ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta elektrotechnická

Diplomová práce

Využití Bayesovského přístupu pro klasifikaci EKG

Bc. Tomáš Faflík

2014

Vedoucí práce: Ing. Jiří Spilka, PhD

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne …… ……

Podpis autora práce

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra kybernetiky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: Bc. Tomáš F a f l í k

Studijní program: Biomedicínské inženýrství a informatika (magisterský)

Obor: Biomedicínské inženýrství

Název tématu: Využití Bayesovského přístupu pro klasifikaci EKG

Pokyny pro vypracování:

Elektrokardiografie (EKG) je základní vyšetřovací metoda v kardiologii. Jejím principem je snímání elektrické srdeční aktivity a v podobě elektrokardiogramu (časový záznam EKG křivek) umožňuje její hodnocení. EKG vyšetření je většinou neinvazivní. Pomocí elektrod umístěných na kůži, ale i na stěně jícnu či přímo v srdci, měříme rozdíl napětí jako projev šíření akčního potenciálu myokardem. Protože je elektrická aktivita srdce podmínkou mechanické, má EKG důležitou diagnostickou roli u řady srdečních chorob. Hodnocení EKG záznamů nám umožňuje odhalovat arytmie, jako projevy poruch tvorby, či vedení vzruchu. Významnou roli hraje také při zjišťování ischemických změn, lokalizace i stádia infarktu myokardu. Změny na EKG nacházíme buď ve všech svodech, nebo jen v jednom, či skupině svodů, které spolu vzhledem k anatomii srdce souvisejí.
Cílem práce je seznámit se s problematikou zpracování EKG záznamů, navrhnout vhodný klasifikátor s využitím Bayesovského přístupu, implementovat a otestovat na reálných záznamech. Implementace by měla zároveň otestovat možnost využití nových technologií, jako je cloud computing.

1. Seznamte se s problematikou EKG a metodami jeho hodnocení.
2. Prostudujte využívané metody analýzy EKG signálu.
3. Navrhněte vhodné metody klasifikace EKG záznamů s využitím bayesovského přístupu. Zvolenou metodu implementujte a otestujte na reálných datech.
4. Porovnejte úspěšnost navržené metody s již existujícími.

Seznam odborné literatury:

[1] Mařík, V.; Štěpánková, O.; Lažanský, L. a kolektiv: Umělá inteligence 1-4. Academia Praha (1993, 1997, 2001, 2003).
[2] Murphy, Kevin: Dynamic Bayesian Networks: Representation, Inference and Learning. UC Berkeley, Computer Science Division (2002).
[3] Ben-Gal I.: Bayesian Networks, in Ruggeri F., Faltin F. & Kenett R., Encyclopedia of Statistics in Quality & Reliability, Wiley & Sons (2007).
[4] Richard E. Neapolitan: Learning Bayesian Networks. Prentice Hall (2003).

Vedoucí diplomové práce: Ing. Jiří Spilka

Platnost zadání: do konce zimního semestru 2014/2015

L.S.

doc. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 20. 8. 2013

Poděkování

Na tomto místě bych rád poděkoval vedoucímu diplomové práce Ing. Jiřímu Spilkovi, PhD a dále doc. Ing. Lence Lhotské, CSc. a Mgr. Tomáši Siegerovi za jejich vstřícnost, cenné rady a informace při tvorbě této práce.

Abstrakt

Tato práce se věnuje možnosti využití Bayesovských sítí při klasifikaci spodního infarktu myokardu. Poskytnutá předzpracovaná testovací data obsahovala 2596 EKG záznamů s 37 atributy, rozdělená v poměru 2333 normálních a 263 abnormálních signálů. V práci je postupně testováno několik běžných metod pro sestavení Bayesovských sítí, spolu s porovnáváním jejich výstupů. Znatelného zlepšení rozlišovacích schopností téměř všech algoritmů se podařilo docílit prořezáním kompletního souboru dat pomocí rozhodovacího stromu.

Nejlepšími sítěmi, vytvořených pomocí K2 a HillClimber algoritmu, se podařilo dosáhnout velmi dobrých výsledků se senzitivitou nad 81 % a specificitou nad 95 %.

Abstract

This thesis investigates the possibility of using Bayesian network structures for classification of inferior myocardial infarction. The test data had been preprocessed and contain 2.596 ECG records in 37 individual attributes divided into 2333 normal and 263 abnormal signals. We have successfully tested a multitude of methods of Bayesian network design and compared the results. The resolution of almost all algorithms has been significantly improved by cutting the complete dataset using decision tree.

The best classifiers were based on K2 and HillClimber algorithms where we reached sensitivity of more than 81 % and specificity of more than 95 %.

Obsah

Seznam obrázků

Seznam tabulek

,, The probability of any event is the ratio between the value at which an expectation depending on the happening of the event ought to be computed, and the chance of the thing expected upon its happening “

T. Bayes.

1701 - 1761

1. Úvod

Medicína je dnes velmi široký pojem, propojující stále více různých, někdy zdánlivě neslučitelných, oborů. Jednou z oblastí, kde velmi rychle narůstá její význam, je zpracování dat. Ve spojení se zlepšujícím se technickým zázemím narůstá objem informací, které je potřeba vyhodnocovat. Tato práce se zabývá možností strojového zpracování pořízených dat u spodního infarktu myokardu.

Jednotlivým teoretickým částem se postupně věnuji v kapitole 3 (*stručný rozbor biologických signálů*), kapitole 4 (*úvod do Bayesovských sítí*) a kapitole 5 (*úvod do předzpracování dat*). V kapitole 6 jsou popsány jednotlivé experimenty a metody, které byly použity k sestavení sítí.

Testovací data jsem měl k dispozici z 12-ti svodového systému EKG, poskytnutého firmou Medical Technologies CZ a. s. Původní soubor obsahoval 6332 záznamů. Pro tuto práci jsem již dostal data předpřipravená a pročištěná s celkovým počtem 2596 záznamů, obsahujících 2333 normálních a 263 abnormálních signálů. Data byla sbírána mezi roky 2004 až 2007 pomocí přístroje 12BTL-08 LC EKG s vzorkovací frekvencí 500 Hz a přesností 3,9 μV.

