (viz Obrázek 15). Ty potom nazýváme podpůrné vektory – odtud název.



Obrázek 15: Optimální nadrovina a vyznačené mezní nadroviny

Tyto meze splňují rovnici (6):

$$y_i(\mathbf{w} \cdot \mathbf{x}_i - w_0) = 1 \quad (6)$$

a vzdálenost mezi nimi je:

$$\frac{2}{\|\mathbf{w}\|} \quad (7)$$

Chceme-li maximalizovat tuto vzdálenost, musíme minimalizovat jmenovatele. Z (6) a (7) jsme získali podmínky pro parametry \mathbf{w} a w_0 . Řešení těchto podmínek je realizováno pomocí Lagrangeovy funkce. [29]

3.5.1.2 k-NN

Klasifikátor k nejbližších sousedů zařadí prvek do třídy, do které patří nejvíce z jeho k sousedů, přičemž k můžeme stanovit předem nebo ho necháme vyhledat algoritmem jako nejvhodnější číslo. Důležité je zvolit správnou metriku d , ta musí splňovat pravidla (8) - (11).

$$\text{nezápornost: } d(A,B) \geq 0 \quad (8)$$

$$\text{identita: } d(A,B) = 0 \Rightarrow A = B \quad (9)$$

$$\text{komutativita: } d(A,B) = d(B,A) \quad (10)$$

$$\Delta \text{ nerovnost: } d(A,B) \leq d(A,C) + d(C,B) \quad (11)$$

Tuto metriku dobře splňují následující definice vzdálenosti, které jsou zároveň schopny pracovat s libovolným počtem dimenzí:

- Hammingova vzdálenost (Manhattan): součet absolutních hodnot
- Euklidovská vzdálenost: odmocnina ze součtu čtverců
- Chebyshevova vzdálenost: maximum z absolutních hodnot

Zvolíme-li některou metriku, musíme se ještě rozhodnout, zda budeme přikládat všem vybraným sousedům stejnou váhu nebo, zda použijeme váhovací koeficienty (w_i), abychom zvýhodnili ty sousedy, kteří jsou blíže. Přiřadíme klasifikátorem q danému vzorku \mathbf{x} takovou třídu y , pro kterou platí:

$$y = q(\mathbf{x}) = \operatorname{argmax}_y \sum_{i=0}^K w_i \cdot y_i(\mathbf{x}), \quad (12)$$

kde w_i je váhovací koeficient a y_i třída i -tého souseda. [30]

3.5.1.3 Naive Bayes

Bayesovská klasifikace využívá podmíněné (aposteriori) pravděpodobnosti

$$P(y|\mathbf{x}), \quad (13)$$

kde y je třída a \mathbf{x} jsou všechny příznaky našeho vzorku. Vzorek je přiřazen ke třídě, pro kterou je (13) maximální. Aplikujeme-li Bayesův teorém, získáme:

$$P(y|\mathbf{x}) = \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} \quad (14)$$

přičemž $P(y)$ – apriorní pravděpodobnost třídy y a $P(\mathbf{x})$ – apriorní pravděpodobnost daného vzorku můžeme určit z trénovací množiny. Obtížnější je to ovšem s $P(\mathbf{x}|y)$ – aposteriori pravděpodobností daného vzorku ve třídě y .

Předpokládáme-li nezávislost příznaků (pro tento předpoklad budeme klasifikátor nazývat „naivní“) z definice podmíněné pravděpodobnosti víme, že platí (15):

$$P(\mathbf{x}|y) = \prod_{i=1}^n P(x_i|y), \quad (15)$$

kde x_1 až x_n jsou jednotlivé příznaky vzorku \mathbf{x} . Tyto pravděpodobnosti již z trénovacích dat určit umíme. Jelikož nepotřebujeme znát přesnou hodnotu pravděpodobnosti (14), nemusíme uvažovat jmenovatele zlomku, který je konstantou nezávisle na třídě y . [30] Z toho, za použití vztahů (14) a (15), získáme:

$$y = q(\mathbf{x}) = \operatorname{argmax}_y \prod_{i=1}^n P(x_i|y) P(y) \quad (16)$$

3.5.1.4 C 4.5

C 4.5 je algoritmus navržený Rossem Quinlanem. Algoritmus vytváří rozhodovací strom z trénovacích dat. Je založen na myšlence, že původní množina se dá rozdělit na homogennější podmnožiny. Za homogennější je v tomto případě považována množina, která má nižší entropii, přičemž vycházíme ze Shannonova chápání entropie:

$$S = - \sum_i p_i \log_2 p_i, \quad (17)$$

kde S je entropie a p_i pravděpodobnost i -tého stavu. Základem je tedy stanovení pravděpodobnosti výběru prvku z množiny tak, aby patřil do určité třídy. Algoritmus hledá takový atribut, který rozdělí množinu na podmnožiny s nejnižší možnou entropií. Postup se opakuje tak dlouho, dokud lze entropii zmenšovat.

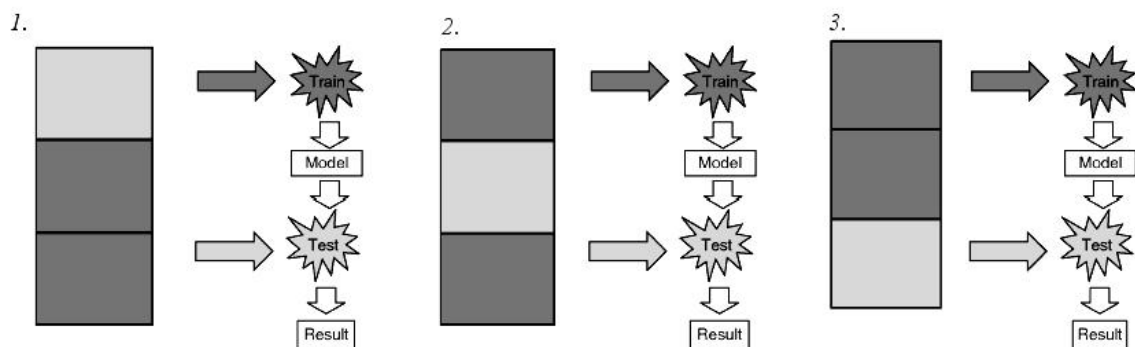
Když klasifikujeme nový vzorek, projdeme diagramem přes rozhodovací uzly až ke konkrétnímu listu, který představuje třídu, kterou klasifikátor vzorku přiřadí. [31]

3.5.2 Validace

Jak již bylo zmíněno, správnost „naučení se“ na trénovacích datech ověřujeme na datech testovacích. Často nemíváme dostatek dat, nebo chceme zajistit co největší přesnost a obecnost klasifikátoru (výkonnost), aby byl schopen ohodnotit i data, se kterými se ještě nesetkal, s přiměřenou přesností. Dnes známe různé metody ověřování správnosti (validace), z nichž některé používáme nejen k optimalizaci výkonnosti daného klasifikátoru, ale také nám pomáhají vybrat nejvhodnější klasifikátor pro danou úlohu. Rozlišujeme 5 různých metod validace: resubstituční, Hold-Out, K-Fold cross-validation (CV), Leave-One-Out CV a opakovanou K-Fold CV (Repeated).

Resubstituční metoda je nejjednodušším způsobem validace, používá stejná data pro trénování i testování. Je snadná na implementaci, ale protože způsobuje overfitting, vůbec se není využívána. Druhou metodou, jen o něco málo náročnější, je tzv. Hold-Out (vynechání). Tu část dat, kterou použijeme k trénování, vynecháme při testování – rozdělíme tedy data na dvě části a každou použijeme k jinému účelu. Nebezpečím je velká závislost na volbě dělení. Hold-Out validaci používáme, máme-li velký objem dat, protože je málo výpočetně náročná. V praxi se při experimentech používá obdoba, tzv. Three-Way Data Split, kdy data rozdělíme na 3 části – jednu použijeme jako data trénovací, druhou jako data testovací a třetí část použijeme jako nezávislá data, na která natrénovaný a otestovaný klasifikátor použijeme.

Další metody jsou typy tzv. cross-validation. Cross-validation rozdělí data na určitý počet částí. V každém kroku potom vyloučí jednu část k testování a ostatní použije k trénování. Toto opakuje vždy s jinou částí (viz Obrázek 16). K-Fold CV rozdělí data na K dílů (K předem určíme). Nejčastěji používáme $K=10$. Leave-One-Out CV je vlastně speciálním případem K-Fold CV, kdy dělíme na tolik částí, jako je počet segmentů – k testování vždy vyčleníme jeden vzorek. Repeated K-Fold CV, jak název napovídá, je opakovaně prováděná K-Fold CV, přičemž mezi každým opakováním vzorky promícháme. Přehled výhod a nevýhod všech způsobů validace viz Tabulka 1. [30; 32; 33]



Obrázek 16: Princip cross-validation [32]

Tabulka 1: Výhody a nevýhody různých metod validace [32]

metoda validace	výhody	nevýhody
resubstituční	jednoduchost	overfitting
Hold-Out	nezávislost trénovacích a testovacích dat	velká závislost výkonnosti na volbě dělení;
K-Fold CV	přesné určení výkonnosti	omezený počet rozdělení; překrývající se trénovací data
Leave-One-Out CV	nezkreslené určení výkonnosti	velký rozptyl; výpočetní náročnost
Repeated K-Fold CV	velký počet odhadů výkonnosti	překrývající se trénovací a testovací data mezi opakováními

4 Selekce příznaků

Jak již bylo zmíněno v kapitole 3.4, počet příznaků je někdy příliš vysoký, a proto potřebujeme snížit dimenzi prostoru. K tomuto kroku většinou přistupujeme z důvodu zefektivnění klasifikace, ale můžeme snižovat dimenzi i za účelem vhodnější reprezentace dat. My se v této práci soustředíme na snižování dimenze v klasifikačním procesu. [34]

Vzhledem ke stále se zvyšujícím datovým kapacitám, není dnes již nutné omezovat objem nasbíraných dat. Velký počet nasbíraných příznaků může vzniknout tím, že:

- a) data jsou nashromážděna pro více úloh a vyskytují se v nich příznaky specifické pro jednotlivé úlohy, které jsou pro jiné nepodstatné,
 - b) neznáme příznaky, které jsou důležité pro danou úlohu, z experimentu proto získáme tolik údajů, kolik můžeme,
 - c) máme data z různých zdrojů (např. senzorů), jejichž informace se překrývají.
- [35]

Pro klasifikaci však není nejvhodnější používat příliš velký počet příznaků, ačkoli jsme schopni je získat a uchovat. Ne vždy totiž platí „čím více, tím lépe“. Máme-li body, které jsou v dvojrozměrném prostoru blízko u sebe, mohou být od sebe ve storozměrném prostoru velmi vzdálené. Navíc se nám se zvyšujícím počtem příznaků výrazně zvětšuje i počet možných kombinací. Vezmeme-li v úvahu dvojstavové příznaky, máme pro pět příznaků 32 možných kombinací, ale pro deset příznaků je tento počet již 1024. Segmenty EEG z našeho experimentu mají 547 příznaků, kdyby byly dvojstavové (což nejsou), počet možných kombinací by byl 2^{547} , tedy něco kolem 10^{164} . Chceme-li potom natrénovat klasifikátor, potřebujeme trénovací množinu, která se tomuto počtu řádově blíží.

Shrneme-li tyto důvody, snížení dimenze přináší:

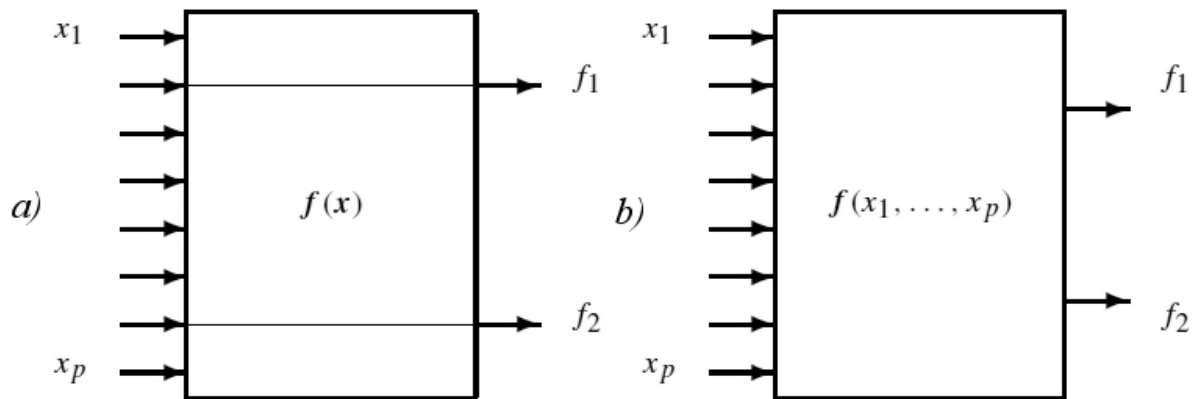
- snížení výpočetní náročnosti a objemu ukládaných dat
- nutnost menší trénovací množiny a snížení času potřebného k trénování
- větší přesnost klasifikace [36; 37]

4.1 Metody snižování dimenze

Rozlišujeme dva způsoby snižování dimenze – extrakci (feature extraction – FE) a selekci (feature selection - FS). Extrakce generuje nové příznaky, kdy každý z nich je nějakou kombinací příznaků původních. Většina algoritmů FE provádí lineární transformaci,

založenou na vlastních číslech a vektorech. Patří sem například PCA (Principal Component Analysis – analýza hlavních komponent) založená na Karhunen-Loèevově transformaci.

Oproti tomu, selekce příznaků vybírá nejvhodnější příznaky z původní množiny. Používáme jí, potřebujeme-li zachovat význam příznaků (viz Obrázek 17). [34; 38]

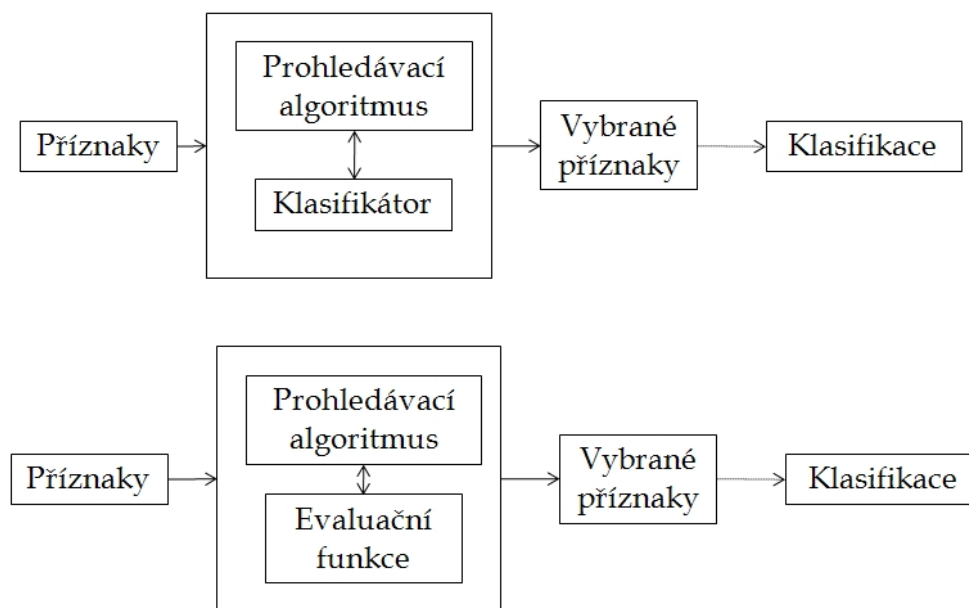


Obrázek 17: Princip selekce (a) a extrakce příznaků (b) [38]

Úkolem FS je odstranit ty příznaky, které nejsou relevantní, a/nebo ty, které jsou redundantní. Příznak považujeme za relevantní, existuje-li nějaká dvojice instancí rozdílných tříd, které se liší pouze v tomto příznaku a které mohou být do různých tříd rozděleny pouze na základě tohoto příznaku. Redundantní příznaky jsou takové, které obsahují nadbytečné informace. Máme-li opět dvě instance různých tříd a dva příznaky, nenalezneme takovou dvojici příznaků, která by se v jednom z příznaků lišila a v druhém shodovala. [36; 39]

4.2 Charakteristika selekce příznaků

Algoritmy selekce příznaků využívají většinou jednoho ze dvou modelů – modelu typu „Wrapper“ (tzv. obalovací) nebo modelu typu „Filter“. Algoritmy modelu Filter vybírají příznaky pouze na základě dat samotných pomocí evaluační (ověřovací) funkce např. výpočtu vzdálenosti mezi třídami. Na rozdíl od toho, je model Wrapper jakousi obálkou prohledávacího algoritmu s klasifikátorem. Algoritmy modelu Wrapper mohou v sobě zahrnovat téměř jakýkoli známý klasifikátor (více viz Obrázek 18). [36; 40]



Obrázek 18: Wrapper (nahore) a Filter (dole) schéma algoritmu selektce příznaků [41]

4.2.1 Obecný algoritmus

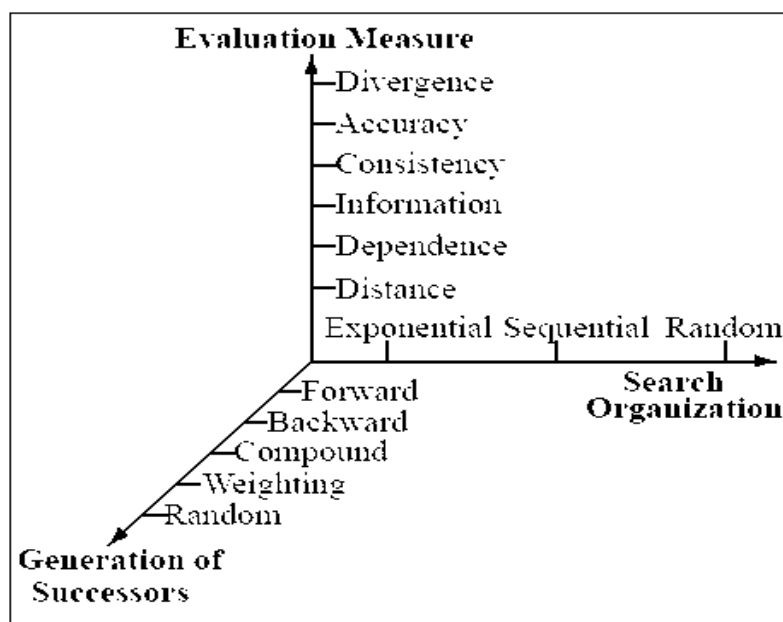
Mějme \mathbf{X} soubor vektorů $\mathbf{x}_1 - \mathbf{x}_n$, s příznaky X ($x_1 - x_m$). Je-li H oceňování příznaků, které má být co největší, a G mechanismus, kterým vybíráme následující prvek, potom obecný algoritmus FS vypadá v pseudokódu nějak takto:

```

L := StartPoint(X);
Solution := {best of L according to H};
repeat
    L := SearchStrategy(L, G(J), X);
    X' := {best of L according to H};
    if  $H(X') \geq H(\text{Solution})$  or ( $H(X') = H(\text{Solution})$ )
        and NumberOfElements(X')
            < NumberOfElements(Solution)
        then Solution := X';
until Stop(H, L);
    
```

V čem se jednotlivé algoritmy liší, je: způsob hledání, mechanismus, kterým vybíráme následující prvek (G), a oceňování příznaků (H). Rozlišujeme exponenciální (náročnost $O(2^n)$), sekvenční (nepřipouští krok zpátky) a náhodný způsob hledání. Následující prvek můžeme vybírat dopředně (začínáme od jednoho a postupně přibíráme), zpětně (prvky odebíráme), kombinovaně (odebíráme i přibíráme) nebo náhodně. Způsobů možného oceňování příznaků je mnoho od vzdálenosti mezi třídami přes nejistotu a nekonzistenci až po

pravděpodobnost chyby, což je vlastností algoritmů modelu Wrapper. Možné kombinace znázorňuje Obrázek 19. [35; 39]



Obrázek 19: Prostor charakteristik FS algoritmů [39]

4.3 Algoritmy selektce příznaků

Z velkého množství existujících algoritmů jsme vybrali 6 algoritmů – 2 typu Wrapper (SFS a SBS) a 4 typu Filter (SFFS, B&BS, mRMR a ReliefF). Algoritmy SFS, SBS a SFFS používají sekvenční metodu prohledávání, B&BS využívá tzv. metodu větvení a mezí, mRMR je založena na vzájemné informaci a ReliefF je algoritmus z rodiny Relief, který používá náhodné prohledávání.

4.3.1 SFS

Sekvenční dopředné hledání je nejjednodušší sekvenční algoritmus společně s SBS. Existuje jako Filter, tak i Wrapper, my ho použijeme v kombinaci s klasifikátorem 1-NN (One Nearest Neighbour) jako typ Wrapper.

SFS začíná s prázdnou množinou a postupně přidává vhodné příznaky. Mějme množinu příznaků X a současný soubor příznaků $X_m (x_1 - x_m)$ – podmnožinu X . Budiž x^+ takový příznak, pro který platí:

$$x^+ = \operatorname{argmax} H^+(X_m, f), \quad (19)$$

kde $H^+(X_m, f)$ je oceňovací funkce, kterou použijeme k ohodnocení množiny X_m s přidaným příznakem f . Nazvěme přidáním ($\text{ADD}(X_m)$) operaci, při které přidáme x^+ k současnému souboru, tedy:

$$\text{ADD}(X_m) \equiv X_m \cup \{x^+\} \quad (20)$$

SFS vybere d nejlepších příznaků tak, že:

$$X_d = \text{ADD}^d(\emptyset) \quad (21)$$

kde ADD^d značí d iterací operace ADD . Číslo d buď zvolíme předem, nebo můžeme nechat optimalizovat algoritmem. Jako funkci $H^+(X_m, f)$ považujeme vybrání takového příznaku, který zaručí největší redukci chyby námi zvoleného klasifikátoru. [34; 38]

4.3.2 SBS

Sekvenční zpětné hledání začíná s celou množinou příznaků a postupně odebírá nejméně vhodné příznaky. Stejně jako u SFS ho použijeme v kombinaci s 1-NN klasifikátorem jako typ Wrapper. Mějme množinu příznaků X a současný soubor příznaků X_m ($x_l - x_m$) – podmnožinu X . Budiž x^- takový příznak, pro který platí:

$$x^- = \text{argmax} H^-(X_m, f), \quad (22)$$

kde $H^-(X_m, f)$ je oceňovací funkce, kterou použijeme k ohodnocení množiny X_m bez příznaku f . Nazvěme odebráním ($\text{REMOVE}(X_m)$) operaci, při které odebereme x^- ze současného souboru, tedy:

$$\text{REMOVE}(X_m) \equiv X_m \setminus \{x^-\} \quad (23)$$

SBS vybere d nejlepších příznaků tak, že:

$$X_d = \text{REMOVE}^d(X) \quad (24)$$

kde REMOVE^d značí d iterací operace REMOVE . Jako funkci $H^-(X_m, f)$ považujeme odebrání takového příznaku, který vytvoří podmnožinu s největší redukcí chyby. Algoritmus může skončit v okamžiku, kdy odebráním každého dalšího prvku zvýšíme chybu, nebo po předem zadaném počtu iterací. [34; 38]

4.3.3 SFFS

SFFS (sekvenční dopředné plovoucí prohledávání) je obdobou SFS s „backtrackingem“. Použijme pojmy definované v (19), (20), (22) a (23). Začínáme s prázdnou množinou. Předpokládejme, že vybíráme opět d příznaků, potom bude algoritmus v k -tém kroku vypadat takto:

1. ADD(X_k)
2. **if** $x_{k+1}^- \neq x_k^+$ REMOVE(X_{k+1}) (25)
else goTo 1.
3. **repeat** 2. **until** $H^-(X_{current}, f) > H(X_{current})$ **then goTo** 1.

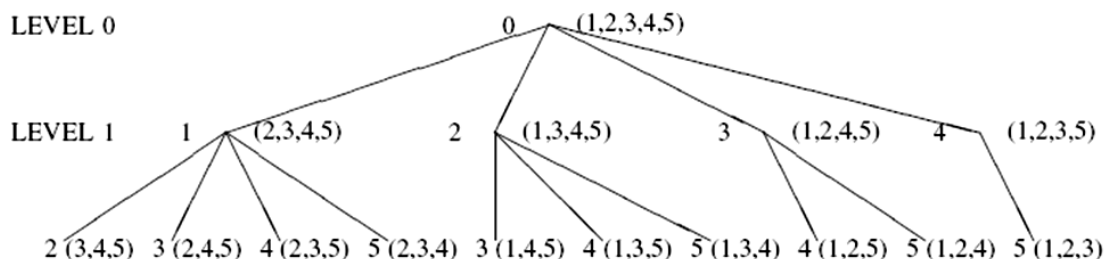
Tedy přidáme prvek, následně odebereme nejhorší prvek (pokud to není ten, co jsme právě přidali). Pokračujeme, dokud jsme schopni H zvyšovat. Algoritmus končí při dosažení požadovaného počtu vybraných příznaků. [38; 42]

4.3.4 B&BS

B&BS používá „Branch and Bound“ tzv. metodu větvení a mezí. Vychází z důležité vlastnosti oceňování, totiž:

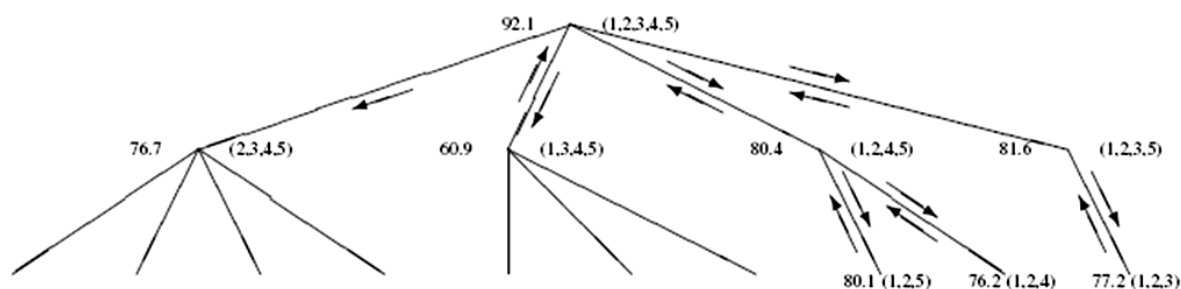
$$X \subset Y \Rightarrow H(X) < H(Y). \quad (26)$$

Algoritmus začíná vytvořením stromu, kdy v nulté úrovni je celá množina příznaků, v první úrovni všechny podmnožiny s odebraným jedním prvkem, v další se dvěma odebranými prvky..., dokud nemáme listy s požadovaným počtem příznaků. Všimněme si, že strom není symetrický (viz Obrázek 20). V každém uzlu máme nalevo napsaný prvek, který jsme odebrali a napravo zbylou množinu.



Obrázek 20: Příklad stromu algoritmu B&B [38]

Takto vytvořený strom nyní začneme prohledávat (viz Obrázek 21). Začneme od listu úplně vpravo. Každý uzel i list má své ohodnocení a my se snažíme nalézt list s tím nejlepším. Narazíme-li na uzel s ohodnocením nižším než nejlepší doposud nalezené, nemusíme již vstupovat do větví z něj vycházejících, protože platí (26).



Obrázek 21: Příklad prohledávání stromu [38]

Tento algoritmus je velmi výpočetně náročný pro velký počet příznaků, protože prohledává celý prostor. Zaručí ovšem na rozdíl od sekvenčních metod, že nalezne nejlepší řešení. [38; 42] Vhodné je kombinovat ho s jinou rychlejší metodou, která vybere podmnožinu příznaků (sníží dimenzi vstupního prostoru), ve které potom B&BS nalezne nejlepší řešení.

4.3.5 mRMR

Algoritmus mRMR vychází z myšlenky, že obecně je minimální klasifikační chyba definovatelná jako maximální statistická závislost třídy c na rozložení dat v prostoru příznaků.

Vyjděme z definice vzájemné informace. Mějme dvě proměnné x a y , jejich vzájemnou informaci $I(x; y)$ určíme z pravděpodobnostních funkcí $p(x)$, $p(y)$ a $p(x, y)$:

$$I(x; y) = \iint p(x, y) \log \frac{p(x, y)}{p(x)p(y)} dx dy. \quad (27)$$

Kritérium maximální závislosti (Max-Dependency) pro soubor příznaků X ($x_1 - x_m$) a třídu y zapíšeme jako:

$$\max D(X, y), \text{ kde } D = I(\{x_i, i = 1 \dots m\}; y). \quad (28)$$

Jelikož bývá problém určit pravděpodobnosti $p(x_1 \dots x_m)$ a $p(x_1 \dots x_m, y)$ z důvodu malého počtu segmentů, je složité toto kritérium implementovat. Místo toho používáme kritérium maximální významnosti (Max-Relevance). Aproximujeme D v rovnici (28) průměrnou hodnotou vzájemných informací $I(x_i, y)$:

$$\max D(X, y), \text{ kde } D = \frac{1}{|X|} \sum_{x_i \in X} I(x_i, y). \quad (29)$$

Lze předpokládat, že takto vybrané třídy budou mít obsahovat redundantní informace, použijeme kritéria minimální redundance (Min-Redundancy), které porovnává dvojice příznaků a snaží se minimalizovat jejich vzájemnou informaci:

$$\min R(X) \text{ , kde } D = \frac{1}{|X|^2} \sum_{x_i, x_j \in X} I(x_i, x_j) \quad (30)$$

Spojíme-li podmínky (29) a (30) dohromady, obdržíme konečnou podmínku mRMR (minimal Redundancy, Maximal Relevance), která může vypadat jako (31) (diference) nebo jako (32) (poměr):

$$\max \Phi(D, R) \text{ , kde } \Phi = D - R \quad (31)$$

$$\max \Phi(D, R) \text{ , kde } \Phi = \frac{D}{R} \quad (32)$$

Podle pokusů popsaných v [43] podmínka (32) dosahuje lepších výsledků. Tento algoritmus lze použít jak samostatně, tak i v kombinaci s jinými algoritmy. V první fázi mRMR vybere tzv. soubor kandidátů, v druhé fázi použijeme jiný algoritmus (např. SFS, SBS), kterému se tímto sníží vstupní dimenze. [43; 44]

4.3.6 Relief a ReliefF

Algoritmy z rodiny Relief jsou založené na náhodném prohledávání. Základní myšlenkou je brát při trénování v potaz nejen vzdálenost mezi třídami (rozdíl v hodnotě příznaků), ale také vzdálenost mezi instancemi.

Základní algoritmus Relief je určen pouze pro dvě třídy. Mějme množinu instancí X ($x_I - x_M$) se souborem příznaků X ($x_I - x_N$) a parametr m , kterým zvolíme velikost náhodné podmnožiny $X_m \subset X$. Označme x_{hit} jako nejbližší prvek k vybranému prvku x_I , pro který platí:

$$y(x_{hit}) = y(x_I), \quad (33)$$

kde y značí třídu, do které prvek náleží. Označme x_{miss} jako nejbližší prvek k vybranému prvku x_I , pro který platí:

$$y(x_{miss}) \neq y(x_I). \quad (34)$$

Pro každý příznak x_i určíme jeho kvalitu $w(x_i)$ tím, že porovnááme vybraný prvek s x_{hit} a x_{miss} následujícím algoritmem:

```

for  $i = 1$  to  $N$  do  $w(x_i) = 0$  end
for  $l = 1$  to  $m$  do
    selectRandomly  $x_l$ 
    find  $x_{hit}$  and  $x_{miss}$ 
    for  $i = 1$  to  $N$  do
        
$$w(x_i) = w(x_i) - \frac{diff(i, x_l, x_{hit})}{m} + \frac{diff(i, x_l, x_{miss})}{m}$$

    end
end
    
```

Přičemž pro dvě instance x_a a x_b definujeme funkci $diff()$ jako:

$$diff(i, x_a, x_b) = \frac{|x_a(i) - x_b(i)|}{\max(x_i) - \min(x_i)} \quad (36)$$

kde $x_a(i)$ značí hodnotu i -tého příznaku instance x_a .

Vylepšení tohoto algoritmu přináší ReliefF. Umožňuje selekci příznaků i pro více než dvě třídy a není tolik citlivý na šum (chyby v hodnotě třídy nebo příznaku). Citlivost odstraňuje tím, že bere vždy n nejbližších instancí. Máme-li C tříd a jejich pravděpodobnosti $p(y)$, potom bude algoritmus vypadat následovně:

```

for  $i = 1$  to  $N$  do  $w(x_i) = 0$  end
for  $l = 1$  to  $m$  do
    selectRandomly  $x_l$  (with class  $y(x_l)$ )
    for  $y = 1$  to  $C$  do
        find  $n$  nearest instances  $x_{y,j}$  from class  $y$ ,  $j = 1 \dots n$ 
        for  $i = 1$  to  $N$  do
            for  $j = 1$  to  $n$  do
                if  $y = y(x_l)$ 
                    
$$w(x_i) = w(x_i) - \frac{diff(i, x_l, x_{y,j})}{m \cdot n}$$

                else  $w(x_i) = w(x_i) + \frac{p(y)}{1 - p(y(x_l))} \frac{diff(i, x_l, x_{y,j})}{m \cdot n}$ 
            end
        end
    end
end
    
```

Přičemž funkci $diff()$ definujeme jako (36), pokud se nemáme instance, kde hodnota některého příznaku není známa. [36; 45; 46]

5 Experimentální část

Cílem experimentu je porovnat různé algoritmy selekce příznaků a klasifikátory s ohledem na jejich vhodnost pro úlohu klasifikace novorozeneckého spánku. Druhým cílem je porovnat výsledky selekce jednotlivých algoritmů a určit, zda lze vybrat několik málo příznaků, které by pro naši úlohu byly dostačující – jako je tomu například u klasifikace spánku dospělých. V závěru práce jsou tyto příznaky a jejich vhodnost vyzkoušeny na datasetu, který nebyl použit k jejich zjištění – tedy na nezávislém datasetu.

5.1 Použitý software

Celý experiment byl prováděn v programovém prostředí MatLab (viz 5.1.1) s využitím toolboxu PRTools (viz 5.1.1.3), algoritmů z repozitáře Arizonské Univerzity (viz 5.1.1.1), algoritmu mRMR (viz 5.1.1.2) a softwaru WEKA (viz 5.1.2), ke spuštění tohoto softwaru z prostředí MatLab byly použity kódy od Matta Dunhama [47]. Vše bylo integrováno do několika souborů spustitelných v MatLabu, které jsou uvedené v příloze a přiložené k práci. Vstupem byla data ve formátu arff (viz 5.1.2.1), výstupem data ve formátu arff a mat.

5.1.1 MatLab

MatLab (původně zkratka Matrix Laboratory) je interaktivní systém založený na maticovém počtu. Je licencován společností The MathWorks, Inc. [48] Systém obsahuje vlastní interpret jazyku MATLAB, ve kterém lze definovat nové funkce nebo vytvořit „dávkové soubory“ [49], které volají již existující funkce. Velmi důležitou částí jsou knihovny funkcí, tzv. toolboxy. Každý toolbox je ve skutečnosti adresář se soubory a zpracovává nějaký určitý obor nebo úlohu, pro které nachází MatLab uplatnění.

5.1.1.1 ASU Feature Selection Repository

Tento repozitář shromažďuje nejpopulárnější algoritmy selekce příznaků a několik klasifikátorů a algoritmů shlukové analýzy. Algoritmy jsou implementovány pro MatLab. V repozitáři je uložena i jejich dokumentace a reference na publikované články. Zároveň zde nalezneme jejich vzorové použití (včetně vypočtené výkonnosti) na několika datasetech, které jsou také k dispozici pro zhodnocení a porovnání vlastních algoritmů. [50; 51]

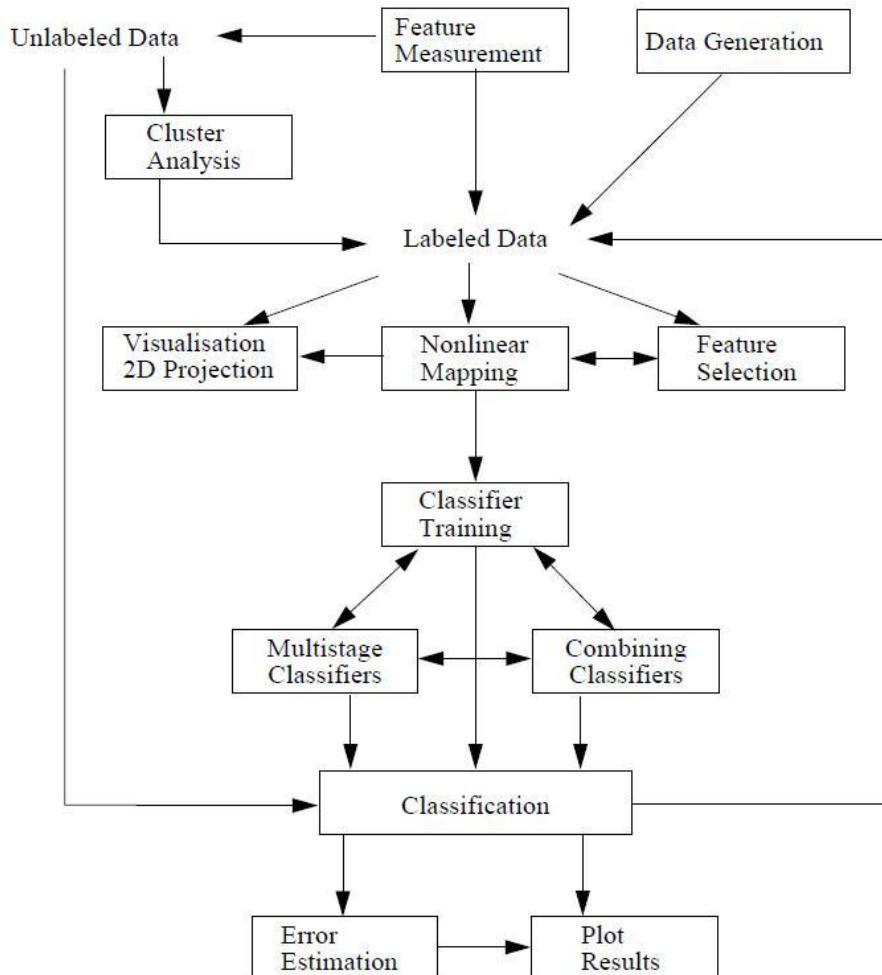
5.1.1.2 mRMR a Mutual Information Toolbox

Hanchuang Peng, Fuhui Long a Chris Ding – vědci z laboratoře Lawrence Berkeleyho na Kalifornské Univerzitě vyvinuli algoritmus spojující kritérium maximální relevance a

minimální redundance (odtud název mRMR). Algoritmus implementovali v C/C++ a v MatLabu, přičemž vytvořili tzv. Mutual Information toolbox (knihovnu funkcí k určování vzájemné informace), který je algoritmem využíván. [44; 52]

5.1.1.3 PRTools

PRTools je volně přístupná knihovna funkcí pro rozpoznávání vzorů (Pattern Recognition). Skupina zabývající se rozpoznáváním vzorů na Technické univerzitě Delft v Holandsku (The Pattern Recognition Research Group of the TU Delft) vyvinula tuto knihovnu a je jejím vlastníkem. [53] Knihovna obsahuje komplex funkcí umožňující provádět celý proces klasifikace nebo jeho části. Základní strukturou je tzv. dataset, který se skládá z objektů reprezentujících matici příznaků. [54] Přehled jednotlivých částí knihovny je zobrazen v následujícím diagramu (viz Obrázek 22).



Obrázek 22: Přehled jednotlivých částí toolboxu PRTools [54]

5.1.2 WEKA

WEKA (Waikato Environment for Knowledge Analysis) je software vyvíjený na univerzitě Waikato v Hamiltonu na Novém Zélandě za finanční podpory tamní vlády. Jedná se o sbírku algoritmů strojového učení (Machine Learning) k získávání užitečné informace z dat (úlohy Data Miningu). Celý software je napsán v programovacím jazyku Java a distribuován pod licencí GNU Public licence. [55]



Obrázek 23: Logo projektu WEKA [55]

WEKA obsahuje nástroje pro předzpracování dat, jejich klasifikaci, regresní, shlukovou či asociační analýzu a v neposlední řadě i jejich vizualizaci. [56] Je určena pro výzkumné účely i pro výuku nebo vývoj nových aplikací. Tento software jsme použili především pro úpravu vstupních dat.

5.1.2.1 Arff formát

Jedním z možných vstupních formátů dat do WEKy je formát ARFF (Attribute-Relation File Format), který byl vyvinut přímo se softwarem. Vlastní soubor je typu ASCII a je rozdělen do dvou částí – hlavičky a samotných dat. [57] Hlavička nese název relace, seznam atributů a jejich typy. Její řádky jsou uvozeny deklarací „@relation“ respektive „@attribute“. Blok dat je uvozen deklarací „@data“, po níž následují samotná data v řádcích (viz Obrázek 24).

```
@RELATION FEATURES

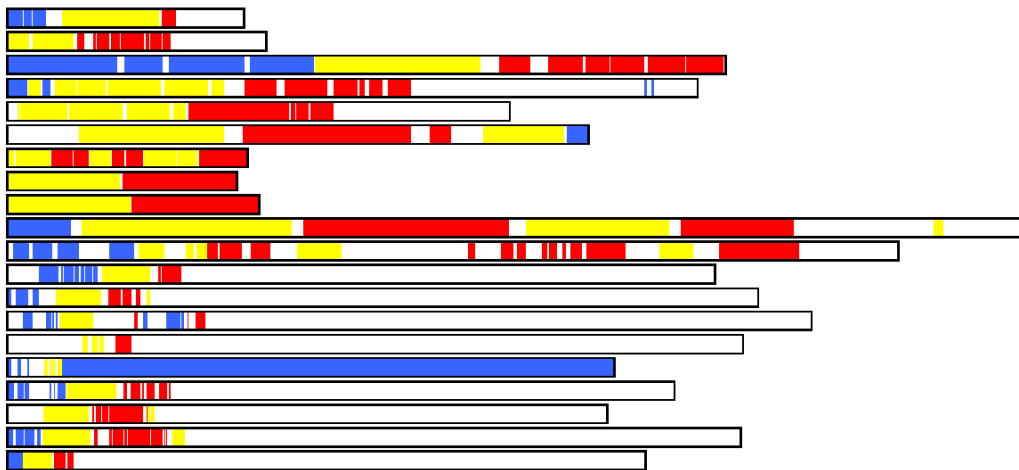
@ATTRIBUTE ch_FP1---min_value REAL
@ATTRIBUTE ch_FP1---max_value REAL
@ATTRIBUTE ch_FP1---mean REAL
@ATTRIBUTE ch_FP1---std REAL
@ATTRIBUTE ch_FP1---1st_diff_mean REAL...

@DATA
-37.500 28.120 -2.869 11.631 -1.032 ...
-46.880 73.440 14.588 30.927 ...
-50.000 43.750 -13.110 16.493 ...
...
```

Obrázek 24: Struktura souboru ve formátu arff

5.2 Data

Jako data budeme používat novorozenecké záznamy z Ústavu pro péči o matku a dítě (ÚPMD), které hodnotil MUDr. Karel Paul. Máme k dispozici 20 záznamů od 20 různých pacientů podobného věku (donošení novorozenci). Z každého je extrahován soubor 547 příznaků (viz Příloha 1) a následně jsou převedeny do formátu arff. Vyskytují se v nich 4 třídy: 0 – bdělost, 1 - quiet sleep (klidný spánek), 2 - active sleep (aktivní spánek) a 3 – blíže neurčeno, které jednotlivým segmentům přiřadil MUDr. Paul. Pro náš experiment jsou důležité pouze třídy 1 a 2. Protože nejsou obě třídy v záznamech stejně zastoupeny (viz Obrázek 25), vybereme maximální počet segmentů tak, aby měly obě třídy stejné zastoupení, a odstraníme segmenty tříd 0 a 3. Takto upravená data tvoří vstup pro samotnou selekci příznaků.



Obrázek 25: Přehled zastoupení jednotlivých tříd v záznamech: modrá – 0, žlutá – 1, červená – 2, bílá – 3

Tabulka 2: Velikosti výsledných datasetů (počet segmentů)

Dataset	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Počet segmentů	100	562	1192	1015	1010	1355	76	19	16	2290	874	634	747	434	480	333	1125	1444	1590	506

5.3 Struktura experimentu

Experiment se skládá ze dvou částí: výpočtu výkonnostních křivek jednotlivých algoritmů selekce a ze samotné selekce určeného počtu příznaků. Použity byly algoritmy popsané v kapitolách 3.5.1 a 4.3. Branch&Bound algoritmus byl použit pouze omezeně z důvodu nedostatku velikosti haldy pro větší datasety. Dále jsme vyzkoušeli délku trvání kombinace 2 algoritmů, kdy byl trojnásobek požadovaného počtu příznaků selektován rychlejším algoritmem (ReliefF nebo mRMR) a pro finální selekci bylo použito SBS.

5.3.1 Výkonnostní křivky

Prvním krokem byl výpočet výkonnostních křivek – závislostí na počtu selektovaných příznaků a na počtu segmentů datasetu - pro jednotlivé algoritmy a klasifikátory. Jeho výsledkem bylo určení ideálního počtu příznaků, který pro všechny algoritmy minimalizuje klasifikační chybu. Použity byly 4 různé klasifikátory, jejichž chyba byla určena jednak pomocí Hold-Out validace, tak i pomocí 10-Fold-Cross-Validation s jedním opakováním. Tyto výpočty byly provedeny na 4 různě velkých datasetech (16, 100, 480 a 1192 segmentů). Zajímalo nás také, zda existuje nějaká kombinace algoritmus selekce – klasifikátor, která by vždy dosahovala nejlepších výsledků.

Druhým výsledkem tohoto kroku bylo porovnání výpočetní náročnosti jednotlivých algoritmů v závislosti na počtu selektovaných příznaků a na velikosti datasetu. Pro tyto závislosti byla použita doba potřebná k nalezení 20 příznaků. Výpočty byly prováděny na dvou různých počítačích – s procesorem Intel®Core™ i5-2400s, respektive Intel®Core™ i7-930. Oba počítače byly během výpočtů běžně užívány. Abychom byli schopni porovnat dobu trvání jednotlivých algoritmů na obou počítačích, jsou údaje vztažené na dobu trvání nalezení 1 příznaku metodou SFS na jednom konkrétním datasetu - cca 5, respektive 10s. Tyto údaje jsou označené jako „relative time“.

5.3.2 Selektce zvoleného počtu příznaků

V druhém kroku jsme vybrali z 19 datasetů všemi algoritmy zvolený počet příznaků. Sledováno bylo, jak se shodují výsledky různých algoritmů na jednotlivých datasetech a jak se shodují výsledky jednoho algoritmu na různých datasetech. Nakonec jsme se pokusili vybrat příznaky, které byly nejčastěji voleny, a ověřit jejich vhodnost na nezávislém, dvacátém datasetu.