2. Cíl práce

Základním cílem práce je využití struktury zvané Bayesovská síť ve zdravotnictví. Tato implementace konkrétně testuje využití sítě pro posouzení signálu EKG při vyhodnocování, zda pacient prodělal spodní infrakt (*inferior myocardial infarction*). V dnešní době se k tomuto vyhodnocení používá naměřený EKG signál, se kterým nadále pracuje lékař. U něho je nutná expertní znalost problematiky, zkušenosti a jistý druh citu. Naproti tomu výpočetní technika prodělala v posledních několika desetiletích raketový vzestup výkonu a zejména v minulém desetiletí i neuvěřitelný nárůst informačních databází. Díky tomu můžeme dnes vidět velké posuny v oboru umělé inteligence. Příkladem může být superpočítač IBM Watson, který dokáže přirozeně komunikovat s lidmi (*předvedl to v období televizní soutěže Riskuj*). Jeho největším technologickým pokrokem je propracovaný datamining informací dostupných z internetu.

To je technologická špička dnešní doby. V medicíně je to ale se zaváděním nejmodernějších technologií do praxe vždy pomalejší. Důvodem je nutná jistota, že daný algoritmus či lék bude ideálně bezchybný a bezpečný, protože se jedná o lidské životy. Doba přechodu přístrojů z čistě analogových na digitální se v medicíně stává již samozřejmostí a číslicové zpracování přináší řadu benefitů. Díky vysokému výkonu přístrojů nám je například umožněno provádět vyšetření na magnetické rezonanci či počítačové tomografii prakticky v reálném čase.

Jedním z důležitých prvků dnešní medicíny je práce s naměřenými signály. Spolu s evolucí a prostupem techniky do medicínských oborů se zvyšuje význam zpracování tohoto důležitého zdroje dat. Úprava probíhá v několika krocích, počínaje základním pročištěním dat od zjevně chybných případů, přes různé druhy filtrací a odstraňování rušivých fragmentů, extrakcí jednotlivých příznaků až po vlastní klasifikaci. Jedná se o velmi složitý řetězec událostí, které je potřeba provést pro co nejlepší přesnost finálního vyhodnocení.

Dalším oborem je právě umělá inteligence, která se snaží obsáhnout jak odborné znalosti, tak i zkušenosti lékařů. Výhodou takových systémů je pak jejich nezávislost. Systém nezná únavu a nepociťuje žádné osobní problémy. Velmi účinně může obor umělé inteligence pomáhat lékařům při stanovení diagnóz u pacientů. Speciálně pak rozhodovací stromy, neuronové sítě či Bayesovské sítě. Rozhodovací stromy jsou pro toto použití vhodné jen částečně, protože jejich síla je současně i poměrně velkou slabinou. Stromy se dokážou rozhodovat velmi striktně, to však nemusí být vždy žádoucí. Naproti tomu Neuronové

a Bayesovské sítě pracují s pravděpodobnostními tabulkami a jejich výsledky nemusí být vždy tak přímočaré. To otevírá možnosti pro jistý druh strojové emulace lékařské praxe.

3. Teoretický úvod, biologické signály

K základním projevům života všech organismů patří vydávání různých signálů. Škála typů signálů je veliká, počínaje zvukovými projevy až po objemové změny různých orgánů. Různé signály je tedy nutné měřit specializovanými technikami, podle jejich fyzikální podstaty.

Základní rozdělení biologických signálů podle fyzikální podstaty:

- Elektrické projevy
- Magnetické projevy
- Ostatní projevy

Pod ostatní projevy řadíme sledování všeho, co nemá jako výstup elektrickou veličinu a musíme využít převodník na veličinu elektrickou – strojově detekovatelnou.

Pro mapování specifických signálů se zrodily různé vědní obory. Zaměřují se na danou frekvenční a napěťovou oblast a ideálně pouze na místo, kde je možné signál naměřit. To vše z důvodů vyšší přesnosti.

3.1 Základní měření elektrických projevů biologických signálů

- EKG – elektrokardiogram (*rozsahy napětí do 5 mV a frekvence do 150 Hz*)
- EEG – elektroencefalogram (*rozsahy napětí do 100 µV a frekvence do 80 Hz*)
- EMG – elektromyogram (*rozsah napětí je do 1mV a frekvence do 100 Hz*)
- EOG – elektrookulogram (*rozsahy napětí jsou do 1mV, frekvence do 100 Hz*)
- ERG – elektroretinogram (*rozsahy napětí do 100 µV a frekvence do 50 Hz*)
- EGG – elektrogastrogram (*rozsahy napětí jsou 100 µV a frekvence cca 3 Hz*)

Z uvedených rozsahů napětí a frekvencí můžeme vidět, že signály v lidském těle se velmi mísí. Je tedy celkem obtížné snímat pouze veličinu, která nás zajímá, většinou se k ní přidává okolní šum. Šum ovšem může být způsoben taktéž okolím (*nejtypičtějším příkladem je síťová frekvence 50 Hz*). Ten je nutné nějakým způsobem odstranit ještě před tím, než začneme data zpracovávat. Jednou z možností je využití filtrů. Pokud známe průběh rušícího signálu, můžeme díky této znalosti nežádoucí signál odstranit.

Dále máme měření magnetických projevů (*MKG, MEG, MMG, MRG*) či projevů, které nemají elektromagnetický výstup (*například pletysmogram, kde je jako zdroj signálu změna objemu*).

Zmíněné elektrické signály jsou samozřejmě spojité a pro účely číslicového zpracování je nutné provést diskretizaci.

3.2 Signál EKG

Zdrojem tohoto signálu jsou srdeční potenciály. Měří se pomocí povrchových elektrod a detekovatelné rozsahy napětí jsou do 5 mV a frekvence řádově do 150 Hz.

3.3 Princip elektrokardiografie

Vlastní měření EKG signálu je běžně neinvazivní vyšetření, ačkoliv pro vyšší přesnost lze provádět i přímo na srdci. Srdeční sval při svém stahu vyvolává malé napětí, které se následně šíří tělem k povrchu kůže a zde jej detekujeme elektrodami. Toto napětí je dáno rozdílem stavů na různých místech srdce při šíření akčního potenciálu. Výsledkem měření je elektrokardiogram, což je záznam elektrické aktivity srdce v čase. Tato kapitola se svým obsahem opírá zejména o knihu Bioelectomagnetism (*J. Malmivuo, R. Plansey*) zdroj [8].

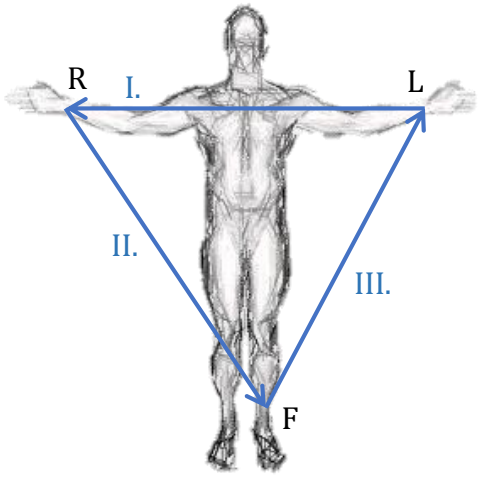
Pro správně provedené měření křivky EKG je nutné dodržet korektní zapojení svorek – svodů. Dnes se používá několik svodových systémů.