5.4 Výsledky a diskuze

5.4.1 Výkonnost algoritmů

Z důvodu přílišné paměťové náročnosti algoritmu SBBS jsme nakonec porovnávali pouze 6 algoritmů – mRMR MID a MIQ, ReliefF, SFFS (typ Filter) a SBS a SFS (typ Wrapper). Obecně můžeme říci, že algoritmy typu Wrapper jsou řádově časově náročnější, ačkoli nedosahují řádově lepších výsledků, ale výsledků srovnatelně kvalitních (viz Obrázek 28). Z těchto algoritmů dosahuje lepších výsledků zpětné hledání - SBS, které lépe minimalizuje klasifikační chybu, ovšem je výrazně pomalejší než hledání dopředné - SFS (viz Obrázek 37 a Obrázek 38).

Jako nejrychlejší se ukazují algoritmy mRMR (viz Tabulka 5), pro větší počet selektovaných příznaků algoritmus ReliefF (viz Obrázek 42). Zajímavostí algoritmu SFFS je, že jeho časová náročnost je téměř nezávislá na velikosti datasetu (viz Obrázek 35), což mu poskytuje výhodu oproti ostatním algoritmům.

Kapitoly 5.4.1.1 a 5.4.1.2 popisují blíže výkonnost algoritmů z hlediska klasifikační chyby, respektive výpočetní náročnosti.

5.4.1.1 Výkonnost z hlediska klasifikační chyby

Klasifikační chybu jednotlivých klasifikátorů jsme určovali jednak pomocí Hold-Out validace, tak i metodou Cross-Validation. Z tabulek zobrazujících klasifikační chybu jednotlivých klasifikátorů (Tabulka 3 a Tabulka 4) vidíme, že Hold-Out validace přináší zkreslené výsledky. Příliš záleží na rozdělení dat na trénovací a testovací množinu, většinou jsme získali pesimistické výsledky, ačkoli někdy vyšel výsledek překvapivě dobrý. Nejvíce volba rozdělení dat ovlivňuje klasifikátor k nejbližším sousedům (k-NN). Dále proto budeme uvažovat pouze chybu určenou pomocí CV, která je přesnější (více odpovídá skutečnosti).

Nejlepších výsledků dosahuje většinou klasifikátor Naive Bayes (viz Obrázek 26 a Obrázek 27). Z následujících tabulek (Tabulka 3 a Tabulka 4), vidíme, že tento klasifikátor dosahuje nejnižší klasifikační chyby jak pro nejlepší (max. 9%), tak i pro nejhorší možný algoritmus selekce na daném datasetu. Klasifikátor SVM dosahuje vysoké klasifikační chyby, v nejhorších případech – až 48%.

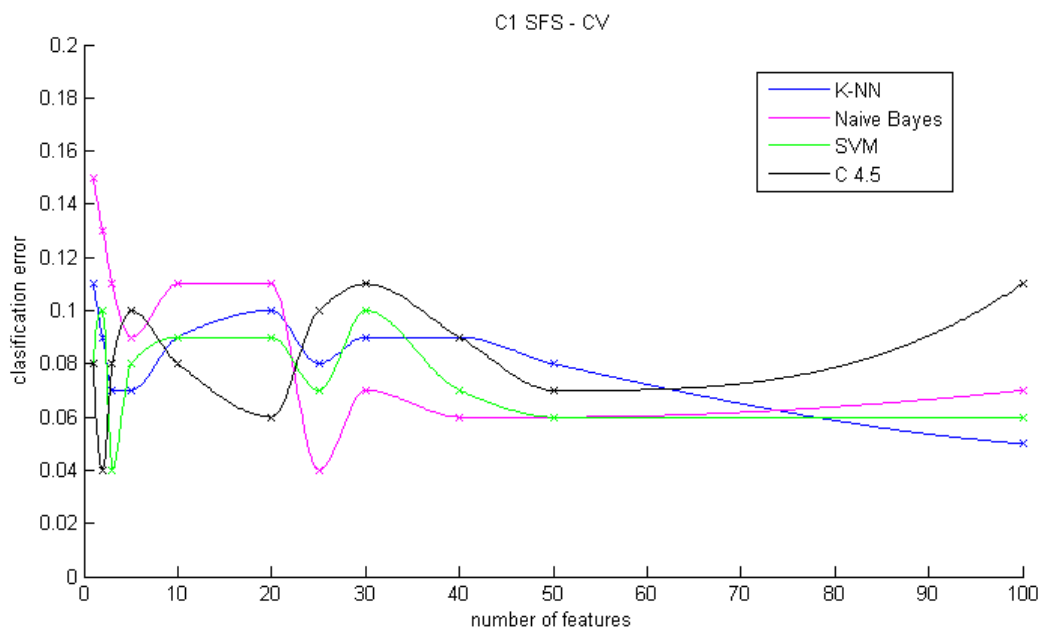
Z důvodu přílišné výpočetní náročnosti jsme klasifikaci klasifikátorem C4.5 provedli pouze na datasetu 1, kde dosáhl výsledků srovnatelných s klasifikátorem Naive Bayes.

Tabulka 3: Klasifikační chyby jednotlivých klasifikátorů pro nejlepší algoritmus selekce pro 4 různé datasey

	Cross-Validation			Hold-Out validace		
	Min	Max	Ø	Min	Max	Ø
Naive Bayes	3%	9%	6%	10%	15%	12%
k-NN	4%	12%	8%	18%	22%	20%
SVM	4%	6%	5%	13%	19%	17%

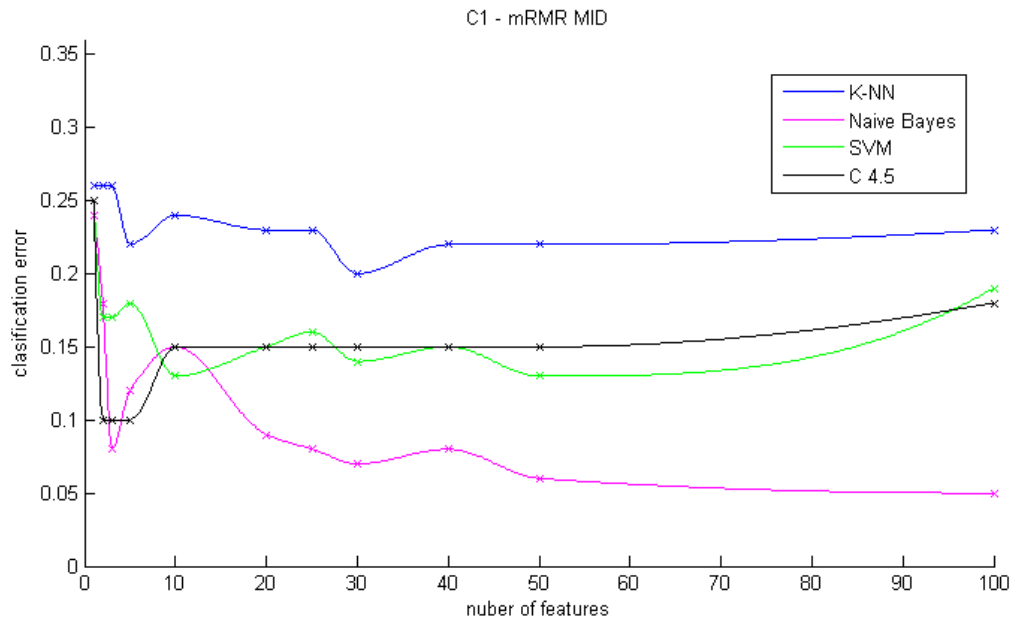
Tabulka 4: Klasifikační chyby jednotlivých klasifikátorů pro nejhorší algoritmus selekce pro 4 různé datasey

	Cross-Validation			Hold-Out validace		
	Min	Max	Ø	Min	Max	Ø
Naive Bayes	13%	42%	29%	15%	38%	27%
k-NN	12%	40%	32%	23%	39%	30%
SVM	11%	48%	33%	19%	52%	37%

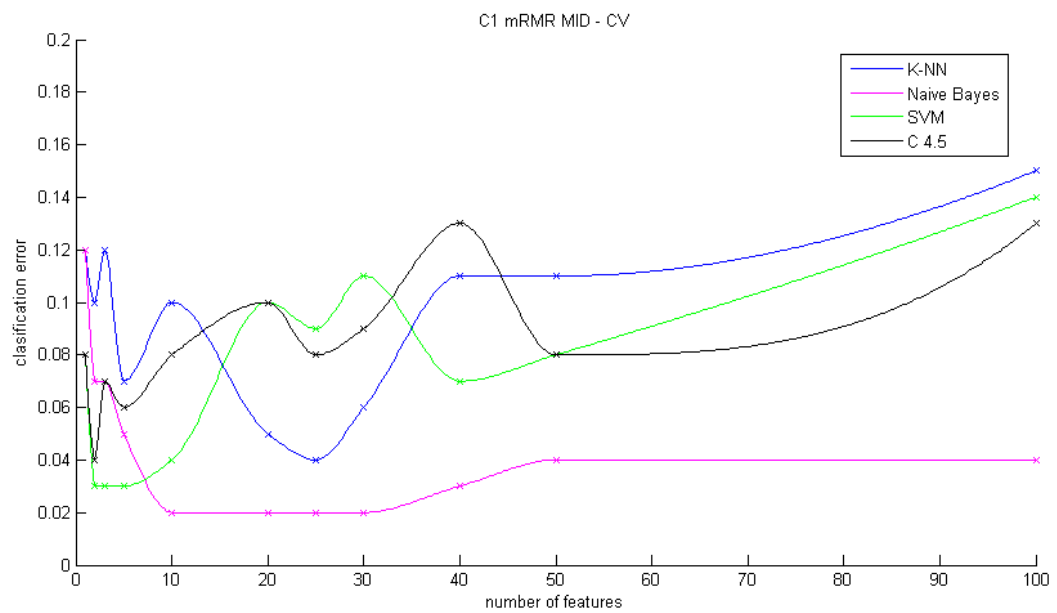


Obrázek 26: Závislost klasifikační chyby na počtu selektovaných příznaků; dataset 1, algoritmus SFS; porovnání různých klasifikátorů

a)



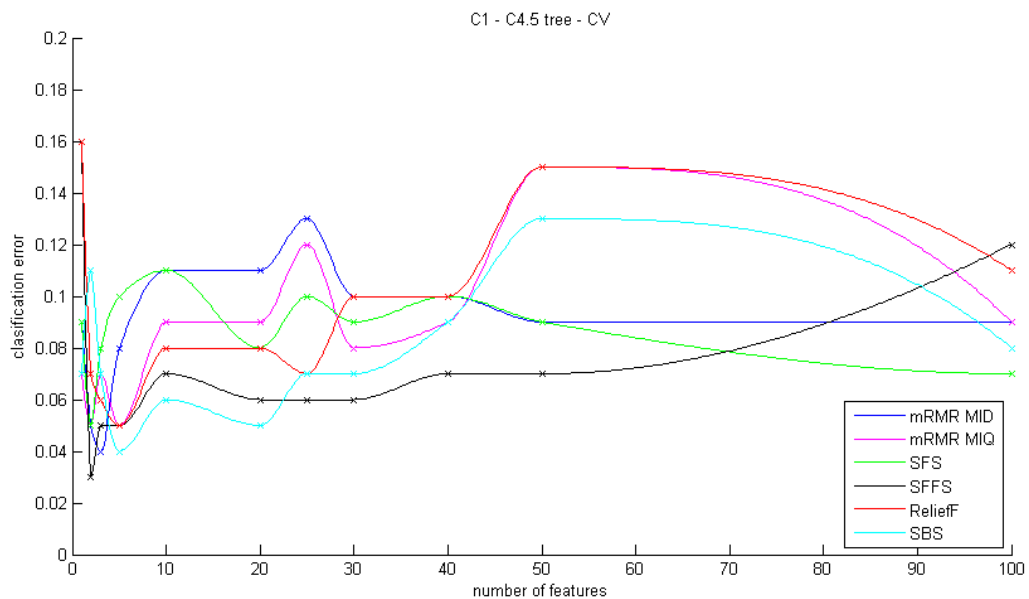
b)



Obrázek 27a, b: Závislost klasifikační chyby na počtu selektovaných příznaků; dataset 1, algoritmus mRMR MID; porovnání různých klasifikátorů; a) Hold-Out validace b) Cross-Validation

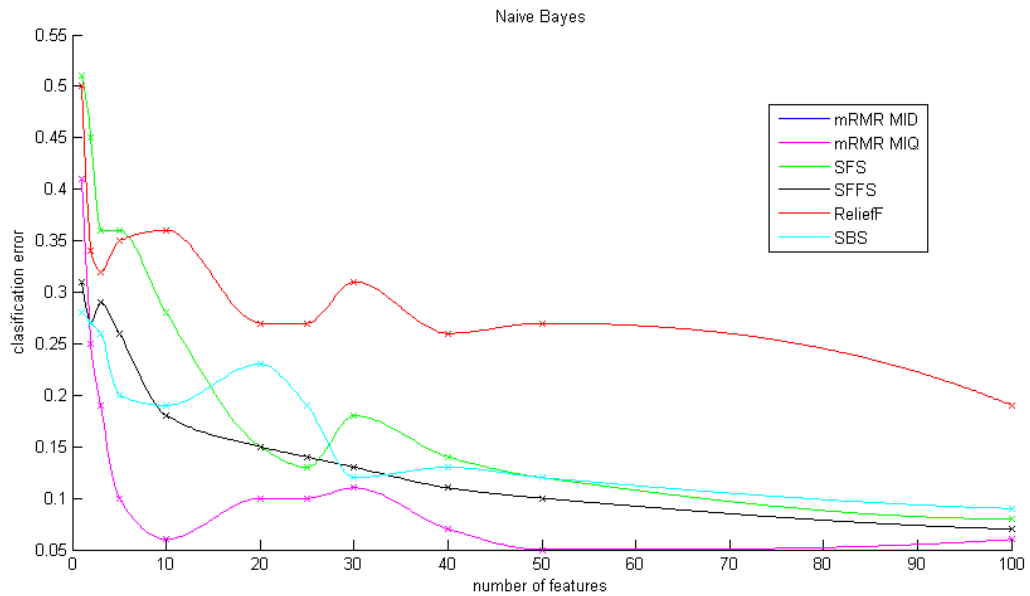
Porovnáním výkonnostních křivek jednotlivých kombinací klasifikátor – algoritmus selekce (viz Obrázek 28 až Obrázek 31) nám jako nejlepší kombinace vyšel klasifikátor Naive Bayes v kombinaci se SFFS (chyba do 10%), časově méně náročnou variantou je kombinace s algoritmem mRMR, která dosahuje nepatrně horších výsledků (chyba až 15%). Klasifikátor k-NN nejlépe využívá příznaky selektované zpětnou selekcí (SBS). Tyto příznaky jsou také nejvhodnější pro klasifikátor SVM, ačkoli tento umí dobře pracovat i s příznaky vybranými pomocí SFFS. Stromový klasifikátor C4.5 využívá vhodně příznaky všech tří sekvenčních metod (SFS, SBS, SFFS) i algoritmu ReliefF.

Chceme-li porovnáním stejných křivek zvolit nejlepší algoritmus selekce, vybereme algoritmy SFFS a SBS. Obecně nejhorších výsledků dosahujeme s algoritmem ReliefF, který se spíše hodí k určení váhovacích konstant jednotlivých příznaků než k jejich selekci. Dopředné hledání (SFS) dosahuje většinou průměrných výsledků. Algoritmus mRMR je vhodné používat v kombinaci s klasifikátorem Naive Bayes, protože oba algoritmy používají obdobných statistických výpočtů.

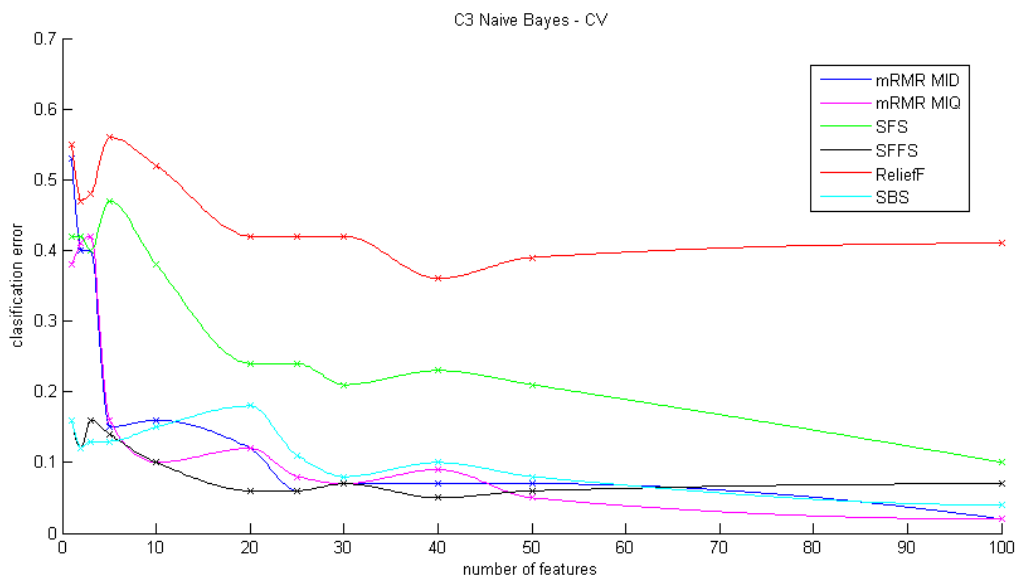


Obrázek 28: Závislost klasifikační chyby na počtu selektovaných příznaků; dataset 1, klasifikátor C4.5; porovnání různých algoritmů selekce; Cross-Validation

a)

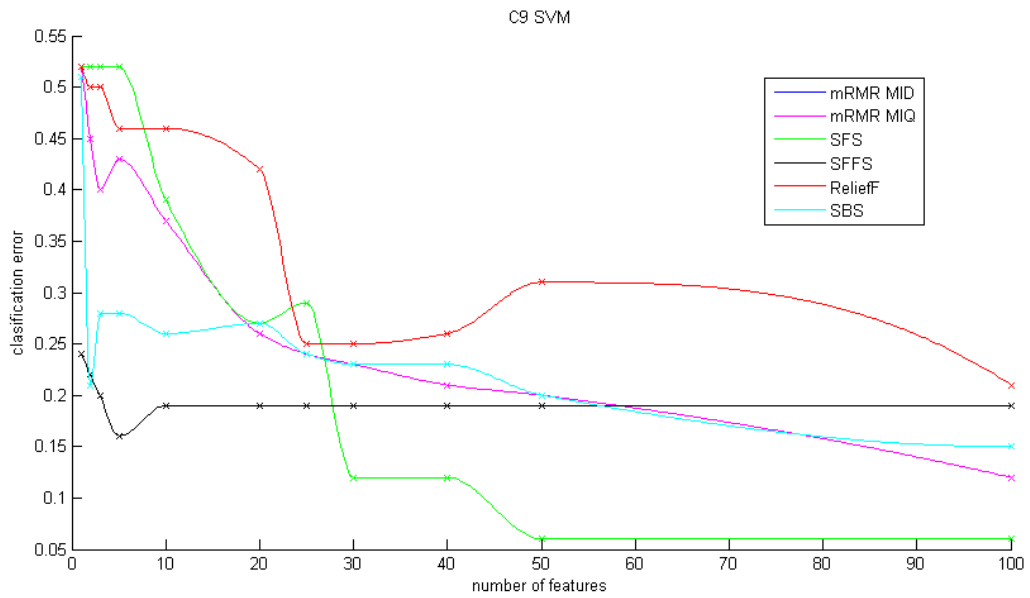


b)

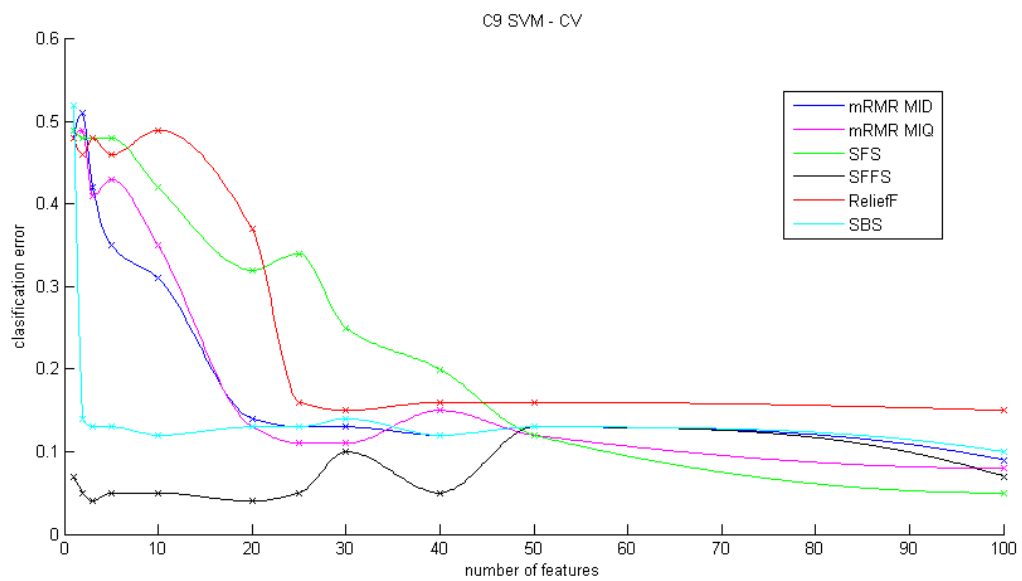


Obrázek 29a, b: Závislost klasifikační chyby na počtu selektovaných příznaků; dataset 3, klasifikátor Naive Bayes; porovnání různých algoritmů selekce; a) Hold-Out validace b) Cross-Validation

a)

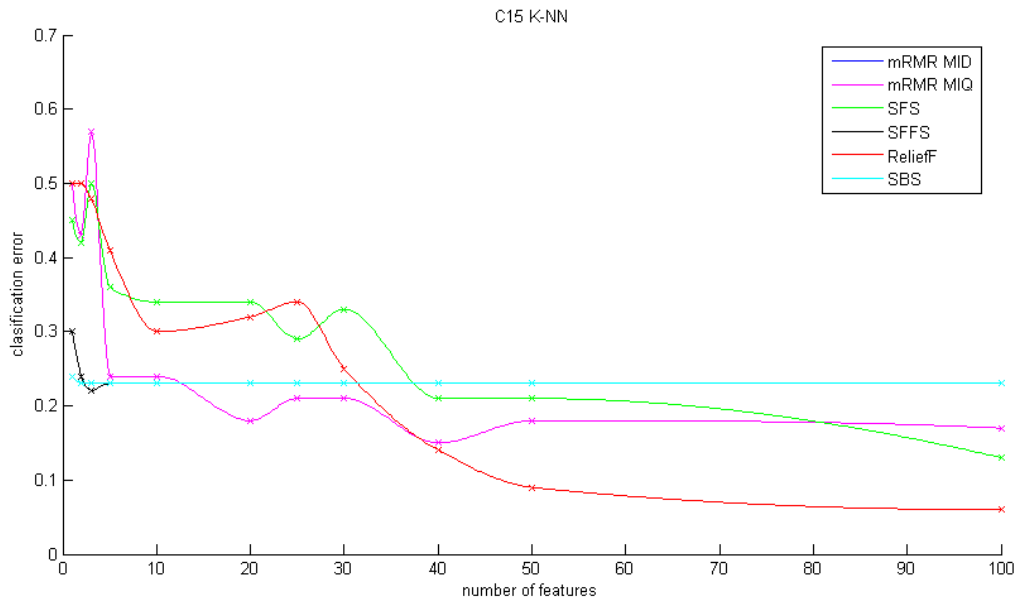


b)

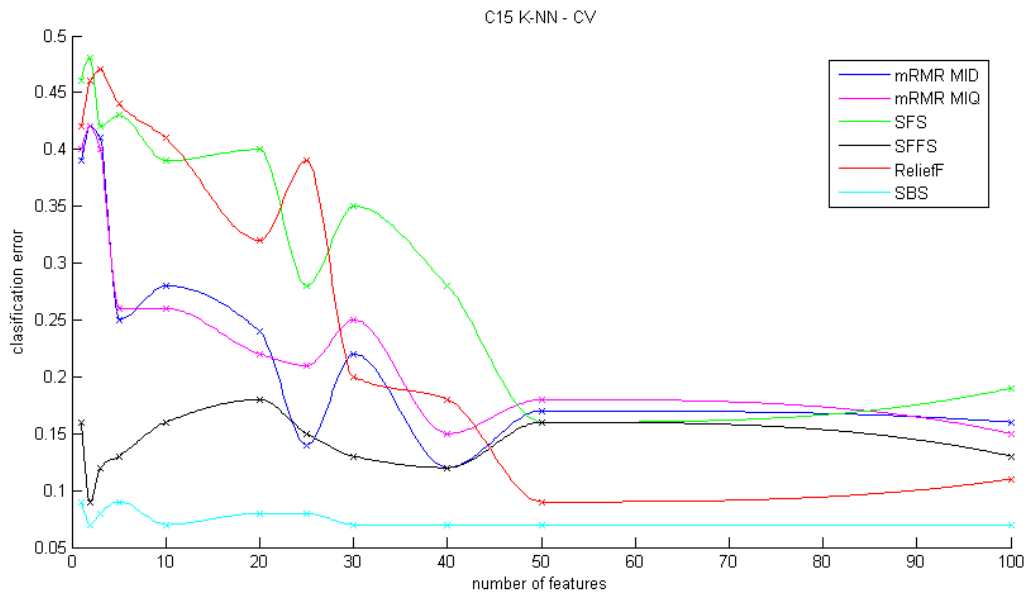


Obrázek 30a, b: Závislost klasifikační chyby na počtu selektovaných příznaků; dataset 9, klasifikátor SVM; porovnání různých algoritmů selekce; a) Hold-Out validace b) Cross-Validation

a)



b)



Obrázek 31a, b: Závislost klasifikační chyby na počtu selektovaných příznaků; dataset 15, klasifikátor k-NN; porovnání různých algoritmů selekce; a) Hold-Out validace b) Cross-Validation

5.4.1.2 Výpočetní náročnost

Tabulka 5 a Tabulka 6 ukazují čas v sekundách potřebný k výpočtu jednotlivých selekcí. Z první tabulky vidíme, že nejrychlejší je algoritmus mRMR, který dokázal selektovat příznaky z každého datasetu pod 2s. O řád pomalejší je SFFS s nejhorsím časem pod 1 minutu. Algoritmus ReliefF dosáhl nejvyššího času 20 minut. Oba algoritmy typu Wrapper jsou nejpomalejší – SFS s časy pod 1,5 hodiny, SBS s časy řádově v hodinách. Tyto algoritmy bylo z časového hlediska výhodnější kombinovat s algoritmem mRMR (viz Obrázek 39). Druhá tabulka je zde uvedena pouze pro orientaci.