3.3.1 Einthovenovy svody I., II., III.

Bipolární končetinové svody, měřící změny potenciálu mezi dvěma elektrodami. Jednotlivá místa pro přiložení jsou označena

- R – pravá ruka (*obvykle červená svorka*)
- L – levá ruka (*obvykle žlutá svorka*)
- F – kotník levé nohy (*značeno zelenou elektrodou*)
- N – neutrální elektroda (*černá*) fungující jako uzemnění

Jako I. svod je označován signál vznikající mezi elektrodami R-L, II. svod vzniká mezi elektrodami R-F a III. svod mezi elektrodami L-F.

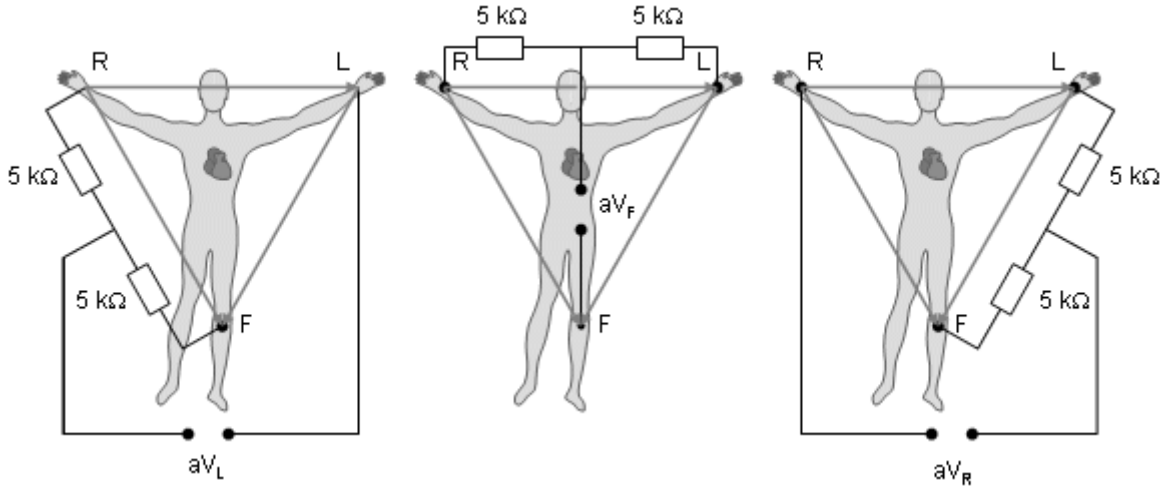


Obrázek 1 - Zapojení Einthovenových svodů

Při tomto zapojení nám vzniká tzv. Einthovenův trojúhelník, který je přibližně rovnostranný, svírající úhly 60° mezi stranami. Díky vzniklému tvaru můžeme detekovat vektor elektrické srdeční osy. Ten se dopočítává v závislosti na velikostech vln a polaritě v jednotlivých vrcholech R-L-F zapojeného obvodu.

3.3.2 Goldbergerovy svody

Goldbergerovy augmentované svody aVL, aVR, aVF jsou unipolární končetinové a měří změny potenciálů mezi elektrodou a příslušnou svorkou, vznikající spojením zbylých protilehlých elektrod.



Obrázek 2 - Zapojení Goldbergových svodů; zdroj [8]

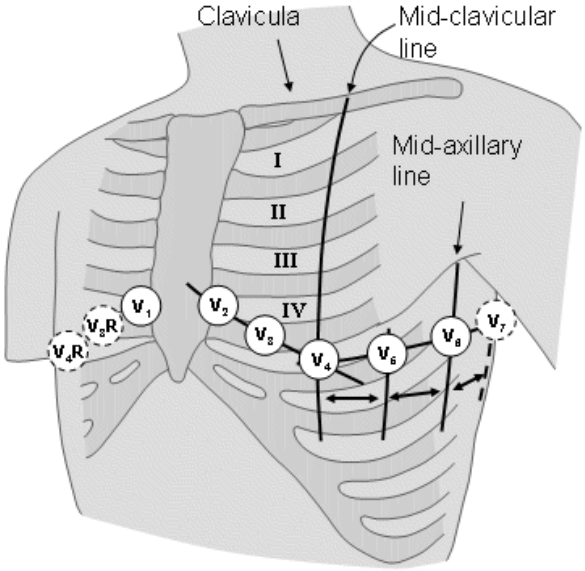
Samy tyto svody nesnímají rozdíly proti elektrodám, ale vůči nulovému potenciálu, vznikajícího spojením všech elektrod přes Wilsonovu svorku s odporem 5 kΩ.

V současné době se používají jako doplnění pro Einthovenovy svody a vytváří 6 os, které jsou vzájemně pootočené o 30° a dávají nám jemnější škálu pro nalezení vektoru srdeční osy.

3.3.3 Wilsonovy svody

Wilsonovy svody V1 – V6 jsou unipolární hrudní, měřící změny potenciálu mezi elektrodou a svorkou, vznikající spojením všech 3 končetinových svodů.

Při měření signálu musíme kompenzovat chyby a nepřesnosti vznikající různými způsoby. Například v okolí 1 Hz leží harmonické složky, které jsou způsobeny pohybovými artefakty.

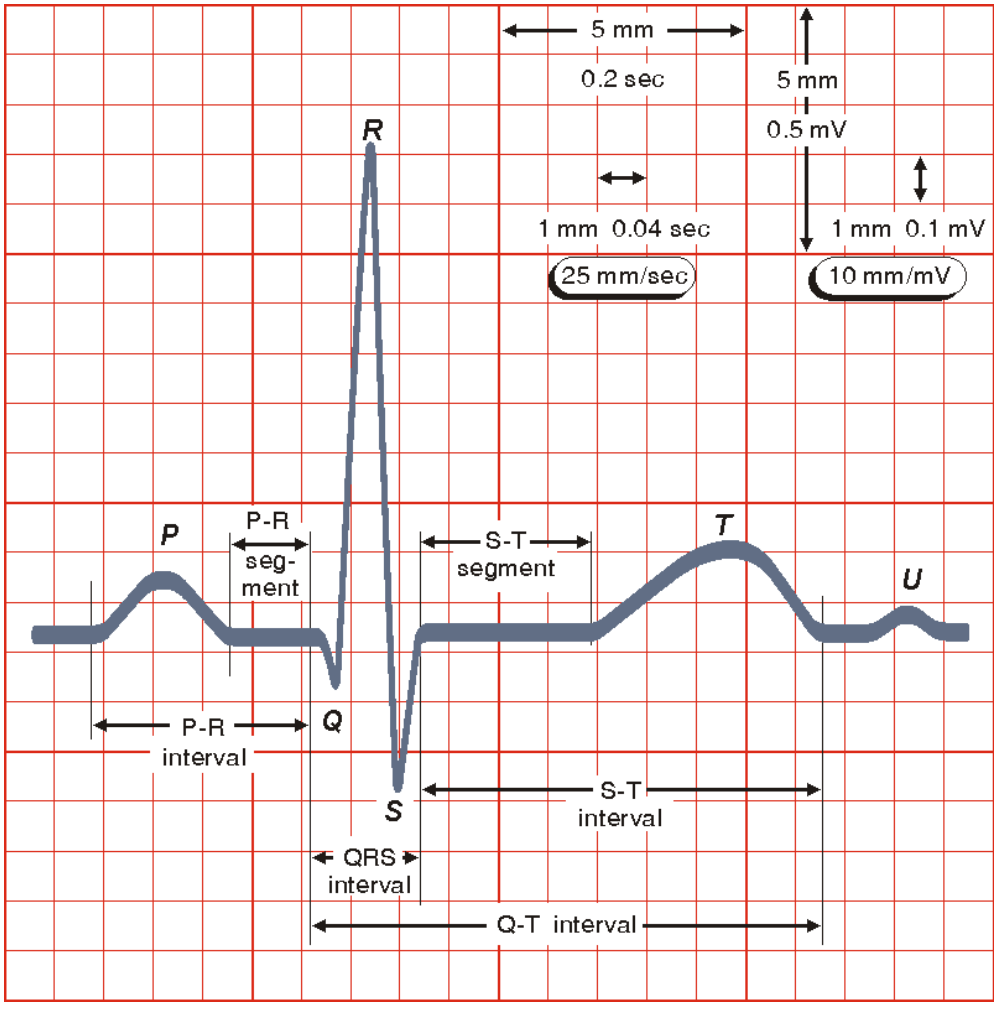


Obrázek 3 - Zapojení Wilsonových svodů na těle, zdroj [8]

Všechny jmenované svodové systémy se v dnešní době vzájemně doplňují a propojují do tzv. 12ti svodového zapojení EKG, pro co nejlepší přesnost při pořizování elektrokardiogramu.

3.4 Popis EKG křivky

Křivka EKG je periodický signál, který je vyvoláván srdeční činností. Po správném odměření, je možné z charakteristiky usuzovat, zdali je dotyčný člověk v pořádku či trpí nějakými zdravotními obtížemi. Signály by měli dodržovat určitou charakteristiku, která sice může mít různě veliké amplitudy, ale špičky v signálu by měli být v určitém vzájemném poměru.



Obrázek 4 - běžná křivka EKG signálu, zdroj [8]

1. P vlna – vzniká při depolarizaci síní srdce.
2. Q šička – začínají se depolarizovat srdeční komory
3. R špička – hrotová depolarizace, směr vektoru depolarizace se kryje s anatomickou osou srdce
4. S špička – pozdní depolarizace levé komory
5. T vlna – je způsobena repolarizací komor

Tyto úseky jsou typické pro zdravého jedince a jejich běžné hodnoty jsou uvedené v tabulce (*Tabulka 1*).

	Doba trvání (s)	Amplituda (mV)
P vlna	0,06 – 0,11	< 0,25
QRS komplex	< 0,12	0,8 – 1,2
S-T segment	0,12	
T vlna	1,16	< 0,5

Tabulka 1 - Porovnání hodnot jednotlivých úseků EKG signálu

3.5 Elektrody

Obrázek 5 - Příklady různých typů elektrod; zdroj [10]

Živý organismus a elektronické přístroje jsou rozdílné typy vodičů. Lidské tělo je vodičem 2. třídy, kde elektrická energie probíhá na bázi iontů. Naproti tomu měřicí přístroje jsou vodiči první třídy, kde se náboj předává elektrony. Na rozhraní dochází ke změně typu vodivosti. Musí tedy docházet k předání iontů mezi elektrodou a elektrolytem a následné oxidační reakci, kdy se z materiálu uvolní elektrony.

Možné připojení elektrod na člověka závisí na jejich provedení. Elektrody je možné použít jehlové přímo na měřený orgán, přisávací nebo jednorázově přilepit (*viz Obrázek 5*).

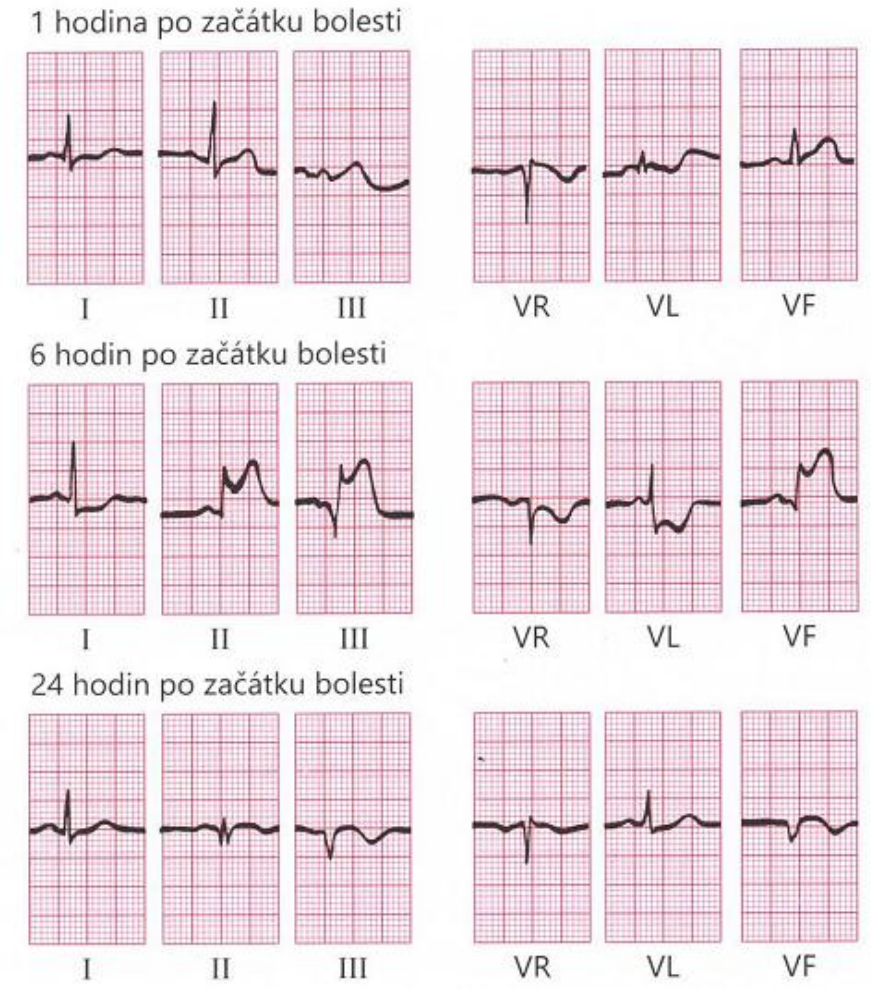
Jako materiály pro nejběžnější povrchové elektrody se používají kovy obalené těžko rozpustnou solí ponořené do elektrolytu, se kterým mají společný aniont. Dochází tak ke kombinovanému rozhraní *elektroda – elektrolyt - měřená osoba*. Na rozhraní elektrolyt – povrch těla je vhodné použít vodivý gel, který vyrovná nerovnosti kůže.