Tabulka 5: Časová náročnost selekce 20 příznaků pomocí jednotlivých algoritmů – dataset 1

Algoritmus	SFS	SBS	SFFS	ReliefF	mRMR MID	mRMR MIQ	SBBS	ReliefF + SBS	MID + SBS	MIQ + SBS
Doba výpočtu (s)	120,7	1862,2	51,6	5,9	0,6	0,6	11,0	19,4	20,9	22,2

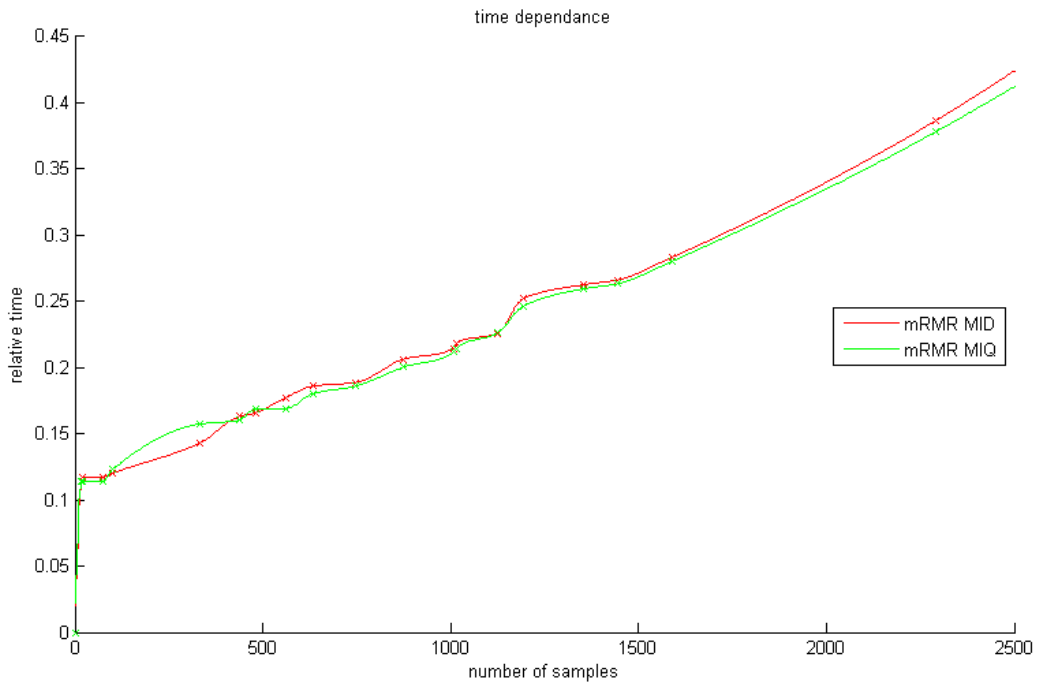
Tabulka 6: Časová náročnost selekce 20 příznaků pomocí algoritmu SFS pro různé datasety

Dataset	1	2	3	4	5	6	7	8	9	10
Počet segmentů	100	562	1192	1015	1010	1355	76	19	16	2290
Doba výpočtu (s)	120,7	398,0	1374,1	1021,6	1025,5	1774,0	114,8	91,0	98,5	4291,2

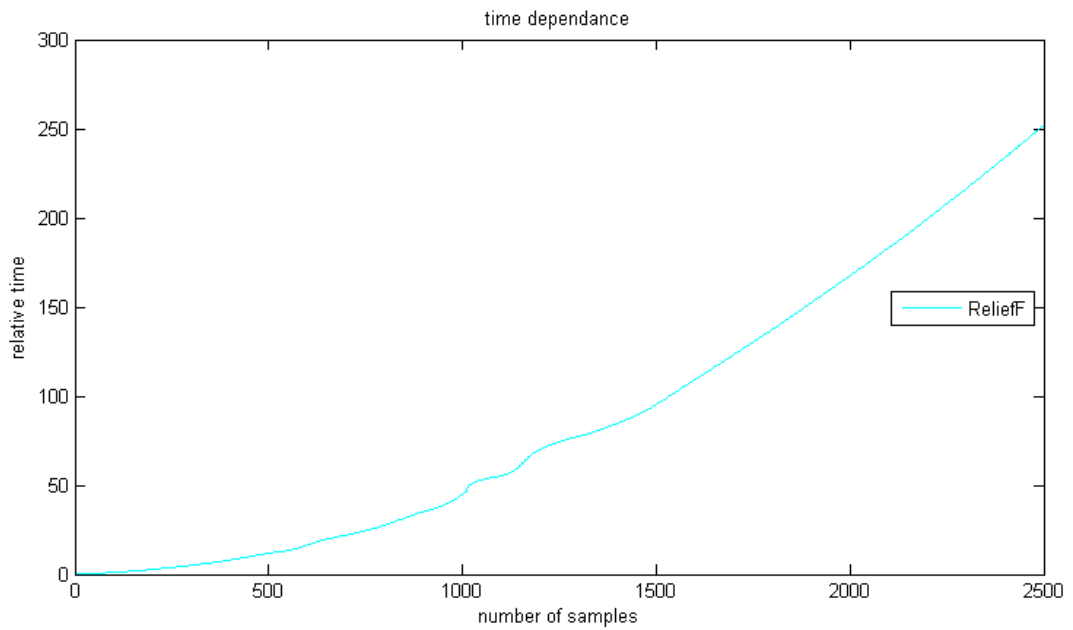
Závislost doby výpočtu na počtu segmentů datasetu

Algoritmus mRMR je rychlý i pro velké datasety, závislost je zhruba lineární a je téměř shodná pro MIQ i MID variantu (viz Obrázek 32). ReliefF má pro větší datasety (>20 segmentů) exponenciální závislost, což je nežádoucí vlastnost (viz Obrázek 33a v detailu Obrázek 34). Obdobný tvar průběhu mají i algoritmy SFS (viz Obrázek 36 a Obrázek 37) a SBS (viz Obrázek 38). Algoritmus SFFS má téměř konstantní závislost (viz Obrázek 35), což mu poskytuje oproti ostatním algoritmům jistou výhodu.

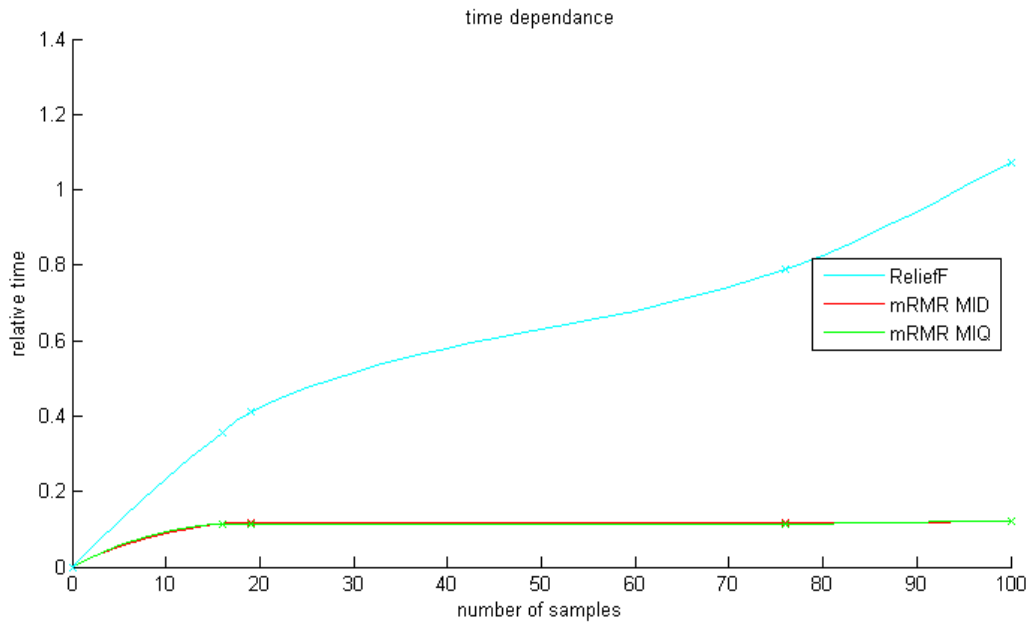
Pro naši úlohu by závislost na velikosti datasetu měla být nejhůře lineární, exponenciální závislost je nevhodná, protože potřebujeme zpracovávat velké datasety. Z tohoto hlediska jsou tedy vhodné algoritmy SFFS a mRMR.



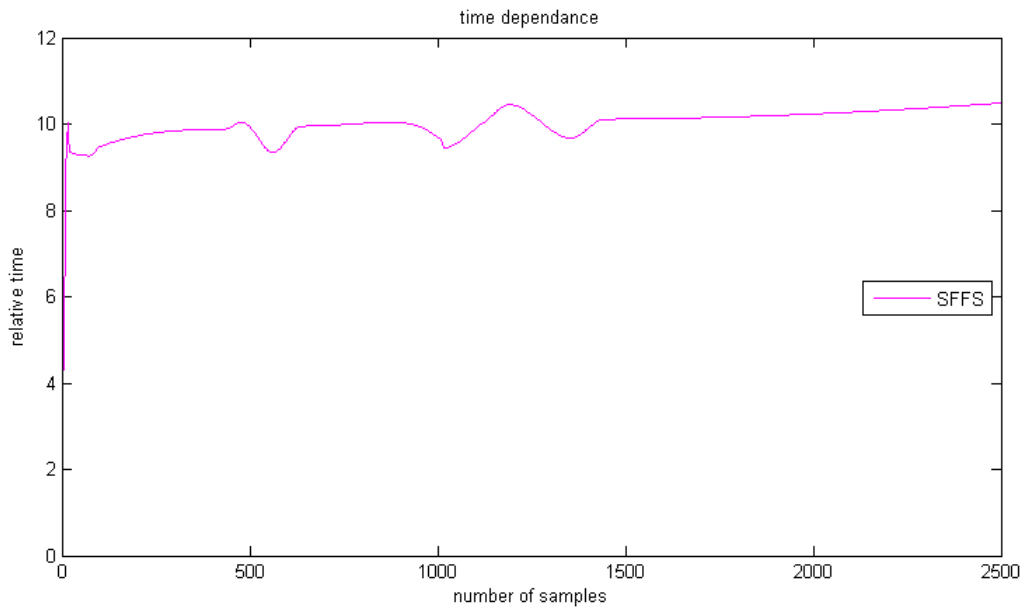
Obrázek 32: Vztah mezi dobou výpočtu a počtem segmentů datasetu - mRMR



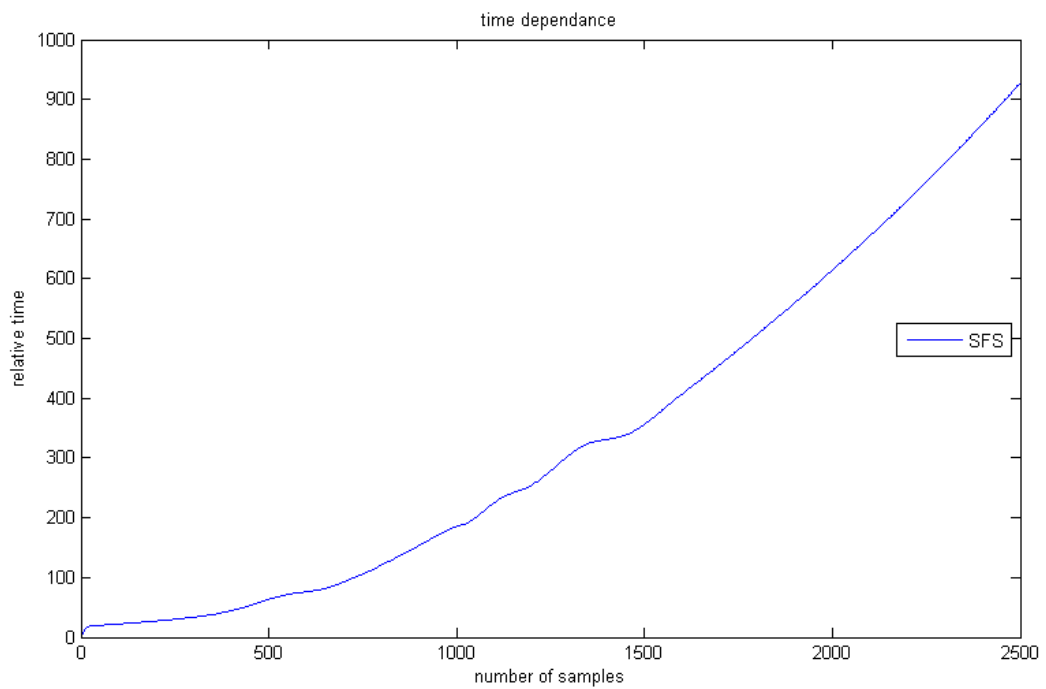
Obrázek 33: Vztah mezi dobou výpočtu a počtem segmentů datasetu – ReliefF



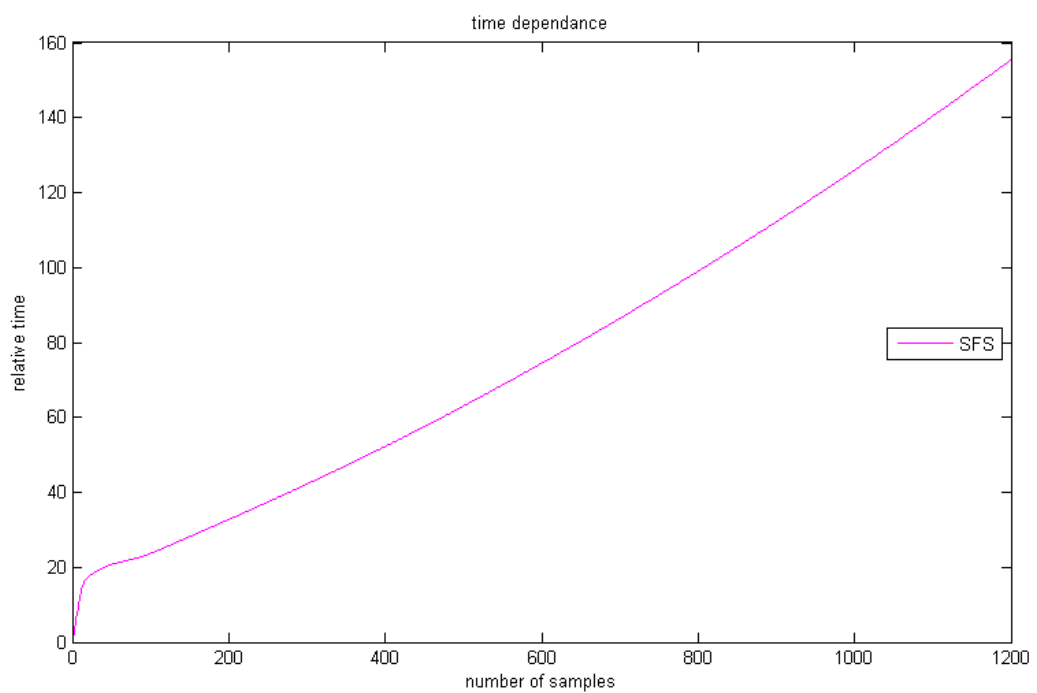
Obrázek 34: Vztah mezi dobou výpočtu a počtem segmentů datasetu – ReliefF a mRMR - detail



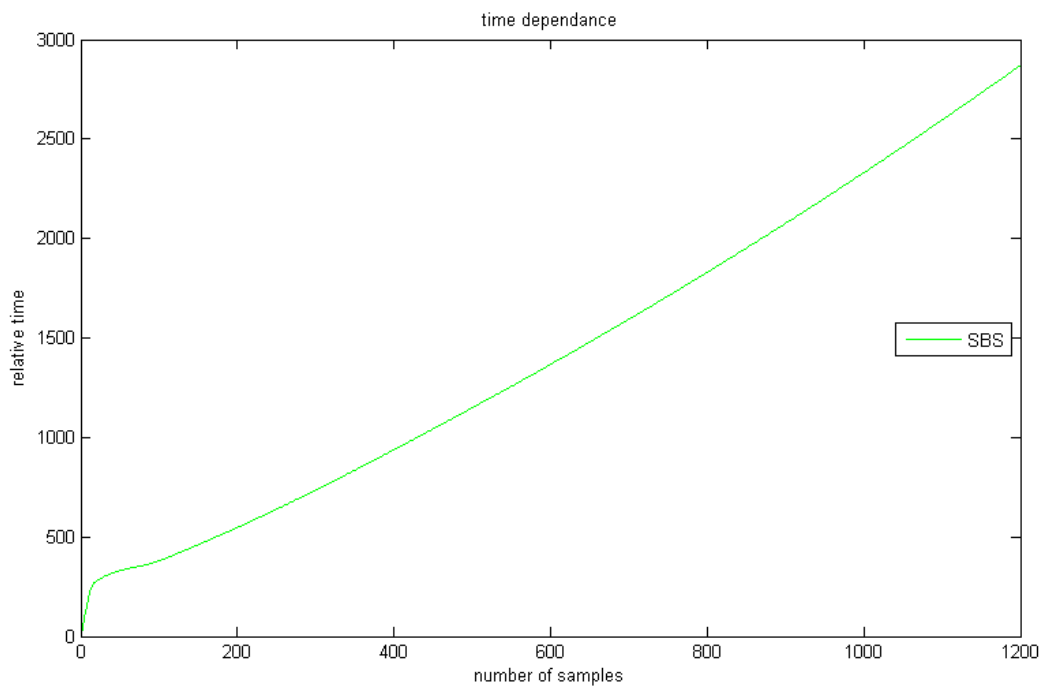
Obrázek 35: Vztah mezi dobou výpočtu a počtem segmentů datasetu - SFFS



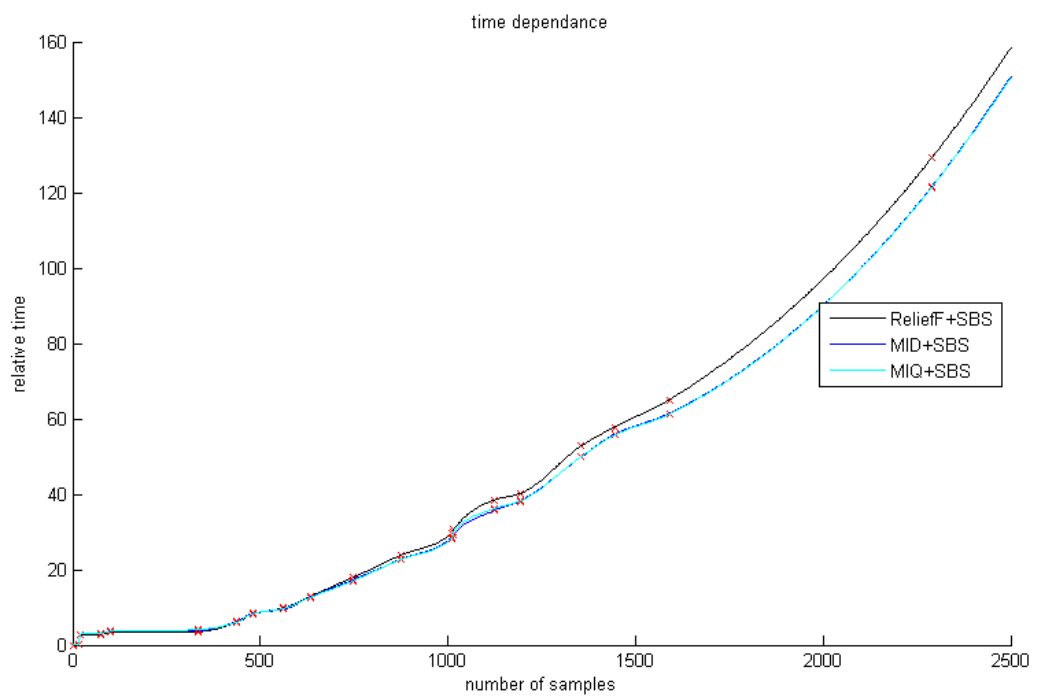
Obrázek 36: Vztah mezi dobou výpočtu a počtem segmentů datasetu - SFS



Obrázek 37: Vztah mezi dobou výpočtu a počtem segmentů datasetu – SFS - detail



Obrázek 38: Vztah mezi dobou výpočtu a počtem segmentů datasetu – SBS - detail

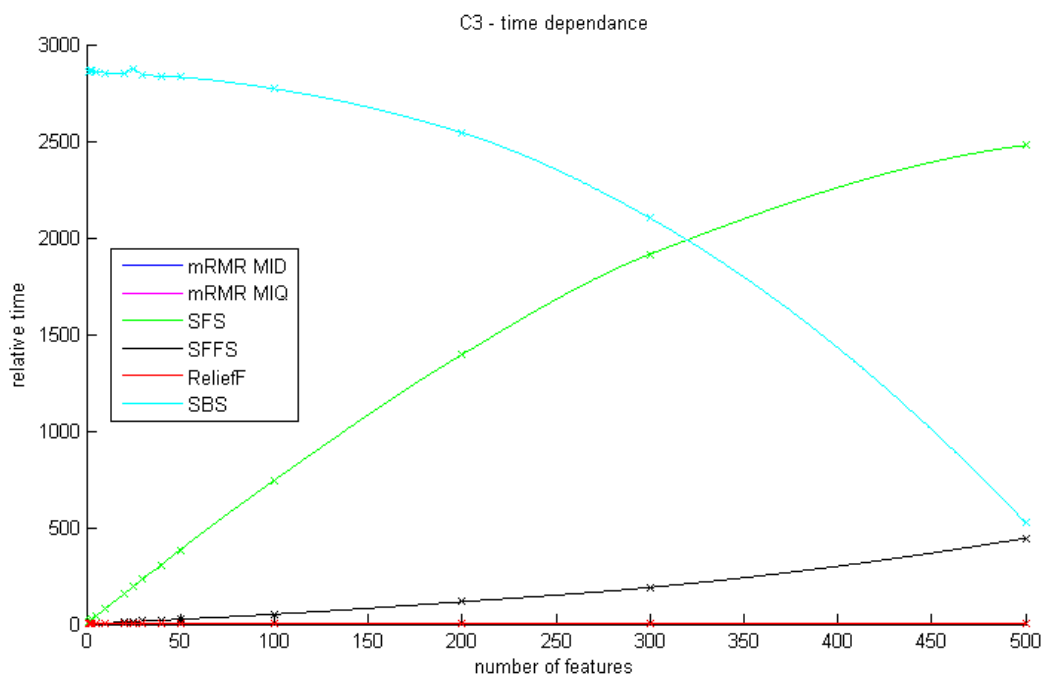


Obrázek 39: Vztah mezi dobou výpočtu a počtem segmentů datasetu – kombinace algoritků

Závislost doby výpočtu na počtu příznaků

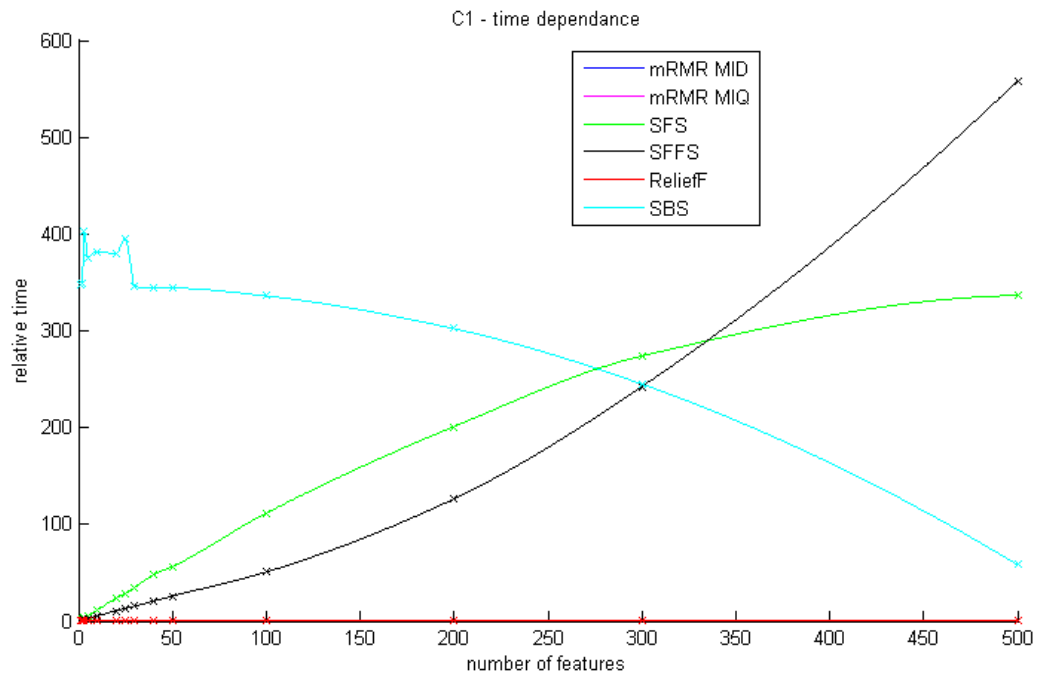
Algoritmus ReliefF (viz Obrázek 42) počítá vždy koeficienty pro všechny příznaky, proto je jeho závislost konstantní. Je tedy relativně pomalý pro malý počet příznaků (<70), zároveň ale vhodný pro selekci většího počtu příznaků. Porovnáme-li ho s algoritmem mRMR, který má logaritmickou závislost, vidíme, že se křivky protínají někde okolo 70 příznaků. I křivky příslušející algoritmům SBS a SFS mají logaritmický průběh (viz Obrázek 40 a Obrázek 41a, b). Jsou zrcadlově obrácené, a v závislosti na datasetu se protínají někde mezi 275 a 320 příznaky. Jelikož je celkový počet příznaků 547 (z čehož je polovina 273,5), můžeme říci, že algoritmus SFS je rychlejší. Algoritmus SFFS má závislost mírně exponenciální (viz Obrázek 41a, b), její přesný tvar ovšem závisí na konkrétním datasetu.

Stejně jako závislost na velikosti datasetu by i závislost na počtu příznaků měla být nejhůře lineární. Toto platí pro všechny algoritmy kromě SFFS. Protože ale selektujeme malý počet příznaků, není toto takový nedostatek, jako nevhodná závislost na velikosti datasetu.

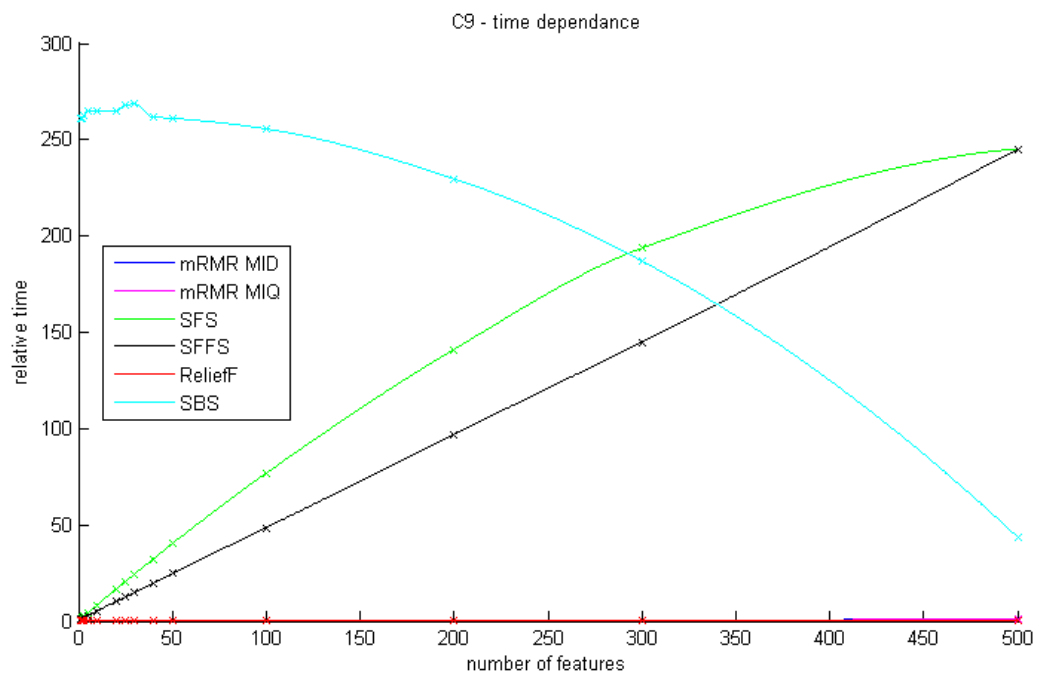


Obrázek 40: Závislost délky trvání na počtu selektovaných příznaků – dataset 3 (1192 segmentů)

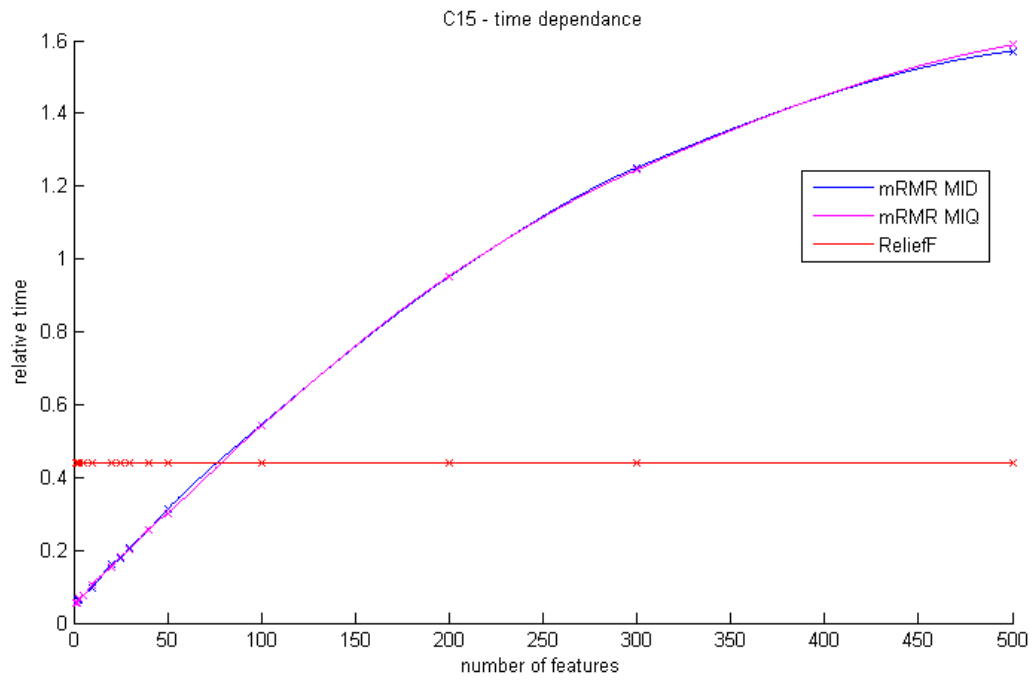
a)



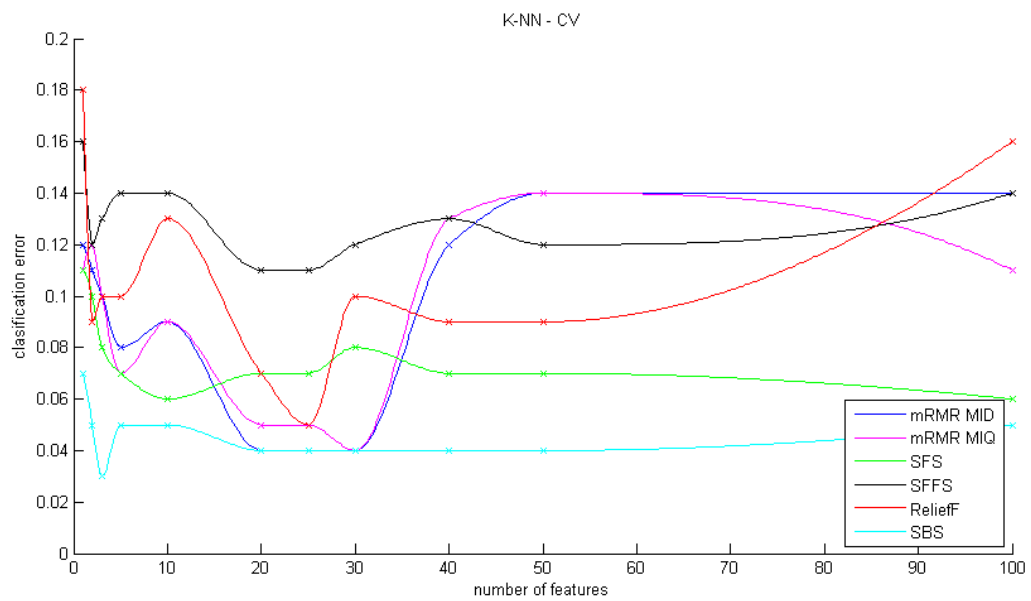
b)



Obrázek 41a, b: Závislost délky trvání na počtu selektovaných příznaků – tvar závislosti algoritmu SFSS; a) dataset 1 (100 segmentů), b) dataset 9 (16 segmentů)



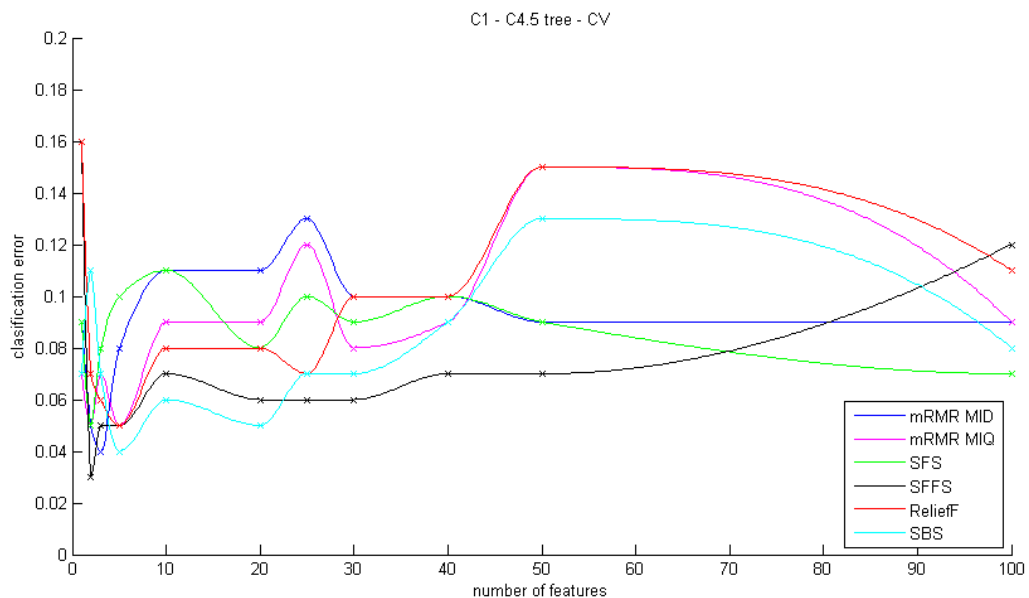
Obrázek 42: Závislost délky trvání na počtu selektovaných příznaků – mRMR a ReliefF; dataset 15



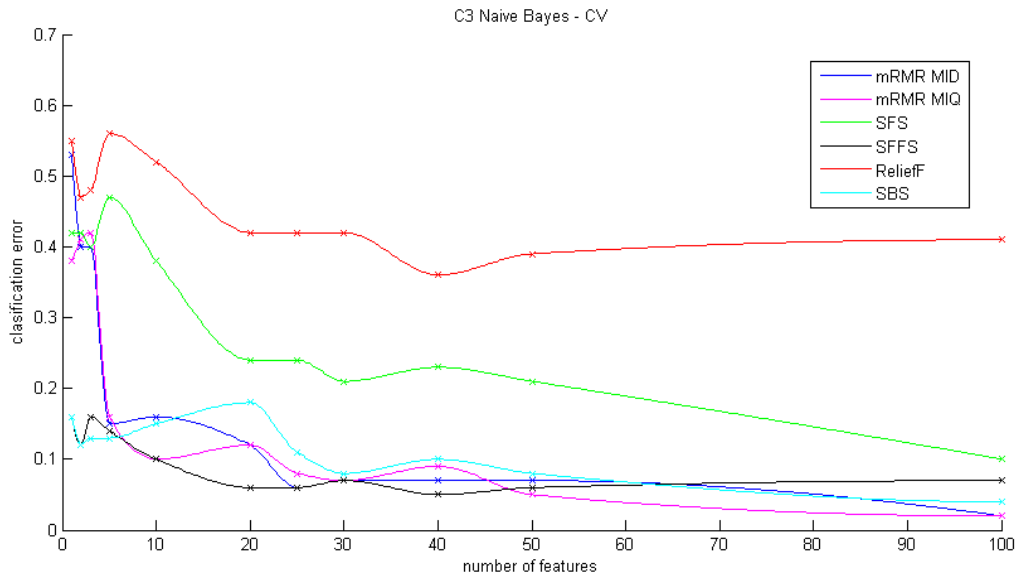
Obrázek 43: Závislost klasifikační chyby na počtu příznaků – hledání optimálního počtu; dataset 1; Cross-Validation k-NN

5.4.2 Určení počtu příznaků

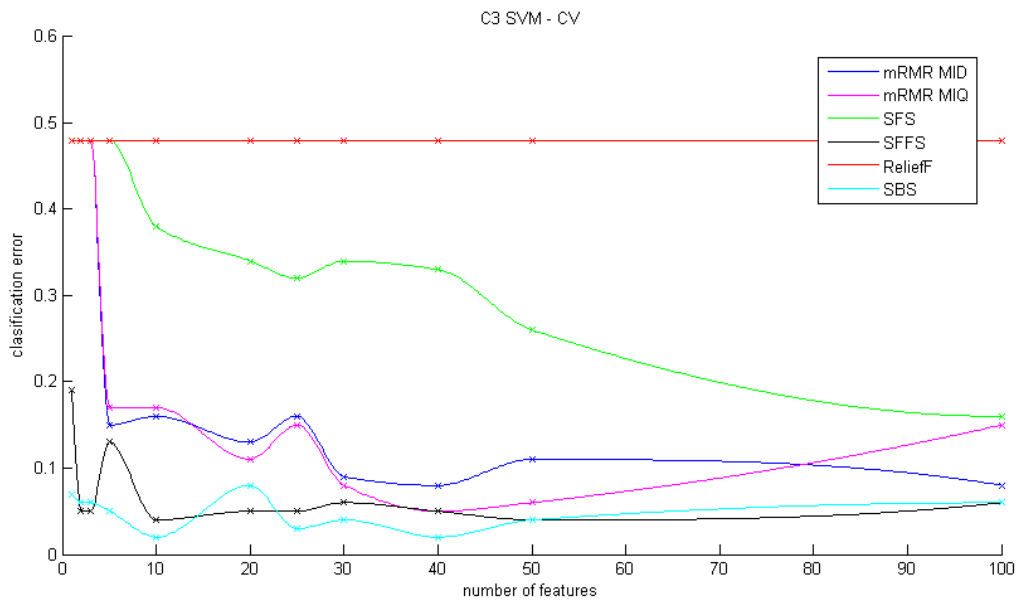
Při určování počtu příznaků jsme vycházeli z výkonnostních křivek vytvořených jednotlivými klasifikátory pro všechny metody selekce na 4 různých datasetech. Jak můžeme vidět z následujících křivek (viz Obrázek 43 až Obrázek 49), 20 příznaků je ve většině případů bod, kde výkonnostní křivka klesá, nebo bod, ve kterém ještě nestoupá. 20 příznaků se tedy většinou nachází v blízkosti lokálního minima nebo v blízkosti bodu zlomu křivky. Zároveň je tento počet dostatečně malý, abychom byli schopni výsledky jednotlivých selekcí mezi sebou porovnat (viz 5.4.3).



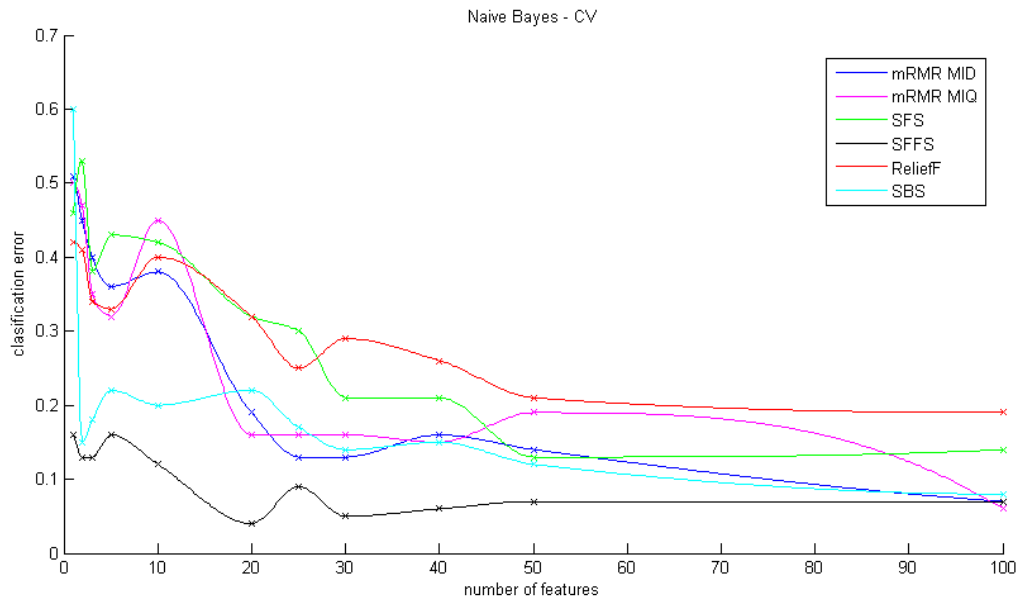
Obrázek 44: Závislost klasifikační chyby na počtu příznaků – hledání optimálního počtu; dataset 1; Cross-Validation C4.5



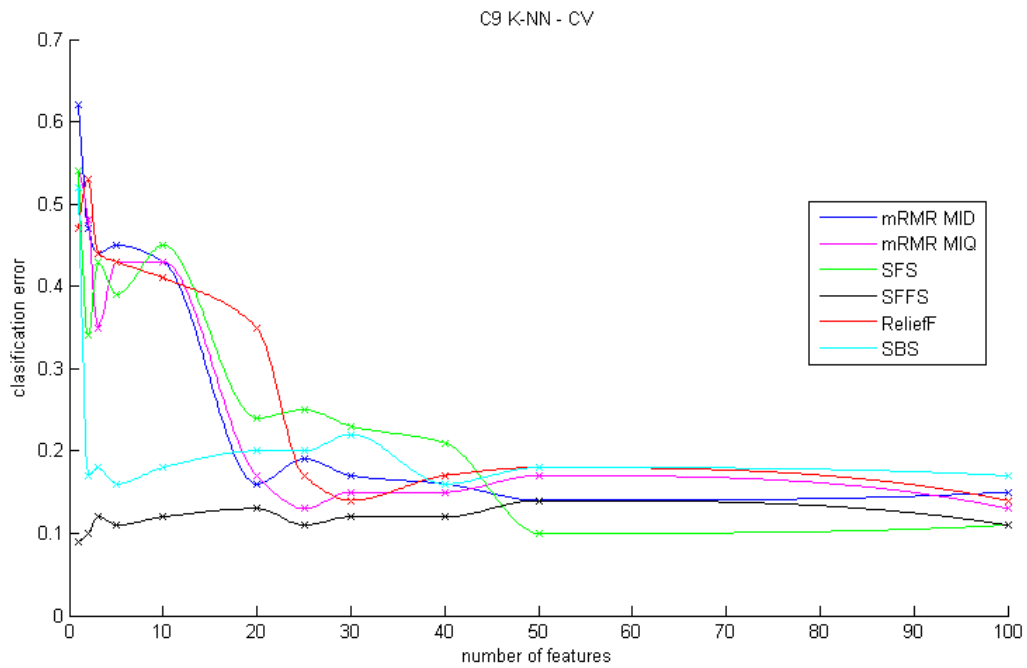
Obrázek 45: Závislost klasifikační chyby na počtu příznaků – hledání optimálního počtu; dataset 3; Cross-Validation Naive Bayes