3.6 Abnormality EKG při infarktu myokardu

U infarktu myokardu spodní stěny se změny na EKG signálu projevují na svodech III a aVF. Detekovatelný postupný sled změn je vždy následující:

1. Elevace úseku S-T
2. Vznik kmitů Q
3. Normalizace úseku S-T
4. Inverze T vlny

Tento postupný proces se většinou pohybuje v rozmezí 24 – 48 hodin. Inverze vlny T je většinou trvalá (*zdroj [15]*). Pokud elevace úseku S-T nastává po S kmitu, pak se může jednat o normální EKG signál, zvaný *high take off*. Důležité je vyhledávat souvislost se jmenovanou T vlnou. V případech, kdy infarkt nepostihne celou šířku srdeční stěny, nemusí dojít ke vzniku kmitů Q a rovnou proběhne inverze vlny T.



Obrázek 6 - Rozvoj infarktu myokardu spodní stěny; zdroj [15]

4. Teoretický úvod, Bayesovské sítě

Jedná se o pravděpodobnostní model systému, který je vizualizován jako orientovaný graf. Náhodné veličiny jsou reprezentovány jako *uzly* a pravděpodobnostní závislosti jako *hrany*. Každému uzlu je vypočtená a přiřazená pravděpodobnostní distribuce, která je závislá na svých předcích a má tvar (4.1).

$$P(u|parents(u)) \tag{4.1}$$

Můžeme tedy vidět, že každý uzel je přímo závislý na všech předcházejících uzlech, ze kterých přicházejí hrany.

Pro celou Bayesovskou síť existuje právě jedna společná pravděpodobnostní distribuce a lze ji vypočítat podle vzorce (4.2).

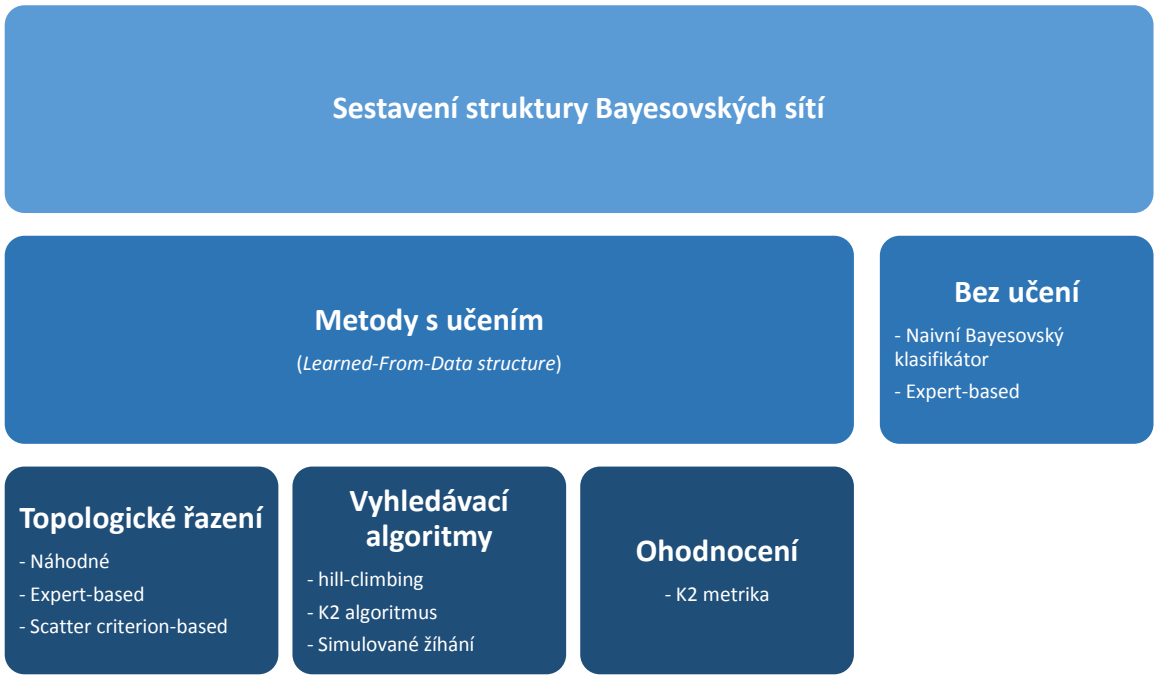
$$R((X_i)_{i\subset V}) = \prod_{i\in V} P\left(X_i|(X_j)_{j\subset pa(i)}\right) \tag{4.2}$$

Bayesovská síť se používá v systémech, u nichž je potřebné využití pravděpodobnostního odvozování. Pokud je nám známa struktura sítě a pravděpodobnostní distribuce v jednotlivých uzlech, pak můžeme jednoduše vypočítat aposteriorní pravděpodobnost libovolného uzlu. Nejjednodušším příkladem Bayesovské sítě je naivní Bayesovský algoritmus. Ten je ostatně možné použít jako inicializační strukturu sítě pro vytváření BN. Naivní Bayesovský klasifikátor předpokládá, že veškeré parametry mezi sebou nemají žádné vazby a přímo ovlivňují rozdělení třídy. V některých případech nám tato představa může postačovat a dokáže podávat dobré výsledky. U spousty aplikací však jenom toto vyobrazení nestačí. Velmi často se stává, že pravděpodobnostní rozdělení jednoho atributu je přímo závislé na jiném atributu. Pro tyto případy se konstruují Bayesovské sítě.