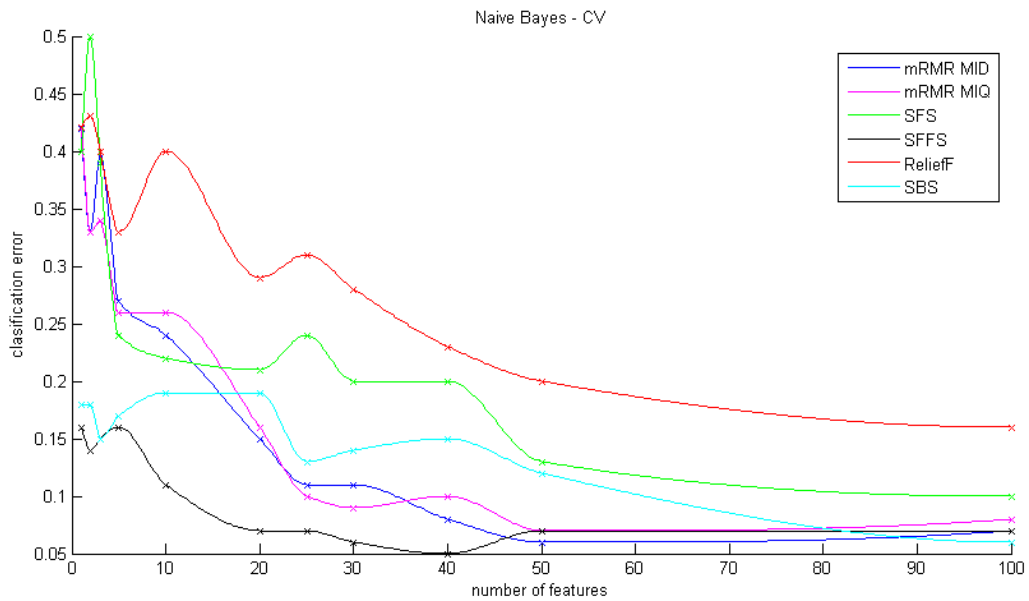
Obrázek 46: Závislost klasifikační chyby na počtu příznaků – hledání optimálního počtu; dataset 3; Cross-Validation SVM



Obrázek 47: Závislost klasifikační chyby na počtu příznaků – hledání optimálního počtu; dataset 9; Cross-Validation Naive Bayes



Obrázek 48: Závislost klasifikační chyby na počtu příznaků – hledání optimálního počtu; dataset 9; Cross-Validation k-NN



Obrázek 49: Závislost klasifikační chyby na počtu příznaků – hledání optimálního počtu; dataset 15; Cross-Validation Naive Bayes

5.4.3 Porovnání selektovaných příznaků

Porovnávali jsme 20 příznaků selektovaných jednotlivými algoritmy na 19 datasetech. Zjistili jsme, že v rámci příznaků selektovaných různými algoritmy na jednom datasetu se některé příznaky opakují. Neexistují ale příznaky, které by se vyskytovaly ve všech selekcích (viz Tabulka 7a Tabulka 8). Toto tvrzení platí i pro příznaky selektované jednou metodou pro různé datasety (viz Tabulka 9). Jedinou výjimkou je algoritmus SFSS, který selektoval téměř shodné příznaky pro všechny datasety (viz Tabulka 10). Pro názornost jsou příznaky v tabulkách (Tabulka 7 až Tabulka 10) uváděny pořadovými čísly, která jim byla přidělena (viz Příloha 1).

Všimněme si, že nejčastější příznaky jsou koeficienty vypočtené pomocí Waveletové transformace a koeficienty rychlé Fourierovy transformace pro jednotlivá pásma (beta, alfa, delta a theta). Koeficienty vzájemných korelací a koherence se mezi nejčastějšími příznaky téměř nevyskytují (viz Tabulka 11 a Tabulka 12).

Celkem jsme provedli 160 různých selekcí (liší se datasetem a algoritmem) a určili četnost jednotlivých příznaků. Téměř jedna třetina (32%) příznaků se nevyskytuje vůbec a více než jedna polovina (56%) se vyskytuje méně než třikrát – jedná se především o koeficienty vzájemných korelací a koherence, které pro tuto úlohu nejsou podstatné.

Příznaky selektované různými algoritmy pro jeden dataset

Tabulka 7: Příznaky (uvedené pořadovými čísly – viz Příloha 1) selektované pro dataset 1 různými metodami; barevně vyznačeno 5 nejčtetnějších příznaků

59	30	15	1	2	3	4	5	6	7	8	9	10	14	16	17	18	19	20	21
60	72	73	74	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547
75	74	72	60	73	71	59	11	57	58	12	15	25	30	45	13	29	24	56	44
15	44	97	7	24	69	145	11	96	8	42	12	99	4	98	493	79	45	495	59
59	520	19	91	35	247	22	370	145	24	508	15	133	461	524	74	7	495	85	9
59	7	145	24	15	22	74	9	44	85	97	12	42	29	8	96	524	98	69	60
44	56	24	29	13	45	30	25	15	12	58	57	11	59	71	73	60	72	74	75
30	41	42	43	44	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
16	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
7	20	26	43	44	45	46	47	48	49	51	52	53	54	55	56	57	58	59	60
22	32	46	31	2	5	23	28	18	54	48	1	12	47	8	20	41	56	30	45
49	5	19	41	28	55	38	43	52	7	34	25	10	21	12	22	1	44	48	16
28	6	27	58	56	9	41	4	14	45	5	12	33	40	1	50	20	26	7	35

Tabulka 8: Příznaky (uvedené pořadovými čísly – viz Příloha 1) selektované pro dataset 2 různými metodami; barevně vyznačeno 5 nejčtetnějších příznaků

401	115	107	473	463	527	483	531	471	517	523	519	521	61	437	525	479	461	485	501
25	35	40	60	68	71	72	83	85	92	95	103	368	369	370	482	492	512	528	546
75	60	74	72	71	11	73	59	15	30	25	70	12	45	103	40	13	96	67	35
400	401	9	375	7	97	405	402	10	404	399	367	403	532	406	348	374	376	392	365
9	510	493	405	375	401	518	490	323	359	270	292	392	498	330	429	542	433	402	373
9	510	493	405	401	518	375	501	392	323	292	449	542	402	7	330	406	498	270	531
2	3	10	12	16	17	21	25	30	34	35	37	42	44	47	48	50	51	55	60
1	2	9	10	12	13	14	19	22	24	25	27	32	35	41	42	47	52	53	58
1	2	4	7	9	10	11	22	24	29	32	34	36	37	45	47	49	50	55	58

Příznaky selektované jedním algoritmem pro různé datasey

Tabulka 9: Příznaky (uvedené pořadovými čísly – viz Příloha 1) selektované metodou SFS pro datasey 1-10; barevně vyznačeno 5 nejčtetnějších příznaků

59	30	15	1	2	3	4	5	6	7	8	9	10	14	16	17	18	19	20	21
401	115	107	473	463	527	483	531	471	517	523	519	521	61	437	525	479	461	485	501
507	516	518	374	441	517	364	528	439	61	539	524	522	20	531	519	505	521	485	503
501	372	507	543	491	509	483	61	497	493	527	447	525	533	20	521	545	435	439	481
371	484	516	545	539	459	497	505	429	449	483	531	519	529	61	433	521	523	503	485
514	531	376	507	348	391	518	495	502	547	471	107	511	529	517	62	524	501	100	519
9	289	445	487	174	126	20	61	63	127	62	64	429	100	107	171	65	203	111	216
449	61	63	108	19	20	62	64	120	100	109	18	112	105	65	79	110	111	89	113
153	107	61	62	63	152	20	64	147	108	65	115	119	131	137	110	117	118	121	122
484	10	371	496	492	494	513	521	61	529	519	531	517	525	527	528	535	541	499	447

Tabulka 10: Příznaky (uvedené pořadovými čísly – viz Příloha 1) selektované metodou SFFS pro datasey 1-10; barevně vyznačeny shodné příznaky

75	74	72	60	73	71	59	11	57	58	12	15	25	30	45	13	29	24	56	44
75	60	74	72	71	11	73	59	15	30	25	70	12	45	103	40	13	96	67	35
75	74	60	72	73	71	59	11	57	12	25	30	15	58	45	29	13	24	1	2
75	74	72	73	71	60	59	57	58	12	25	29	24	30	45	56	11	27	1	44
75	74	72	73	71	60	59	57	58	12	11	25	15	29	24	56	45	13	30	27
75	74	72	73	71	60	59	57	58	11	12	25	15	30	56	13	45	24	29	22
75	60	74	71	72	73	59	11	57	56	58	40	15	12	35	25	98	30	13	66
75	74	72	60	73	71	59	11	57	58	15	12	70	56	24	66	30	29	27	25
74	75	72	73	71	59	57	60	12	58	25	29	103	30	24	15	27	45	44	35
75	74	72	73	60	71	59	57	11	30	25	12	58	45	29	24	13	66	1	40

Tabulka 11: Nejčastější příznaky při výběru 20 příznaků na 19 datasetech všemi metodami (význam pojmenování příznaků viz Příloha 1)

Příznak	Četnost
ch_FP1---wav_band_rozsah5_db4	69/160
ch_FP1---wav_band_rozsah2_db4	64/160
ch_FP1---wav_band_rozsah3_db4	64/160
ch_FP1---wav_band_rozsah4_db4	63/160
ch_FP1---wav_band_rozsah1_db4	60/160
ch_FP1---std_wav_value_rozsah5_db4	46/160
ch_FP1---fft_abs_theta_(3_to_7Hz)	44/160
ch_FP1---fft_abs_beta2_(20_to_30Hz)	41/160
ch_FP1---skewness_wav_value_rozsah5_db4	41/160
ch_FP1---kurtosis_wav_value_rozsah1_db4	41/160
ch_FP1---kurtosis_wav_value_rozsah5_db4	40/160
ch_FP1---max_wav_value_rozsah5_db4	39/160
ch_FP1---2st_diff_mean	38/160
ch_FP1---std_wav_value_rozsah4_db4	38/160
ch_FP1---skewness_wav_value_rozsah2_db4	38/160
ch_FP1---kurtosis_wav_value_rozsah3_db4	37/160
ch_FP1---kurtosis_wav_value_rozsah2_db4	36/160
ch_FP1---skewness	34/160
ch_FP1---fft_abs_delta_(0.1_to_3Hz)	34/160
ch_FP1---max_wav_value_rozsah4_db4	34/160
ch_FP1---min_value	33/160
ch_FP1---min_wav_value_rozsah5_db4	33/160
ch_FP1---skewness_wav_value_rozsah3_db4	32/160
ch_FP1---std_wav_value_rozsah2_db4	31/160
ch_FP1---kurtosis_wav_value_rozsah4_db4	31/160
ch_FP1---skewness_wav_value_rozsah1_db4	30/160
ch_FP1---skewness_wav_value_rozsah4_db4	30/160
ch_FP1---min_wav_value_rozsah4_db4	27/160
ch_FP1---std_wav_value_rozsah3_db4	27/160
ch_FP1---1st_diff_mean	26/160
ch_FP1---entropy_wav_rozsah4_db4	26/160
ch_C3---ch_C4---coher_5-6hz	25/160
ch_FP1---fft_abs_beta1_(12_to_20Hz)	24/160

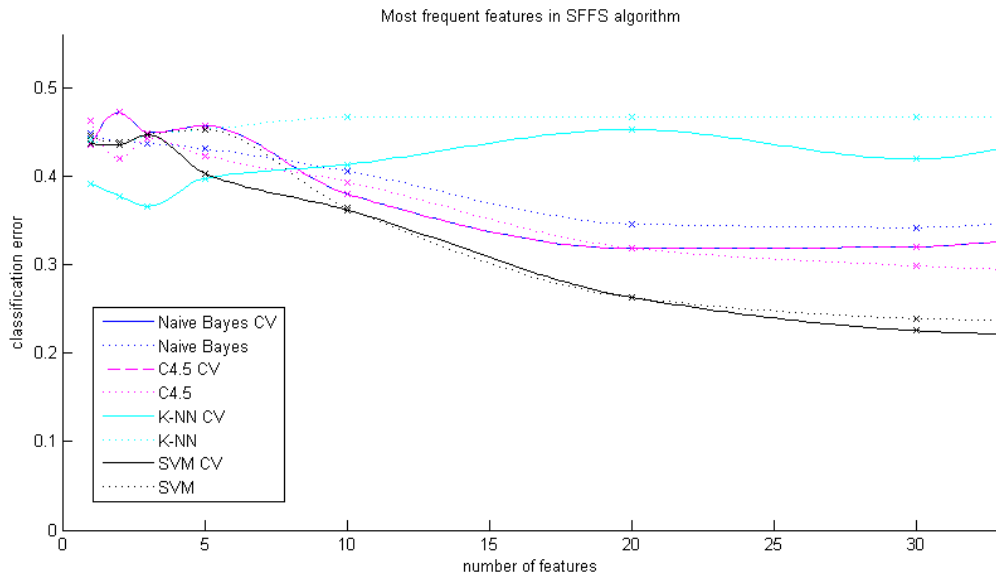
5.4.4 Ověření na nezávislém datasetu

V souvislosti se zjištěními z kap. 5.4.3 jsme vybrali k ověření na nezávislém datasetu jednak všechny příznaky vybírané metodou SFFS, tak i stejný počet příznaků, které byly nejčastěji vybírány všemi metodami (viz Tabulka 11 a Tabulka 12). Vypočetli a vykreslili jsme výkonnostní křivky a jako optimální počet nám opět vyšlo 20 příznaků (viz Obrázek 50 a Obrázek 51). Těchto 20 příznaků jsme použili k určení klasifikační chyby na všech datasetech. Zároveň jsme zjistili, že příznaky vybrané SFFS není vhodné použít obecně, protože klasifikační chyba je příliš vysoká. Způsob, jakým algoritmus vybírá příznaky,

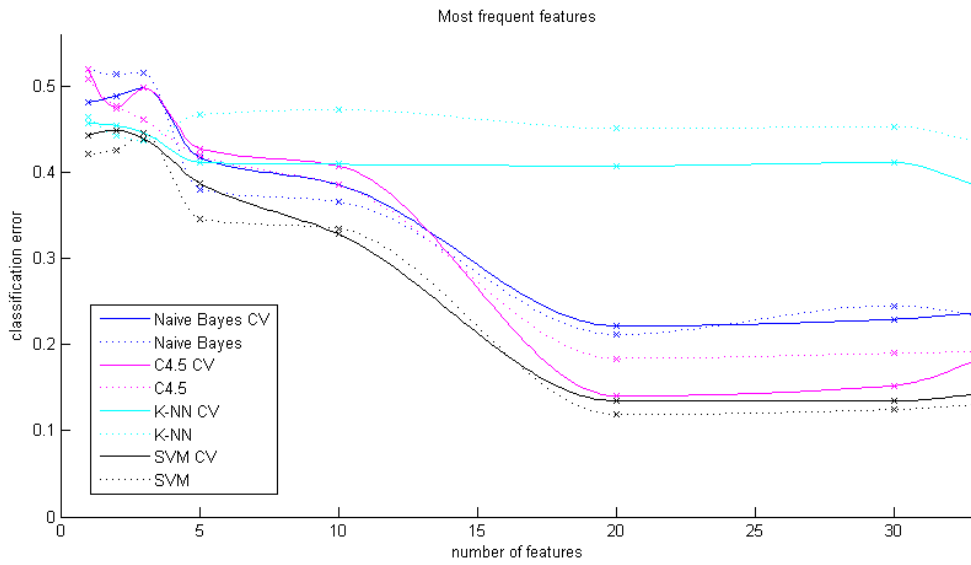
způsobuje, že jsou optimální pouze v daném kontextu, tedy pouze pokud zvolíme všechny, které byly společně vybrány.

Tabulka 12: Příznaky selektované metodou SFFS (význam pojmenování příznaků viz Příloha 1)

Příznak	Četnost
ch_FP1---entropy_wav_rozsah5_db4	19/19
ch_FP1---entropy_wav_rozsah4_db4	19/19
ch_FP1---entropy_wav_rozsah2_db4	19/19
ch_FP1---entropy_wav_rozsah3_db4	19/19
ch_FP1---wav_band_rozsah5_db4	19/19
ch_FP1---entropy_wav_rozsah1_db4	19/19
ch_FP1---wav_band_rozsah4_db4	19/19
ch_FP1---mean_wav_value_rozsah1_db4	19/19
ch_FP1---min_wav_value_rozsah5_db4	19/19
ch_FP1---fft_abs_theta_(3_to_7Hz)	19/19
ch_FP1---fft_abs_delta_(0.1_to_3Hz)	18/19
ch_FP1---wav_band_rozsah2_db4	18/19
ch_FP1---wav_band_rozsah3_db4	18/19
ch_FP1---fft_abs_beta2_(20_to_30Hz)	16/19
ch_FP1---std_wav_value_rozsah5_db4	16/19
ch_FP1---max_wav_value_rozsah4_db4	15/19
ch_FP1---wav_band_rozsah1_db4	15/19
ch_FP1---fft_abs_alpha_(7_to_12Hz)	14/19
ch_FP1---min_wav_value_rozsah4_db4	13/19
ch_FP1---nonlin_energy	8/19
ch_FP1---min_value	6/19
ch_FP1---fft_abs_beta1_(12_to_20Hz)	4/19
ch_FP1---max_wav_value_rozsah2_db4	4/19
ch_FP1---mean_wav_value_rozsah5_db4	4/19
ch_FP1---std_wav_value_rozsah4_db4	4/19
ch_FP1---entropy_log_wav_rozsah1_db4	4/19
ch_FP1---median_wav_value_rozsah5_db4	3/19
ch_FP1---energy_percent_wav_rozsah2_db4	3/19
ch_FP1---max_value	2/19
ch_FP1---entropy_log_wav_rozsah5_db4	2/19
ch_FP1---min_wav_value_rozsah2_db4	1/19
ch_FP1---entropy_log_wav_rozsah2_db4	1/19
ch_FP1---rms	1/19



Obrázek 50: Klasifikační chyba na nezávislém datasetu při použití příznaků vybraných algoritmem SFFS



Obrázek 51: Klasifikační chyba na nezávislém datasetu při použití nejčastějších příznaků

V následující tabulce jsou shrnuty výsledky validace námi vybraného setu příznaků na všech 20 datasetech. Pouze u 6 datasetů je průměrná klasifikační chyba menší než 25% a pouze u 2 nižší než 15%. Ačkoli jsme zvolili příznaky, které byly vybírány nejčastěji, nedosahujeme požadovaných výsledků. Klasifikátory jsou příliš citlivé na daný dataset.

Tabulka 13: Klasifikační chyba (CV) na 20 datasetech při použití 20 nejčastějších příznaků; červeně označeny hodnoty větší než 25%

dataset	počet segmentů	Naive Bayes	C4.5	k-NN	SVM	Min	Ø
1	100	6%	4%	11%	8%	4%	7%
2	562	40%	43%	45%	44%	40%	43%
3	1192	21%	24%	24%	19%	19%	22%
4	1015	35%	38%	38%	31%	31%	36%
5	1010	36%	35%	44%	35%	35%	38%
6	1355	41%	41%	41%	36%	36%	40%
7	76	36%	26%	50%	39%	26%	38%
8	19	37%	68%	47%	53%	37%	51%
9	16	25%	19%	31%	31%	19%	27%
10	2290	39%	31%	46%	23%	23%	35%
11	874	32%	35%	44%	31%	31%	35%
12	634	9%	11%	12%	7%	7%	10%
13	747	18%	20%	20%	14%	14%	18%
14	437	19%	22%	24%	18%	18%	21%
15	480	43%	43%	47%	48%	43%	45%
16	333	29%	28%	29%	25%	25%	28%
17	1125	32%	35%	37%	28%	28%	33%
18	1444	27%	31%	34%	23%	23%	29%
19	1590	26%	34%	28%	25%	25%	28%
20	506	22%	17%	42%	13%	13%	23%
	Min	6%	4%	11%	7%		
					Ø	25%	30%

6 Závěr

Cílem práce bylo nalézt soubor příznaků segmentů signálu EEG, který je vhodný pro klasifikaci spánku donošených novorozenců, a zároveň porovnat jednotlivé algoritmy selekce v kombinaci s různými klasifikátory. Experiment se skládal ze dvou hlavních částí – z výpočtu výkonnostních křivek jednotlivých algoritmů a z analýzy četnosti výběru jednotlivých příznaků. Z důvodu přílišné paměťové náročnosti algoritmu SBBS jsme nakonec porovnávali pouze 6 algoritmů – mRMR MID a MIQ, ReliefF, SFFS (typ Filter) a SBS a SFS (typ Wrapper) v kombinaci se 4 klasifikátory (k-NN, SVM, C4.5 a Naive Bayes). Výsledky klasifikací jsme vždy porovnávali s hodnocením odborníka.

Vypočetli a vykreslili jsme dva druhy výkonnostních křivek – závislost klasifikační chyby a závislost doby výpočtu na počtu příznaků a segmentů.

Klasifikační chybu jednotlivých klasifikátorů jsme určovali pomocí Hold-Out validace i pomocí Cross-Validation. Ověřili jsme, že výsledky určené Hold-Out validací jsou zkreslené a většinou příliš pesimistické, proto jsme uvažovali pouze chyby určené pomocí Cross-Validation. Nejlepších výsledků dosahuje klasifikátor Naive Bayes, většinou s klasifikační chybou do 10%. Naopak nejvyšší klasifikační chyby, při nevhodném výběru příznaků, dosahuje klasifikátor SVM. Volba rozdělení dat na testovací a trénovací nejvíce ovlivňuje klasifikátor k-NN. Klasifikátor C4.5 je velmi výpočetně náročný, proto jsme provedli klasifikace pouze na jednom datasetu. Nejlepší kombinace klasifikátor – algoritmus selekce vyšla kombinace Naive Bayes a SFFS.

Algoritmy typu Wrapper jsou výrazně časově náročnější než algoritmy typu Filter, ačkoli nedosahují lepších výsledků. SFS dosahuje časů pod 1,5h, zatímco SBS počítá řádově hodiny. Oba algoritmy mají logaritmickou závislost na počtu příznaků a exponenciální závislost na počtu segmentů. Nejsou tedy příliš vhodné pro naši úlohu, která má obvykle velké datasety. Nejrychlejším algoritmem je mRMR s dobou výpočtu pod 2s, varianty MID a MIQ se příliš neliší. Algoritmus SFFS je jen o něco pomalejší, doba výpočtu je vždy menší než 1 minuta a je téměř nezávislá na počtu segmentů. Algoritmus ReliefF má pro datasety s více než 20 segmenty exponenciální závislost na počtu segmentů a není proto vhodný.

Z výkonnostních křivek jsme určili optimální počet 20 příznaků. Následně jsme porovnávali příznaky selektované jednotlivými algoritmy. Zjistili jsme, že příznaky selektované jednotlivými algoritmy na jednom datasetu se opakují, ale neshodují. Stejně to

platí i pro příznaky selektované jedním algoritmem pro různé datasey. Výjimkou je algoritmus SFFS, který selektoval vždy téměř shodné příznaky.

Celkem jsme provedli 160 různých selekcí 20 příznaků, abychom určili četnost jejich výběru. Nejčastějšími příznaky byly koeficienty vypočtené pomocí Waveletové transformace. Koeficienty vzájemných korelací a koherence jsou nejméně četné a můžeme je tedy v budoucnu vynechat. Téměř třetina z celkového počtu příznaků nebyla ani jednou vybrána.

20 nejčastějších příznaků celkem a 20 nejčastějších příznaků vybrané metodou SFFS jsme jako sadu ověřili na nezávislém datasetu. Zjistili jsme, že takto zvolené příznaky jsou vhodné k použití pouze jako sada, se všemi příznaky, se kterými byly společně nalezeny. Velmi markantní to je u příznaků vybraných metodou SFFS, což je způsobeno tím, že algoritmus hledá nejlepší řešení v daném kontextu. Nejlepší metoda pro klasifikaci novorozeneckého spánku, kterou jsme v práci našli, je kombinace algoritmu SFFS a klasifikátoru Naive Bayes, která je rychlá i přesná zároveň.

Součástí práce bylo vytvoření systému v MatLabu, který sjednocuje již naimplementované algoritmy selekce a klasifikátory a umožňuje provádět selekci příznaků z dat ve formátu arff. Systém lze snadno spustit pomocí jednoho kmenového souboru, který umožňuje různá nastavení. Lze vykreslit a vypočítat výkonnostní křivky, selektovat zvolený počet příznaků některým z algoritmů nebo klasifikovat zvolená data vybraným klasifikátorem. Toto nám umožňuje snadno v práci pokračovat.

V budoucnosti bychom chtěli na tuto práci navázat nejen rozšířením o další algoritmy selekce. Úpravy by se týkaly také zpracování signálu EEG a extrakce příznaků. Uvažujeme o využití adaptivní segmentace a o selekci odlišných příznaků. Protože se mezi nejčastěji selektovanými příznaky objevovaly příznaky získané Waveletovou transformací, právě tato transformace bude předmětem dalšího zkoumání. Další možností je vypočítat příznaky, které budou sledovat poměr jednotlivých rytmů v delších časových úsecích. Konečně bychom chtěli vyzkoušet i odlišné klasifikátory nebo jejich kombinace pomocí učících algoritmů (například Adaboost), vzhledem k odlišnostem, které jsme zaznamenali mezi použitými klasifikátory.

7 Literatura

- [1] **Kantůrek, M.** *BP: Adaptivní segmentace EEG záznamu*. Praha : ČVUT, 2009.
- [2] **Estevéz, P. A. a další.** Polysomnographic pattern recognition for automated classification of sleep–waking states in infants. *Med. Biol. Eng. Comput.* 2002, Sv. 40.
- [3] **Kostílek, M.** *DP: Vliv emočních podnětů na charakteristiku EEG signálu*. Praha : ČVUT, 2009.
- [4] **Koukolík, F.** *Mozek a jeho duše*. Praha : Makropulos, 1995.
- [5] **Orel, M., Facová, V. a kol.** *Člověk, jeho mozek a svět*. Praha : Grada, 2009.
- [6] Neuron Basics. [Online] [Citace: 1. 5 2011.]
<http://www.mindcreators.com/NeuronBasics.htm>.
- [7] **Faber, J.** *Elektroencefalografie a psychofyziologie*. Praha : ISV nakladatelství, 2001.
- [8] **Gerla, V.** *Classification Concept in EEG Analysis (preparation for doctoral thesis)*. Praha : ČVUT, 2010.
- [9] **Svoboda, P.** *DP: Metody analýzy EEG aktivity*. Praha : ČVUT, 2011.
- [10] Spirit Alchemy: Articles: Sleep and Dreams. [Online] [Citace: 2. 5 2011.]
<http://www.spiritalchemy.com/articles/SleepAndDreams.html>.
- [11] **Rieger, J., Lhotská, L. a Krajča, V.** Zpracování dlouhodobých EEG záznamů. *Advances in electrical and electronic engineering*. 2005, Sv. 4, 3.
- [12] **Hřebeňár, J.** *Analýza novorozeneckých polysomnografických záznamů*. Praha : ČVUT, 2008.
- [13] EEG Artefakty a jejich rozdělení. [Online] [Citace: 3. 5 2011.] <http://zivotni-energie.cz/eeg-artefakty-a-jejich-rozdeleni.html>.
- [14] **Huy, N. Q.** *Metody pro zpracování EEG signálu*. Praha : ČVUT, 2007.
- [15] Polysomnography. [Online] [Citace: 3. 5 2011.]
http://www.lookfordiagnosis.com/mesh_info.php?term=Polysomnography&lang=1.
- [16] Elektroenzephalographie - TFODE. [Online] [Citace: 3. 5 2011.]
<http://de.enc.tfode.com/Elektroenzephalografie>.
- [17] **Řehoř, M.** *BP: Metody klasifikace EEG signálu*. Praha : ČVUT, 2007.
- [18] Spánek, spánkové fáze a hypnogram. [Online] [Citace: 4. 5 2011.] <http://zivotni-energie.cz/spanek-spankove-faze-a-hypnogram.html>.
- [19] **Novák, R.** *DP: Hodnocení EEG v reálném čase*. Praha : ČVUT, 2011.

- [20] **Gerla, V. a další.** Multivariate Analysis od Full-Term Neonatal Polysomnographic Data. *TITB*. 2007, 3.
- [21] Novorozenecké EEG, spánkové fázenovorozenců. [Online] <http://zivotni-energie.cz/novorozenecke-eeeg-spankove-faze-novorozencu.html>.
- [22] **Gerla, V.** PSGLab. [Online] [Citace: 12. 5 2011.] <http://bio.felk.cvut.cz/psglab/>.
- [23] **Agarwal, R., Gotman, J. a Flanagan, D.** Automatic EEG analysis during long-term monitoring in the ICU. *ELSEVIER. Electroencephalography and clinical Neurophysiology*, 1998, 107.
- [24] **Krajča, V. a další.** Neonatal EEG Sleep Stages Modelling by Temporal Profiles. *EUROCAST. Computer Aided Systems and Theory*, 2007.
- [25] **Saeys, Y., Inza, I. a Larrañaga, P.** A review of feature selection techniques in bioinformatics. *Bioinformatics*. 2007, Sv. 23, 19.
- [26] **Konečný, J.** k-nearest neighbor. [Online] [Citace: 14. 5 2011.] http://phoenix.inf.upol.cz/~konecnyj/data/zzd2_5.pdf.
- [27] **Hlaváč, V.** Učení bez učitele. [Online] [Citace: 14. 5 2011.] <http://cmp.felk.cvut.cz/~hlavac/Public/TeachingLectures/UceniBezUcitele.pdf>.
- [28] **BioLab.** Neuronové sítě. [Online] [Citace: 14. 5 2011.] http://cyber.felk.cvut.cz/gerstner/biolab/bio_web/teach/FunBio/neuron/neursite.html.
- [29] **Vomlelová, M.** Přednáška Umělé inteligence II. [Online] [Citace: 14. 5 2011.] <http://kti.mff.cuni.cz/~marta/slistromy.pdf>.
- [30] **Witten, I. H. a Frank, E.** *Data Mining: Practical Machine Learning Tools and Techniques*. San Francisco : Elsevier, 2005.
- [31] **Žižka, J.** WEKA -Data Mining. [Online] [Citace: 15. 5 2011.] https://akela.mendelu.cz/~zizka/Machine_Learning/Presentation/WEKA_Data-mining.pdf.
- [32] **Refaeilzadeh, P., Tang, L. a Liu, H.** Cross-Validation. *Encyclopedia of Database Systems (EDBS)*. 2009.
- [33] **Macaš, M.** Complete Cross-Validation Error of 1-Nearest Neighbour Classifier as Feature Selection Criterion. [*nepublikováno*].
- [34] **Somol, P., Novovičová, J. a Pudil, P.** Efficient Feature Subset Selection an Subset Size Optimization. *Survey. Pattern Recognition Recent Advances*, 2010.
- [35] **Holubec, B.** *DP:Použití PSO metody pro selekci příznaků*. Praha : ČVUT, 2008.

- [36] **Liu, H. a Motoda, H.** *Computational Methods of Feature Selection*. Boca Raton, FL : Taylor & Francis Group, LCC, 2008.
- [37] **Guyon, I. a Elisseeff, A.** An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research* 3. 2003.
- [38] **Webb, A. R.** *Statistical Pattern Recogniton, Secon Edition*. Malvern, UK : QinetiQ Ltd., 2002.
- [39] **Molina, L.C., Belanche, L. a Nebot, Á.** Feature Selection Algorithms: A Survey and Experimental Evaluation. *IEEE International Conference*. 2002.
- [40] **Refaeilzadeh, P., Tang, L. a Liu, H.** On Comparison of Feature Selection Algorithms. *Association for the Advancement of Artificial Intelligence (AAAI) Workshop*. 2007.
- [41] Approaches to dimensionality reduction. [Online] [Citace: 17. 5 2011.]
<http://bib.oxfordjournals.org/content/9/2/102.full>.
- [42] **Jain, A. and Zongker, D.** Feature Selection: Evaluation, Application, and Small Sample Performance. *IEEE. Transactions on Pattern Analysis and Machine Intelligence*, 1997, Vol. 19.
- [43] **Ding, Ch. a Peng, H.** Minimum Redundancy Feature Selection from Microarray Gene Expression Data. *Journal of Bioinformatics an Computational Biology*. 2004, Sv. 3, 2.
- [44] **Peng, H., Long, F. a Ding, Ch.** Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance and Min-Redundancy. *IEEE. Transactions on Pattern Analysis and Machine Learning*, 2005, Sv. 27, 8.
- [45] **Robnik-Šikonja, M. a Kononenko, I.** Theoretical and Empirical Analysis of ReliefF and RReliefF. *Machine Learning Journal*. 2003, 53.
- [46] **Doumpos, M. a Salappa, Á.** Feature Selection Algoritihms in Classification Problems: An Experimental Evaluation. *AIKED'05 Proceedings of the 4th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering Data Bases*. 2005.
- [47] MatLab WEKA interface - File Exchange. [Online] [Citace: 1. 3 2011.]
<http://www.mathworks.com/matlabcentral/fileexchange/21204-matlab-weka-interface>.
- [48] Matlab Primer. [Online] [Citace: 15. 4 2011.]
<http://artax.karlin.mff.cuni.cz/~beda/cz/matlab/primercz/matlab-primer.html#intro>.
- [49] Matlab, laboratoř nejen pro matematiky. [Online] [Citace: 22. 4 2011.]
http://cmp.felk.cvut.cz/~pisa/Public/ST_matlab.html.

- [50] **Zhao, Z., Morstatter, F. a další.** Advancing Feature Selection Research - ASU Feature Selection Repository. <http://featureselection.asu.edu/>. 2010.
- [51] Feature Selection ASU. [Online] [Citace: 1. 4 2011.] <http://featureselection.asu.edu/>.
- [52] mRMR Feature Selection Site. [Online] [Citace: 1. 4 2011.] <http://penglab.janelia.org/proj/mRMR/#matlab>.
- [53] **Duin, R.P.W., Juszczak, P. a Paclik, P.** *PRTools4.1, A Matlab Toolbox for Pattern Recognition*. Hamilton : Delft University of Technology, 2007.
- [54] PRTools 4.1. [Online] [Citace: 15. 4 2011.] <http://prtools.org/files/PRTools4.1.pdf>.
- [55] WEKA 3 - Data Mining with Open Source Machine Learning Software. [Online] [Citace: 1. 5 2011.] <http://www.cs.waikato.ac.nz/~ml/weka/index.html>.
- [56] **Souhrada, V.** *BP:Weka a ERP*. Plzeň : ČVUT, 2008.
- [57] Attribute-Relation File Format. [Online] [Citace: 20. 4 2011.] <http://www.cs.waikato.ac.nz/~ml/weka/arff.html>.
- [58] **Levy, T., Loparo, K. A. a Scher, M.** Automated Scoring of Neonatal Sleep. *Submitted to IEEE*.

Přílohy

Příloha 1: Seznam příznaků

1. ch_FP1---min_value¹
2. ch_FP1---max_value²
3. ch_FP1---mean³
4. ch_FP1---std⁴
5. ch_FP1---skewness⁵
6. ch_FP1---kurtosis⁶
7. ch_FP1---1st_diff_mean⁷
8. ch_FP1---1st_diff_max
9. ch_FP1---2st_diff_mean
10. ch_FP1---2st_diff_max
11. ch_FP1---fft_abs_delta_(0.1_to_3Hz)⁸
12. ch_FP1---fft_abs_theta_(3_to_7Hz)
13. ch_FP1---fft_abs_alpha_(7_to_12Hz)
14. ch_FP1---fft_abs_beta1_(12_to_20Hz)
15. ch_FP1---fft_abs_beta2_(20_to_30Hz)
16. ch_FP1---fft_rel_delta_(0.1_to_3Hz)⁹
17. ch_FP1---fft_rel_theta_(3_to_7Hz)
18. ch_FP1---fft_rel_alpha_(7_to_12Hz)
19. ch_FP1---fft_rel_beta1_(12_to_20Hz)
20. ch_FP1---fft_rel_beta2_(20_to_30Hz)
21. ch_FP1---min_wav_value_rozsah1_db4¹⁰
22. ch_FP1---min_wav_value_rozsah2_db4¹¹
23. ch_FP1---min_wav_value_rozsah3_db4¹²
24. ch_FP1---min_wav_value_rozsah4_db4¹³
25. ch_FP1---min_wav_value_rozsah5_db4¹⁴
26. ch_FP1---max_wav_value_rozsah1_db4
27. ch_FP1---max_wav_value_rozsah2_db4
28. ch_FP1---max_wav_value_rozsah3_db4
29. ch_FP1---max_wav_value_rozsah4_db4
30. ch_FP1---max_wav_value_rozsah5_db4
31. ch_FP1---mean_wav_value_rozsah1_db4
32. ch_FP1---mean_wav_value_rozsah2_db4
33. ch_FP1---mean_wav_value_rozsah3_db4
34. ch_FP1---mean_wav_value_rozsah4_db4
35. ch_FP1---mean_wav_value_rozsah5_db4
36. ch_FP1---median_wav_value_rozsah1_db4¹⁵
37. ch_FP1---median_wav_value_rozsah2_db4
38. ch_FP1---median_wav_value_rozsah3_db4
39. ch_FP1---median_wav_value_rozsah4_db4
40. ch_FP1---median_wav_value_rozsah5_db4
41. ch_FP1---std_wav_value_rozsah1_db4
42. ch_FP1---std_wav_value_rozsah2_db4
43. ch_FP1---std_wav_value_rozsah3_db4
44. ch_FP1---std_wav_value_rozsah4_db4
45. ch_FP1---std_wav_value_rozsah5_db4
46. ch_FP1---skewness_wav_value_rozsah1_db4
47. ch_FP1---skewness_wav_value_rozsah2_db4
48. ch_FP1---skewness_wav_value_rozsah3_db4
49. ch_FP1---skewness_wav_value_rozsah4_db4
50. ch_FP1---skewness_wav_value_rozsah5_db4
51. ch_FP1---kurtosis_wav_value_rozsah1_db4
52. ch_FP1---kurtosis_wav_value_rozsah2_db4
53. ch_FP1---kurtosis_wav_value_rozsah3_db4
54. ch_FP1---kurtosis_wav_value_rozsah4_db4
55. ch_FP1---kurtosis_wav_value_rozsah5_db4
56. ch_FP1---wav_band_rozsah1_db4¹⁶
57. ch_FP1---wav_band_rozsah2_db4
58. ch_FP1---wav_band_rozsah3_db4
59. ch_FP1---wav_band_rozsah4_db4
60. ch_FP1---wav_band_rozsah5_db4
61. ch_FP1---wav_band_relative_rozsah1_db4
62. ch_FP1---wav_band_relative_rozsah2_db4
63. ch_FP1---wav_band_relative_rozsah3_db4
64. ch_FP1---wav_band_relative_rozsah4_db4
65. ch_FP1---wav_band_relative_rozsah5_db4
66. ch_FP1---entropy_log_wav_rozsah1_db4¹⁷
67. ch_FP1---entropy_log_wav_rozsah2_db4
68. ch_FP1---entropy_log_wav_rozsah3_db4
69. ch_FP1---entropy_log_wav_rozsah4_db4
70. ch_FP1---entropy_log_wav_rozsah5_db4
71. ch_FP1---entropy_wav_rozsah1_db4
72. ch_FP1---entropy_wav_rozsah2_db4
73. ch_FP1---entropy_wav_rozsah3_db4
74. ch_FP1---entropy_wav_rozsah4_db4
75. ch_FP1---entropy_wav_rozsah5_db4
76. ch_FP1---1st_diff_wav_mean_rozsah1_db4
77. ch_FP1---1st_diff_wav_mean_rozsah2_db4

¹ min - minimum

² max - maximum

³ mean - průměr

⁴ std - směrodatná odchylka

⁵ skewness - šikmost

⁶ kurtosis - špičatost

⁷ diff - diference

⁸ fft – Fast Fourier Transform

abs – absolutní hodnota

⁹ rel – poměrná hodnota

¹⁰ wav – Waveletová transformace

db4 – označení druhu wav (MatLab)

rozsah1 - 16-32

¹¹ rozsah2 - 12-16

¹² rozsah3 – 8-12

¹³ rozsah4 – 4-8

¹⁴ rozsah5 – 0-4

¹⁵ median - medián

¹⁶ band - pásmo

¹⁷ entropy – entropie

log - logaritmus

78. ch_FP1---1st_diff_wav_mean_rozsah3_db4
79. ch_FP1---1st_diff_wav_mean_rozsah4_db4
80. ch_FP1---1st_diff_wav_mean_rozsah5_db4
81. ch_FP1---1st_diff_wav_max_rozsah1_db4
82. ch_FP1---1st_diff_wav_max_rozsah2_db4
83. ch_FP1---1st_diff_wav_max_rozsah3_db4
84. ch_FP1---1st_diff_wav_max_rozsah4_db4
85. ch_FP1---1st_diff_wav_max_rozsah5_db4
86. ch_FP1---2nd_diff_wav_mean_rozsah1_db4
87. ch_FP1---2nd_diff_wav_mean_rozsah2_db4
88. ch_FP1---2nd_diff_wav_mean_rozsah3_db4
89. ch_FP1---2nd_diff_wav_mean_rozsah4_db4
90. ch_FP1---2nd_diff_wav_mean_rozsah5_db4
91. ch_FP1---2nd_diff_wav_max_rozsah1_db4
92. ch_FP1---2nd_diff_wav_max_rozsah2_db4
93. ch_FP1---2nd_diff_wav_max_rozsah3_db4
94. ch_FP1---2nd_diff_wav_max_rozsah4_db4
95. ch_FP1---2nd_diff_wav_max_rozsah5_db4
96. ch_FP1---rms¹⁸
97. ch_FP1---line_length¹⁹
98. ch_FP1---nonlin_energy
99. ch_FP1---entropy_log
100. ch_FP1---std(diff1)_to_std (hjorth2)
101. ch_FP1---std(diff2)_to_std (hjorth3)
102. ch_FP1---energy_percent_wav_rozsah1_db4
103. ch_FP1---energy_percent_wav_rozsah2_db4
104. ch_FP1---energy_percent_wav_rozsah3_db4
105. ch_FP1---energy_percent_wav_rozsah4_db4
106. ch_FP1---energy_percent_wav_rozsah5_db4
107. ch_FP1---zero_crossing²⁰
108. ch_FP1---ch_C3---coher_1-2hz²¹
109. ch_FP1---ch_C3---coher_2-3hz
110. ch_FP1---ch_C3---coher_3-4hz
111. ch_FP1---ch_C3---coher_4-5hz
112. ch_FP1---ch_C3---coher_5-6hz
113. ch_FP1---ch_C3---coher_6-7hz
114. ch_FP1---ch_C3---coher_7-8hz
115. ch_FP1---ch_C3---coher_8-9hz
116. ch_FP1---ch_C3---coher_9-10hz
117. ch_FP1---ch_C3---coher_10-11hz
118. ch_FP1---ch_C3---coher_11-12hz
119. ch_FP1---ch_C3---coher_12-13hz
120. ch_FP1---ch_C3---coher_13-14hz
121. ch_FP1---ch_C3---coher_14-15hz
122. ch_FP1---ch_C3---coher_15-16hz
123. ch_FP1---ch_C3---coher_16-17hz
124. ch_FP1---ch_C3---coher_17-18hz
125. ch_FP1---ch_C3---coher_18-19hz
126. ch_FP1---ch_C3---coher_19-20hz
127. ch_FP1---ch_C3---coher_20-21hz
128. ch_FP1---ch_T3---coher_1-2hz
129. ch_FP1---ch_T3---coher_2-3hz
130. ch_FP1---ch_T3---coher_3-4hz
131. ch_FP1---ch_T3---coher_4-5hz
132. ch_FP1---ch_T3---coher_5-6hz
133. ch_FP1---ch_T3---coher_6-7hz
134. ch_FP1---ch_T3---coher_7-8hz
135. ch_FP1---ch_T3---coher_8-9hz
136. ch_FP1---ch_T3---coher_9-10hz
137. ch_FP1---ch_T3---coher_10-11hz
138. ch_FP1---ch_T3---coher_11-12hz
139. ch_FP1---ch_T3---coher_12-13hz
140. ch_FP1---ch_T3---coher_13-14hz
141. ch_FP1---ch_T3---coher_14-15hz
142. ch_FP1---ch_T3---coher_15-16hz
143. ch_FP1---ch_T3---coher_16-17hz
144. ch_FP1---ch_T3---coher_17-18hz
145. ch_FP1---ch_T3---coher_18-19hz
146. ch_FP1---ch_T3---coher_19-20hz
147. ch_FP1---ch_T3---coher_20-21hz
148. ch_C3---ch_T3---coher_1-2hz
149. ch_C3---ch_T3---coher_2-3hz
150. ch_C3---ch_T3---coher_3-4hz
151. ch_C3---ch_T3---coher_4-5hz
152. ch_C3---ch_T3---coher_5-6hz
153. ch_C3---ch_T3---coher_6-7hz
154. ch_C3---ch_T3---coher_7-8hz
155. ch_C3---ch_T3---coher_8-9hz
156. ch_C3---ch_T3---coher_9-10hz
157. ch_C3---ch_T3---coher_10-11hz
158. ch_C3---ch_T3---coher_11-12hz
159. ch_C3---ch_T3---coher_12-13hz
160. ch_C3---ch_T3---coher_13-14hz
161. ch_C3---ch_T3---coher_14-15hz
162. ch_C3---ch_T3---coher_15-16hz
163. ch_C3---ch_T3---coher_16-17hz
164. ch_C3---ch_T3---coher_17-18hz
165. ch_C3---ch_T3---coher_18-19hz
166. ch_C3---ch_T3---coher_19-20hz
167. ch_C3---ch_T3---coher_20-21hz
168. ch_C3---ch_O1---coher_1-2hz
169. ch_C3---ch_O1---coher_2-3hz
170. ch_C3---ch_O1---coher_3-4hz
171. ch_C3---ch_O1---coher_4-5hz
172. ch_C3---ch_O1---coher_5-6hz
173. ch_C3---ch_O1---coher_6-7hz
174. ch_C3---ch_O1---coher_7-8hz
175. ch_C3---ch_O1---coher_8-9hz
176. ch_C3---ch_O1---coher_9-10hz
177. ch_C3---ch_O1---coher_10-11hz
178. ch_C3---ch_O1---coher_11-12hz
179. ch_C3---ch_O1---coher_12-13hz
180. ch_C3---ch_O1---coher_13-14hz
181. ch_C3---ch_O1---coher_14-15hz
182. ch_C3---ch_O1---coher_15-16hz
183. ch_C3---ch_O1---coher_16-17hz
184. ch_C3---ch_O1---coher_17-18hz
185. ch_C3---ch_O1---coher_18-19hz
186. ch_C3---ch_O1---coher_19-20hz