U BN se potýkáme s problémem, jak najít vazby mezi jednotlivými příznaky. Druhým extrémem (*na rozdíl od naivního Bayese bez vazeb*) je přiřazení vazeb mezi všemi atributy. Tady narážíme na extrémní výpočetní složitost, kdy s každým přidaným atributem exponenciálně narůstá obtížnost zpracování. Proto je vhodné najít vazby mezi příznaky

vhodným algoritmem nebo některé atributy (*pokud jsou nevhodné či korelují s jinými*) předem vyřadit.

Pro ohodnocení uzlů a hran je nutné síť nastavit. Základní možnosti nastavení máme v dvě (*viz Obrázek 7*).



Obrázek 7 - Metody pro nastavení Bayesovských sítí, zdroj [1]

4.1 Expert-Based Structure

Pokud jsme detailně seznámeni s problematikou, můžeme si dovolit strukturu sítě napřímo vytvořit. V praxi to znamená identifikovat proměnné a určit závislosti mezi nimi. Síť si následně odvodí pouze pravděpodobnostní ohodnocení. Tento způsob je sice velmi intuitivní, ale nastavení je limitováno na malé sítě či dobře známé problémy. Další nevýhodou může být právě nastavení expertních znalostí, protože expertní znalost není u většiny případů exaktně definovaná. Postupy sestavené různými experty se proto nemusí shodovat. Výstupy z této sítě taktéž nemusí být přesné, protože nastavení je prováděno pro dané prostředí. Pokud se parametry okolí změní, pak je nutné síť znovu kalibrovat, případně se zamyslet nad správností propojení jednotlivých uzlů nebo uzly samotnými.

4.2 Learned-From-Data Structure

Základem tohoto nastavení je nechat vše odvodit z trénovacích dat. Toto se nám hodí v případě, že máme velký počet vstupních dat nebo expertní znalosti řešeného problému jsou

omezené či neznámé. Je také možné následné dotvoření sítě právě částečnými znalostmi. Učení Bayesovské sítě probíhá buď pomocí učení s omezeními (*constraint-based*) nebo metodou prohledávání s využitím skórovací funkce (*search&score*).

Metoda učení s omezeními využívá při konstrukci BN základní statické testy jako je chí-kvadrát nebo vzájemná výměna informací (*mutual information*) k nalezení podmíněně nezávislých vztahů mezi jednotlivými proměnnými.

Naproti tomu S&S metoda zahrnuje 2 úkony. Vyhledávací procedura nejprve navrhne síť a následně vyhodnocovací operace vypočte ohodnocení pro každou z nalezených částí. Pro malé sítě je možné použití metody hrubé síly (*brute-force*), díky které najdeme nejvhodnější způsob zapojení, ale zaplatíme za to časovou náročností (*v tomto případě je velikost sítě doporučena řádově do 5ti uzlů*). S každým dalším uzlem roste náročnost výpočtu exponenciálně. Při velkém počtu prvků nám tedy nezbyde nic jiného, než použít vhodnou heuristickou funkci pro hledání.

4.3 Vyhledávací algoritmy používané u Bayesovských sítí

4.3.1 Hill Climbing a Repeated Hill Climbing

Proces hledání začíná stanoveným bodem v prostoru neorientovaného grafu. Tento bod je většinou zvolen náhodně, ale záleží na konkrétní implementaci.

- Uzel je expandován, vyhledají se jeho nejbližší sousedé a ty jsou následně ohodnoceny.
- Vybere se uzel s nejvyšším ohodnocením a ten je dále expandován.
- Tento cyklus expanze uzlů pokračuje do té doby, kdy mají všechny sousední uzly ohodnocení horší, než poslední expandovaný. Tento bod se stává výsledným.

Zásadním nedostatkem tohoto algoritmu je jeho náchylnost na uváznutí v lokálním extrému. Tento problém lze minimalizovat tím, že kód spustíme vícekrát a pokaždé vybereme jiný vstupní uzel (*právě Repeated Hill Climbing*). Samozřejmě záleží na počtu opakování.

4.3.2 LAGD Hill Climbing

Look Ahead Hill Climbing – vylepšená verze předchozího algoritmu, která se snaží eliminovat uváznutí v lokálním extrému pomocí dopředného prohledávání. Prohledávání se provádí v omezené sadě nejlepších výsledků (*podobně, jako čistý HillClimbing*) a současně s jejich následnými expanzemi. Hloubku expanze a množství vybraných bodů můžeme parametricky nastavit.

4.3.3 K2

V dnešní době je to asi nepoužívanější algoritmus pro Bayesovské sítě, publikovaný v roce 1992 (*G. F. Cooper, E. Herskovits*). Jedná se o hladový algoritmus, který se snaží maximalizovat pravděpodobnost shody modelu Bayesovské sítě a zkoumaných dat (*Search&Score metoda*).

Na začátku jsou všechny uzly bez rodičů. Algoritmus prochází jednotlivými uzly a k nim se vytváří množina kandidátů na jejich rodiče. Ty musí maximalizovat ohodnocení uzlu. Potenciální rodiče se do množiny nepřidávají v případě, že již množina obsahuje maximální počet kandidátů na rodiče nebo se nadále nezvyšuje hodnocení sítě. Použití je limitováno na diskrétní rozdělení hodnot a vstupní data musí být úplná. S tím je nutné se důsledně vypořádat v části předzpracování dat. Možným problémem zůstává uváznutí

v lokálním maximu. Další, který vzniká z podstaty K2 algoritmu, je otázka určení maximálního počtu rodičů (*resp. kandidátů na rodiče*).

K2 metrika, používaná pro ohodnocení sítě, se řídí Bayesovskou funkcí a je posteriorní pravděpodobností struktury G, daná náhodným výběrem prvku D ze společné distribuce X (*rovnice (4.3)*).

$$P(G|D) = \frac{P(D|G) \cdot P(G)}{P(D)} = \frac{P(G,D)}{P(D)} \tag{4.3}$$

4.3.4 Simulated Annealing

Metoda, česky nazývaná simulované žíhání, je velmi náročná na výpočet. V mém případě se datový soubor (*viz kapitola 6.2*) počítal 1,5 hodiny a potřeboval něco přes 17 GB RAM. V principu je sice možné algoritmus implementovat pro paralelní zpracování, v programu se mi ale nepodařilo tento režim vynutit.

Postup vychází z fyzikálního principu žíhání oceli. Žíhání je proces při kterém těleso umístěné v peci při vysoké teplotě začínáme velmi pomalu zchlazovat. Díky tomu postupně dochází k žádanému uspořádání vnitřní struktury materiálu a tím i stabilní a pevné konfigurace.