¹⁸ RMS – efektivní hodnota

¹⁹ line length - délka křivky

²⁰ zero crossing – počet průchodů nulou

²¹ coher - koherence

187. ch_C3---ch_O1---coher_20-21hz
188. ch_T3---ch_O1---coher_1-2hz
189. ch_T3---ch_O1---coher_2-3hz
190. ch_T3---ch_O1---coher_3-4hz
191. ch_T3---ch_O1---coher_4-5hz
192. ch_T3---ch_O1---coher_5-6hz
193. ch_T3---ch_O1---coher_6-7hz
194. ch_T3---ch_O1---coher_7-8hz
195. ch_T3---ch_O1---coher_8-9hz
196. ch_T3---ch_O1---coher_9-10hz
197. ch_T3---ch_O1---coher_10-11hz
198. ch_T3---ch_O1---coher_11-12hz
199. ch_T3---ch_O1---coher_12-13hz
200. ch_T3---ch_O1---coher_13-14hz
201. ch_T3---ch_O1---coher_14-15hz
202. ch_T3---ch_O1---coher_15-16hz
203. ch_T3---ch_O1---coher_16-17hz
204. ch_T3---ch_O1---coher_17-18hz
205. ch_T3---ch_O1---coher_18-19hz
206. ch_T3---ch_O1---coher_19-20hz
207. ch_T3---ch_O1---coher_20-21hz
208. ch_FP1---ch_O1---coher_1-2hz
209. ch_FP1---ch_O1---coher_2-3hz
210. ch_FP1---ch_O1---coher_3-4hz
211. ch_FP1---ch_O1---coher_4-5hz
212. ch_FP1---ch_O1---coher_5-6hz
213. ch_FP1---ch_O1---coher_6-7hz
214. ch_FP1---ch_O1---coher_7-8hz
215. ch_FP1---ch_O1---coher_8-9hz
216. ch_FP1---ch_O1---coher_9-10hz
217. ch_FP1---ch_O1---coher_10-11hz
218. ch_FP1---ch_O1---coher_11-12hz
219. ch_FP1---ch_O1---coher_12-13hz
220. ch_FP1---ch_O1---coher_13-14hz
221. ch_FP1---ch_O1---coher_14-15hz
222. ch_FP1---ch_O1---coher_15-16hz
223. ch_FP1---ch_O1---coher_16-17hz
224. ch_FP1---ch_O1---coher_17-18hz
225. ch_FP1---ch_O1---coher_18-19hz
226. ch_FP1---ch_O1---coher_19-20hz
227. ch_FP1---ch_O1---coher_20-21hz
228. ch_FP2---ch_C4---coher_1-2hz
229. ch_FP2---ch_C4---coher_2-3hz
230. ch_FP2---ch_C4---coher_3-4hz
231. ch_FP2---ch_C4---coher_4-5hz
232. ch_FP2---ch_C4---coher_5-6hz
233. ch_FP2---ch_C4---coher_6-7hz
234. ch_FP2---ch_C4---coher_7-8hz
235. ch_FP2---ch_C4---coher_8-9hz
236. ch_FP2---ch_C4---coher_9-10hz
237. ch_FP2---ch_C4---coher_10-11hz
238. ch_FP2---ch_C4---coher_11-12hz
239. ch_FP2---ch_C4---coher_12-13hz
240. ch_FP2---ch_C4---coher_13-14hz
241. ch_FP2---ch_C4---coher_14-15hz
242. ch_FP2---ch_C4---coher_15-16hz
243. ch_FP2---ch_C4---coher_16-17hz
244. ch_FP2---ch_C4---coher_17-18hz
245. ch_FP2---ch_C4---coher_18-19hz
246. ch_FP2---ch_C4---coher_19-20hz
247. ch_FP2---ch_C4---coher_20-21hz
248. ch_FP2---ch_T4---coher_1-2hz
249. ch_FP2---ch_T4---coher_2-3hz
250. ch_FP2---ch_T4---coher_3-4hz
251. ch_FP2---ch_T4---coher_4-5hz
252. ch_FP2---ch_T4---coher_5-6hz
253. ch_FP2---ch_T4---coher_6-7hz
254. ch_FP2---ch_T4---coher_7-8hz
255. ch_FP2---ch_T4---coher_8-9hz
256. ch_FP2---ch_T4---coher_9-10hz
257. ch_FP2---ch_T4---coher_10-11hz
258. ch_FP2---ch_T4---coher_11-12hz
259. ch_FP2---ch_T4---coher_12-13hz
260. ch_FP2---ch_T4---coher_13-14hz
261. ch_FP2---ch_T4---coher_14-15hz
262. ch_FP2---ch_T4---coher_15-16hz
263. ch_FP2---ch_T4---coher_16-17hz
264. ch_FP2---ch_T4---coher_17-18hz
265. ch_FP2---ch_T4---coher_18-19hz
266. ch_FP2---ch_T4---coher_19-20hz
267. ch_FP2---ch_T4---coher_20-21hz
268. ch_C4---ch_T4---coher_1-2hz
269. ch_C4---ch_T4---coher_2-3hz
270. ch_C4---ch_T4---coher_3-4hz
271. ch_C4---ch_T4---coher_4-5hz
272. ch_C4---ch_T4---coher_5-6hz
273. ch_C4---ch_T4---coher_6-7hz
274. ch_C4---ch_T4---coher_7-8hz
275. ch_C4---ch_T4---coher_8-9hz
276. ch_C4---ch_T4---coher_9-10hz
277. ch_C4---ch_T4---coher_10-11hz
278. ch_C4---ch_T4---coher_11-12hz
279. ch_C4---ch_T4---coher_12-13hz
280. ch_C4---ch_T4---coher_13-14hz
281. ch_C4---ch_T4---coher_14-15hz
282. ch_C4---ch_T4---coher_15-16hz
283. ch_C4---ch_T4---coher_16-17hz
284. ch_C4---ch_T4---coher_17-18hz
285. ch_C4---ch_T4---coher_18-19hz
286. ch_C4---ch_T4---coher_19-20hz
287. ch_C4---ch_T4---coher_20-21hz
288. ch_C4---ch_O2---coher_1-2hz
289. ch_C4---ch_O2---coher_2-3hz
290. ch_C4---ch_O2---coher_3-4hz
291. ch_C4---ch_O2---coher_4-5hz
292. ch_C4---ch_O2---coher_5-6hz
293. ch_C4---ch_O2---coher_6-7hz
294. ch_C4---ch_O2---coher_7-8hz
295. ch_C4---ch_O2---coher_8-9hz
296. ch_C4---ch_O2---coher_9-10hz
297. ch_C4---ch_O2---coher_10-11hz
298. ch_C4---ch_O2---coher_11-12hz
299. ch_C4---ch_O2---coher_12-13hz
300. ch_C4---ch_O2---coher_13-14hz
301. ch_C4---ch_O2---coher_14-15hz
302. ch_C4---ch_O2---coher_15-16hz

303. ch_C4---ch_O2---coher_16-17hz
304. ch_C4---ch_O2---coher_17-18hz
305. ch_C4---ch_O2---coher_18-19hz
306. ch_C4---ch_O2---coher_19-20hz
307. ch_C4---ch_O2---coher_20-21hz
308. ch_T4---ch_O2---coher_1-2hz
309. ch_T4---ch_O2---coher_2-3hz
310. ch_T4---ch_O2---coher_3-4hz
311. ch_T4---ch_O2---coher_4-5hz
312. ch_T4---ch_O2---coher_5-6hz
313. ch_T4---ch_O2---coher_6-7hz
314. ch_T4---ch_O2---coher_7-8hz
315. ch_T4---ch_O2---coher_8-9hz
316. ch_T4---ch_O2---coher_9-10hz
317. ch_T4---ch_O2---coher_10-11hz
318. ch_T4---ch_O2---coher_11-12hz
319. ch_T4---ch_O2---coher_12-13hz
320. ch_T4---ch_O2---coher_13-14hz
321. ch_T4---ch_O2---coher_14-15hz
322. ch_T4---ch_O2---coher_15-16hz
323. ch_T4---ch_O2---coher_16-17hz
324. ch_T4---ch_O2---coher_17-18hz
325. ch_T4---ch_O2---coher_18-19hz
326. ch_T4---ch_O2---coher_19-20hz
327. ch_T4---ch_O2---coher_20-21hz
328. ch_FP2---ch_O2---coher_1-2hz
329. ch_FP2---ch_O2---coher_2-3hz
330. ch_FP2---ch_O2---coher_3-4hz
331. ch_FP2---ch_O2---coher_4-5hz
332. ch_FP2---ch_O2---coher_5-6hz
333. ch_FP2---ch_O2---coher_6-7hz
334. ch_FP2---ch_O2---coher_7-8hz
335. ch_FP2---ch_O2---coher_8-9hz
336. ch_FP2---ch_O2---coher_9-10hz
337. ch_FP2---ch_O2---coher_10-11hz
338. ch_FP2---ch_O2---coher_11-12hz
339. ch_FP2---ch_O2---coher_12-13hz
340. ch_FP2---ch_O2---coher_13-14hz
341. ch_FP2---ch_O2---coher_14-15hz
342. ch_FP2---ch_O2---coher_15-16hz
343. ch_FP2---ch_O2---coher_16-17hz
344. ch_FP2---ch_O2---coher_17-18hz
345. ch_FP2---ch_O2---coher_18-19hz
346. ch_FP2---ch_O2---coher_19-20hz
347. ch_FP2---ch_O2---coher_20-21hz
348. ch_FP1---ch_FP2---coher_1-2hz
349. ch_FP1---ch_FP2---coher_2-3hz
350. ch_FP1---ch_FP2---coher_3-4hz
351. ch_FP1---ch_FP2---coher_4-5hz
352. ch_FP1---ch_FP2---coher_5-6hz
353. ch_FP1---ch_FP2---coher_6-7hz
354. ch_FP1---ch_FP2---coher_7-8hz
355. ch_FP1---ch_FP2---coher_8-9hz
356. ch_FP1---ch_FP2---coher_9-10hz
357. ch_FP1---ch_FP2---coher_10-11hz
358. ch_FP1---ch_FP2---coher_11-12hz
359. ch_FP1---ch_FP2---coher_12-13hz
360. ch_FP1---ch_FP2---coher_13-14hz
361. ch_FP1---ch_FP2---coher_14-15hz
362. ch_FP1---ch_FP2---coher_15-16hz
363. ch_FP1---ch_FP2---coher_16-17hz
364. ch_FP1---ch_FP2---coher_17-18hz
365. ch_FP1---ch_FP2---coher_18-19hz
366. ch_FP1---ch_FP2---coher_19-20hz
367. ch_FP1---ch_FP2---coher_20-21hz
368. ch_C3---ch_C4---coher_1-2hz
369. ch_C3---ch_C4---coher_2-3hz
370. ch_C3---ch_C4---coher_3-4hz
371. ch_C3---ch_C4---coher_4-5hz
372. ch_C3---ch_C4---coher_5-6hz
373. ch_C3---ch_C4---coher_6-7hz
374. ch_C3---ch_C4---coher_7-8hz
375. ch_C3---ch_C4---coher_8-9hz
376. ch_C3---ch_C4---coher_9-10hz
377. ch_C3---ch_C4---coher_10-11hz
378. ch_C3---ch_C4---coher_11-12hz
379. ch_C3---ch_C4---coher_12-13hz
380. ch_C3---ch_C4---coher_13-14hz
381. ch_C3---ch_C4---coher_14-15hz
382. ch_C3---ch_C4---coher_15-16hz
383. ch_C3---ch_C4---coher_16-17hz
384. ch_C3---ch_C4---coher_17-18hz
385. ch_C3---ch_C4---coher_18-19hz
386. ch_C3---ch_C4---coher_19-20hz
387. ch_C3---ch_C4---coher_20-21hz
388. ch_O1---ch_O2---coher_1-2hz
389. ch_O1---ch_O2---coher_2-3hz
390. ch_O1---ch_O2---coher_3-4hz
391. ch_O1---ch_O2---coher_4-5hz
392. ch_O1---ch_O2---coher_5-6hz
393. ch_O1---ch_O2---coher_6-7hz
394. ch_O1---ch_O2---coher_7-8hz
395. ch_O1---ch_O2---coher_8-9hz
396. ch_O1---ch_O2---coher_9-10hz
397. ch_O1---ch_O2---coher_10-11hz
398. ch_O1---ch_O2---coher_11-12hz
399. ch_O1---ch_O2---coher_12-13hz
400. ch_O1---ch_O2---coher_13-14hz
401. ch_O1---ch_O2---coher_14-15hz
402. ch_O1---ch_O2---coher_15-16hz
403. ch_O1---ch_O2---coher_16-17hz
404. ch_O1---ch_O2---coher_17-18hz
405. ch_O1---ch_O2---coher_18-19hz
406. ch_O1---ch_O2---coher_19-20hz
407. ch_O1---ch_O2---coher_20-21hz
408. ch_T3---ch_T4---coher_1-2hz
409. ch_T3---ch_T4---coher_2-3hz
410. ch_T3---ch_T4---coher_3-4hz
411. ch_T3---ch_T4---coher_4-5hz
412. ch_T3---ch_T4---coher_5-6hz
413. ch_T3---ch_T4---coher_6-7hz
414. ch_T3---ch_T4---coher_7-8hz
415. ch_T3---ch_T4---coher_8-9hz
416. ch_T3---ch_T4---coher_9-10hz
417. ch_T3---ch_T4---coher_10-11hz
418. ch_T3---ch_T4---coher_11-12hz

419. ch_T3---ch_T4---coher_12-13hz
 420. ch_T3---ch_T4---coher_13-14hz
 421. ch_T3---ch_T4---coher_14-15hz
 422. ch_T3---ch_T4---coher_15-16hz
 423. ch_T3---ch_T4---coher_16-17hz
 424. ch_T3---ch_T4---coher_17-18hz
 425. ch_T3---ch_T4---coher_18-19hz
 426. ch_T3---ch_T4---coher_19-20hz
 427. ch_T3---ch_T4---coher_20-21hz
 428. ch_FP1---ch_FP2---max(abs(xcorr))²²
 429. ch_FP1---ch_FP2---mean(abs(xcorr))
 430. ch_FP1---ch_C3---max(abs(xcorr))
 431. ch_FP1---ch_C3---mean(abs(xcorr))
 432. ch_FP1---ch_C4---max(abs(xcorr))
 433. ch_FP1---ch_C4---mean(abs(xcorr))
 434. ch_FP1---ch_T3---max(abs(xcorr))
 435. ch_FP1---ch_T3---mean(abs(xcorr))
 436. ch_FP1---ch_T4---max(abs(xcorr))
 437. ch_FP1---ch_T4---mean(abs(xcorr))
 438. ch_FP1---ch_O1---max(abs(xcorr))
 439. ch_FP1---ch_O1---mean(abs(xcorr))
 440. ch_FP1---ch_O2---max(abs(xcorr))
 441. ch_FP1---ch_O2---mean(abs(xcorr))
 442. ch_FP2---ch_C3---max(abs(xcorr))
 443. ch_FP2---ch_C3---mean(abs(xcorr))
 444. ch_FP2---ch_C4---max(abs(xcorr))
 445. ch_FP2---ch_C4---mean(abs(xcorr))
 446. ch_FP2---ch_T3---max(abs(xcorr))
 447. ch_FP2---ch_T3---mean(abs(xcorr))
 448. ch_FP2---ch_T4---max(abs(xcorr))
 449. ch_FP2---ch_T4---mean(abs(xcorr))
 450. ch_FP2---ch_O1---max(abs(xcorr))
 451. ch_FP2---ch_O1---mean(abs(xcorr))
 452. ch_FP2---ch_O2---max(abs(xcorr))
 453. ch_FP2---ch_O2---mean(abs(xcorr))
 454. ch_C3---ch_C4---max(abs(xcorr))
 455. ch_C3---ch_C4---mean(abs(xcorr))
 456. ch_C3---ch_T3---max(abs(xcorr))
 457. ch_C3---ch_T3---mean(abs(xcorr))
 458. ch_C3---ch_T4---max(abs(xcorr))
 459. ch_C3---ch_T4---mean(abs(xcorr))
 460. ch_C3---ch_O1---max(abs(xcorr))
 461. ch_C3---ch_O1---mean(abs(xcorr))
 462. ch_C3---ch_O2---max(abs(xcorr))
 463. ch_C3---ch_O2---mean(abs(xcorr))
 464. ch_C4---ch_T3---max(abs(xcorr))
 465. ch_C4---ch_T3---mean(abs(xcorr))
 466. ch_C4---ch_T4---max(abs(xcorr))
 467. ch_C4---ch_T4---mean(abs(xcorr))
 468. ch_C4---ch_O1---max(abs(xcorr))
 469. ch_C4---ch_O1---mean(abs(xcorr))
 470. ch_C4---ch_O2---max(abs(xcorr))
 471. ch_C4---ch_O2---mean(abs(xcorr))
 472. ch_T3---ch_T4---max(abs(xcorr))
 473. ch_T3---ch_T4---mean(abs(xcorr))
 474. ch_T3---ch_O1---max(abs(xcorr))
 475. ch_T3---ch_O1---mean(abs(xcorr))
 476. ch_T3---ch_O2---max(abs(xcorr))
 477. ch_T3---ch_O2---mean(abs(xcorr))
 478. ch_T4---ch_O1---max(abs(xcorr))
 479. ch_T4---ch_O1---mean(abs(xcorr))
 480. ch_T4---ch_O2---max(abs(xcorr))
 481. ch_T4---ch_O2---mean(abs(xcorr))
 482. ch_O1---ch_O2---max(abs(xcorr))
 483. ch_O1---ch_O2---mean(abs(xcorr))
 484. ch_FP1---ch_EMG1+---max(abs(xcorr))
 485. ch_FP1---ch_EMG1+---mean(abs(xcorr))
 486. ch_FP2---ch_EMG1+---max(abs(xcorr))
 487. ch_FP2---ch_EMG1+---mean(abs(xcorr))
 488. ch_C3---ch_EMG1+---max(abs(xcorr))
 489. ch_C3---ch_EMG1+---mean(abs(xcorr))
 490. ch_C4---ch_EMG1+---max(abs(xcorr))
 491. ch_C4---ch_EMG1+---mean(abs(xcorr))
 492. ch_T3---ch_EMG1+---max(abs(xcorr))
 493. ch_T3---ch_EMG1+---mean(abs(xcorr))
 494. ch_T4---ch_EMG1+---max(abs(xcorr))
 495. ch_T4---ch_EMG1+---mean(abs(xcorr))
 496. ch_O1---ch_EMG1+---max(abs(xcorr))
 497. ch_O1---ch_EMG1+---mean(abs(xcorr))
 498. ch_O2---ch_EMG1+---max(abs(xcorr))
 499. ch_O2---ch_EMG1+---mean(abs(xcorr))
 500. ch_FP1---ch_EOG1+---max(abs(xcorr))
 501. ch_FP1---ch_EOG1+---mean(abs(xcorr))
 502. ch_FP2---ch_EOG1+---max(abs(xcorr))
 503. ch_FP2---ch_EOG1+---mean(abs(xcorr))
 504. ch_C3---ch_EOG1+---max(abs(xcorr))
 505. ch_C3---ch_EOG1+---mean(abs(xcorr))
 506. ch_C4---ch_EOG1+---max(abs(xcorr))
 507. ch_C4---ch_EOG1+---mean(abs(xcorr))
 508. ch_T3---ch_EOG1+---max(abs(xcorr))
 509. ch_T3---ch_EOG1+---mean(abs(xcorr))
 510. ch_T4---ch_EOG1+---max(abs(xcorr))
 511. ch_T4---ch_EOG1+---mean(abs(xcorr))
 512. ch_O1---ch_EOG1+---max(abs(xcorr))
 513. ch_O1---ch_EOG1+---mean(abs(xcorr))
 514. ch_O2---ch_EOG1+---max(abs(xcorr))
 515. ch_O2---ch_EOG1+---mean(abs(xcorr))
 516. ch_FP1---ch_ECG1+---max(abs(xcorr))
 517. ch_FP1---ch_ECG1+---mean(abs(xcorr))
 518. ch_FP2---ch_ECG1+---max(abs(xcorr))
 519. ch_FP2---ch_ECG1+---mean(abs(xcorr))
 520. ch_C3---ch_ECG1+---max(abs(xcorr))
 521. ch_C3---ch_ECG1+---mean(abs(xcorr))
 522. ch_C4---ch_ECG1+---max(abs(xcorr))
 523. ch_C4---ch_ECG1+---mean(abs(xcorr))
 524. ch_T3---ch_ECG1+---max(abs(xcorr))
 525. ch_T3---ch_ECG1+---mean(abs(xcorr))
 526. ch_T4---ch_ECG1+---max(abs(xcorr))
 527. ch_T4---ch_ECG1+---mean(abs(xcorr))
 528. ch_O1---ch_ECG1+---max(abs(xcorr))
 529. ch_O1---ch_ECG1+---mean(abs(xcorr))
 530. ch_O2---ch_ECG1+---max(abs(xcorr))

²² xcorr – vzájemná korelace

531. ch_O2---ch_ECG1+---mean(abs(xcorr))
532. ch_FP1---ch_PNG1+---max(abs(xcorr))
533. ch_FP1---ch_PNG1+---mean(abs(xcorr))
534. ch_FP2---ch_PNG1+---max(abs(xcorr))
535. ch_FP2---ch_PNG1+---mean(abs(xcorr))
536. ch_C3---ch_PNG1+---max(abs(xcorr))
537. ch_C3---ch_PNG1+---mean(abs(xcorr))
538. ch_C4---ch_PNG1+---max(abs(xcorr))
539. ch_C4---ch_PNG1+---mean(abs(xcorr))
540. ch_T3---ch_PNG1+---max(abs(xcorr))
541. ch_T3---ch_PNG1+---mean(abs(xcorr))

542. ch_T4---ch_PNG1+---max(abs(xcorr))
543. ch_T4---ch_PNG1+---mean(abs(xcorr))
544. ch_O1---ch_PNG1+---max(abs(xcorr))
545. ch_O1---ch_PNG1+---mean(abs(xcorr))
546. ch_O2---ch_PNG1+---max(abs(xcorr))
547. ch_O2---ch_PNG1+---mean(abs(xcorr))

Příloha 2: Ukázka kmenového souboru vytvořeného systému

```

%% INICIALIZACE %%
clear all;
close all;
fclose('all');
clc;

%% CESTY %% !!!!! změnit v případě jiných dat
global_prom.main_path = 'C:\...';           % hlavní cesta
global_prom.path_define_file = 'C:\...';    % cesta k datům

%% !!!!! NASTAVENÍ !!!!! %%
nacist=0;           %nacteni z formátu arff
nacistload=1;      %nacteni z .mat
vykon=0;           %vypocet vykonnostnich krivek
vykonfigury=0;     %vykresleni vykonnostnich krivek
selekce=0;         %provedeni selekce priznaku
klasifikace=0;     %provedeni klasifikace

%% NAČTENÍ DAT %% if nacist == 1  nebo if nacistload == 1

%.ARFF
if nacist
...
end

%.MAT
if nacistload
...
end
%% VÝPOČET VÝKONNOSTNÍCH KŘÍVEK %% if vykon == 1
if vykon
...
end

%% SELEKCE %% if selekce == 1
if selekce
...
end

%% KLASIFIKACE %% if klasifikace == 1
if klasifikace
...
end

```