Kód využívá této podoby, kdy s vysokou počáteční teplotou se dějí přeskoky minimalizující uváznutí v lokálním extrému. Skoky jsou vyvolávány náhodně (*viz vzorec (4.4)*) a přímo závisí na funkci *random* v hranicích <0,1>. Pro náhodnou funkci se ideálně využívá Gaussovo rozdělení. Postupnému snižování proměnné teploty je přímo úměrné zkracování velikosti náhodných přeskoků. Kód se tedy postupně dostává do optimální pozice.

$$P(x \to x_0) = \begin{cases} 1 & , \text{pro } f(x) < f(x_0) \\ e^{\frac{-(f(x)-f(x_0))}{T}} & , \text{pro } f(x) \geq f(x_0) \end{cases} \tag{4.4}$$

Ze vzorce (4.4) vidíme, že pokud má stav aktuální *x* menší hodnotu, než stav x_0, pak tento nový stav prohlásíme za nejlepší. V opačném případě se do výběru vkládá náhodná funkce, která s určitou pravděpodobností rozhodne, zdali stav bude i tak použit nebo ne.

Akceptace nového stavu v tomto případě závisí na nerovnici (4.5). Pokud nerovnost platí, pak dojde k použití nového stavu.

$$rand(0;1) < e^{\frac{-(f(x)-f(x_0))}{T}} \tag{4.5}$$

Důležitým prvkem je již zmíněná teplota, která svojí velikostí silně ovlivňuje pravděpodobnost přeskoku. Pokud je teplota v kontrastu se jmenovatelem exponenciální funkce vzorce (4.5) velmi vysoká, pak dochází k tomu, že pravá strana nerovnosti se limitně blíží k hodnotě 1 a na levé straně v tu chvíli nezáleží na pravděpodobnostním rozdělení – téměř vždy bude menší než 1.

Postup popsaný výše se nazývá Metropolisův algoritmus a ten se pro každou teplotu opakuje. Hodnotu, kolik iterací má cyklus provést, je možné nastavit. Nejčastěji se využije počet sousedních uzlů. Ovšem i zde je možné nastavit variabilní počet (*v závislosti na teplotě*) nebo je možné využití Markovových řetězců.

Pro použití v simulovaném žíhání se tento základní algoritmus používá iterativně s postupným snižováním teploty. Výstup z jedné iterace o určité teplotě se následně použije jako vstupní parametr pro další cyklus s nižší teplotou.

Částečný kód algoritmu SA v jazyce PHP

```
function SimulatedAnnealing()
{
  $state = initState();

  // pro jednoduchost teplotu snižuji o 1 stupeň. V praxi je pokles teploty
  //pozvolnější (řádově desetiny stupně)
  for ($t = initTemperature(); $t > 0; $t--)
  {
    // $n_t - počet opakování Metropolisova algoritmu; Většinou se použije
    //        stejný počet, jako je počet sousedních uzlů.
    for ($i = 0; $i < $n_t; $i++)
    {
      $state = MetropolisAlg($state, $t);
    }
    // snížíme teplotu
    $t = coolDown($t);
  }
}

function MetropolisAlg($state, $t)
{
  $new = getNextNeighbourRnd($state); // načtení náhodného souseda
  $delta = cost($new) - cost($state);
  if ($delta < 0)
  {
    return $new;
  }
  else
  {
    $r = rand(0, 1); // náhodná hodnota z rozsahu <0,1>
    if($r < exp(-$delta/$t))
    {
      // pokud je $t >> 0 || $delta velmi malá
      // využijeme nový stav
      return $new;
    }
    else
    {
      // jinak využíváme stav původní
      return $state;
    }
  }
}
```

Výsledky simulovaného žíhání vycházejí v porovnání s ostatními technikami v dnešní době nejlépe[1]. Ovšem je zde nutné připomenout vyšší výpočetní náročnost v porovnání s ostatními metodami.

[1] dle Experiments with new stochastic global optimization search techniques, Computers and Operations Research, 27:841-865, 2000

4.4 Naučení pravděpodobnostních tabulek

Ve chvíli, kdy máme za pomoci jmenovaných algoritmů strukturu sítě vytvořenou, je nutné nastavení pravděpodobnostních tabulek. V použitém programu WEKA máme k dispozici základní funkci SimpleEstimator, která vytváří přímé odhady podmíněných pravděpodobností podle vzorce (4.6)

$$P(x_i = k|Pa(x_i) = j) = \frac{N_{ijk} + N'_{ijk}}{N_{ij} + N'_{ij}} \qquad (4.6)$$

Kde N'_{ijk} je nastavitelný parametr, který určuje věrohodnost odhadů. V základu má hodnotu 0,5 a při nastavení hodnoty na 0 dostáváme maximálně věrohodný odhad. V programu Weka se parametr nazývá Alfa (*parametr je dále použit u nastavení všech měření*).

Druhou funkcí je BMAEstimator, s kterým dostáváme odhady podmíněné pravděpodobnosti založené na Bayesovském průměrování celé struktury sítě. Toho se dosahuje odhadem pravděpodobnostní tabulky uzlu, který je dán váženým průměrem všech podmíněných pravděpodobností jeho rodičů. Pro vyhodnocení se využívá BDe metrika nebo K2 metrika.

5. Teoretický úvod, předzpracování dat

Následujícím problémem jsou vlastní data. Ty mohou obsahovat (*a ve většině případů obsahují*) chybné hodnoty, které jsou různého charakteru a vznikají mnoha způsoby.

- Chybějící hodnoty
- Odlehlá pozorování
- Systematická chyba daná měřením

5.1 Chybějící hodnoty a odlehlá pozorování

Jednou z možností jsou chybějící hodnoty. Příkladem může být například odmítnutí sdělení informace pacientem nebo nemožnost hodnotu změřit.

Při počátečním zpracování dat můžeme použít různá řešení. Záleží primárně na tom, jak dalece nám zásah do nekompletních dat naruší přesnost. Jednou z možností je vynechat měření (*pacienty*), u kterých se hodnota nevyskytuje nebo je na první pohled špatně. To ovšem platí pouze tehdy, jedná-li se o ojedinělé případy a zpravidla desetiny procent z celkového souboru poskytnutých dat.

Další možností je tento atribut uměle nahradit. Zde máme možnost využít buď průměru, nebo námi vybrané hodnoty, která je u podobných měření zastoupena nejčastěji.

Pokud si nemůžeme být jisti správností a nechceme zatížit zpracování potenciální chybou, pak můžeme neznámou hodnotu nahradit příznakem, který ponese sám údaj o nevyplnění. V takových hodnotách může být informace ukryta také. Příkladem je již dříve zmíněné úmyslné zamlčení informace pacientem. Poslední z možností je nezohledňovat atribut jako celek a vystačit si s ostatními. Pokud ale máme málo atributů na porovnání, pak je vhodné využít jednu z metod doplnění.

5.2 Systematická chyba měření

Tato chyba nám nastane v případě, že hodnotu měříme nevhodným postupem nebo přístroj samotný měří s chybou. Zdůrazňuji, že se jedná o měření mimo hodnoty pokryté nejistotou přístroje.

Opět je nutné zohlednit účel následného zpracovávání daného atributu. Pokud se vlastnost vyhodnocuje absolutně, pak je buď nutné provést korekci (*v případě, že známe odchylku*) nebo atribut dále nezohledňovat. Pokud se data používají relativně (*pouze mezi*

sebou v souboru měření) nebo konstantně vždy se stejným posunem (*např. na jednom pracovišti*), pak je možné data použít. Může ale nastat problém s odhalováním takové chyby.

5.3 Počet pozorování

Dalším problémem, vznikajícím při vytváření učících se algoritmů nebo rozhodovacích stromů, je počet hodnot, které máme k dispozici. Úlohy, pro které jsou tyto výpočetní techniky určené primárně, většinou nereflektují skutečné rozložení problému.

Příkladem může být problém „Zjištění, kolik procent populace v ČR je aktuálně nemocná?“. Pokud se rozhodneme, že budeme data vyhodnocovat z výstupů od lékařů, pak logicky dojdeme k názoru, že je drtivá většina populace nemocná, protože k lékaři dochází většinou akutně nemocní lidé.

Kromě toho narážíme i na problém velikosti naměřených dat. Těch máme vždy omezený počet a musíme se rozhodnout, jakým způsobem je rozdělíme na trénovací a testovací množinu. V případě, že všechna data použijeme na natrénování sítě, pak nebudeme mít možnost náš postup otestovat a naopak, když natrénujeme velmi malým počtem dat, pak bude průběh velmi nepřesný.

V tomto případě je nutné zvolit vhodný poměr pro obě množiny. Příkladem může být rozdělení vstupního setu dat na 2 skupiny o velikosti 50 %. Polovinu použijeme na nastavení a druhou na otestování. Velmi zajímavou metodou zlepšení, pokud máme k dispozici málo vstupních dat, je křížová validace (*cross-validation*).

Metoda spočívá v tom, že rozdělíme množinu na 2 části, ale jednotlivé prvky vybereme náhodně. Poté, co provedeme natrénování a otestování, pokus opakujeme (*předpokladem je odlišné rozdělení množin*). Nakonec výstupy ze všech iterací porovnáváme.

Metoda křížové validace mi přišla jako nejefektivnější, proto jsem ji pro návrhy sítí zvolil. Parametr opakování jsem nastavil na hodnotu 10.

5.4 Statistická významnost

Pojem statistická významnost je pravděpodobnostní měřítko, které pomáhá odpovědět na otázku, zdali výsledky provedených měření jsou dány náhodně nebo je výsledek skutečně věrohodný. Pokud uvažujeme standardní hranici 5%, pak nám hodnota říká, že s pravděpodobností menší než 5% výsledek vznikl náhodou. Měřítko tak vypovídá o skutečném výsledku a nikoliv o čisté náhodě.

5.5 Chyby I. a II. druhu

Každý algoritmus, který má za úkol práci s pravděpodobnostními funkcemi, produkuje chybová rozhodnutí. Naším cílem je samozřejmě tyto chyby co nejvíce minimalizovat, ovšem nikdy je nelze plně vyloučit. Tuto chybu ve statistice dále dělíme na chybu prvního druhu a chybu druhého druhu (*Tabulka 2*).

	H_0 je nepravdivá	**H_0 je pravdivá**
Odmítnutí H_0	Pravdivě pozitivní	Falešně pozitivní (chyba I. druhu)
Neodmítnutí H_0	Falešně negativní (chyba II. druhu)	Pravdivě negativní

Tabulka 2 - Výsledky možných výsledků testu na odmítnutí nulové hypotézy

U většiny operací je jedna z chyb kritická, naproti tomu druhou můžeme do jisté míry akceptovat. Důležité je ale rozhodnutí, která chyba je pro náš případ podstatnější a kterou budeme částečně umožňovat.

Jako příklad použiji situaci z nemocničního prostředí - testujeme pacienty na nějakou těžkou chorobu (*Tabulka 3*).

	Pacient není zdravý	**Pacient je zdravý**
Nechat pacienta na pozorování	Ano	Ne
Poslat pacienta domů	Ne	Ano

Tabulka 3 - Modelová situace chyby I. a II. druhu

Na výše uvedeném příkladu můžeme celkem jednoduše pochopit rozdíl v druzích chyb. Pokud u pacienta vyhodnocujeme, zdali danou chorobou trpí, můžeme se dopustit chybné klasifikace. Za prvé můžeme označit zdravého pacienta za nemocného (*v tomto případě chyba I. druhu*). Druhou chybou, která může nastat, je odeslání nemocného pacienta domů. Na misku vah se tak dostává situace, kdy pacienta vystavíme stresu proti situaci, kdy jej odešleme domů, což ale může mít fatální následky. Je tedy zřejmé, že oba typy chyb si nejsou rovnocenné. Pro každou implementaci je nutné zjistit, kterou chybu musíme

minimalizovat. V reálných provozech je například možné algoritmus kalibrovat tak, aby úmyslně preferoval jeden typ chyby nad druhým.

Tyto chyby se často vyobrazují na tzv. ROC křivce, která obsahem plochy pod ní udává vztah mezi specificitou a senzitivitou měření.

5.5.1 Specificita

Specificita je hodnota, která nám udává poměr reálně negativních výsledků vůči celkovému počtu negativních výsledků, včetně falešně pozitivních chyb. Jinými slovy nám tato vlastnost říká, s jakou přesností dokážeme z celku vybrat hodnotu, u které sledovaný příznak nenastává.

5.5.2 Senzitivita

Tato hodnota určuje, jak úspěšný je test v nalezení nemocné osoby. Pro tuto práci je senzitivita klíčová a je tedy ideální ji u zpracovávaných algoritmů maximalizovat. Maximální senzitivita testu by nám zaručila, že nalezneme všechny pacienty i za cenu chyby prvního druhu, kdy označíme některé zdravé jedince za nemocné. Této hodnoty samozřejmě nelze v normálních podmínkách dosáhnout, důležité je ale rozhodnutí, kterou z hodnot je nutné pro daný problém maximalizovat.

5.6 Vytvoření vlastních příznaků

Jednou z možností, jak pomoci vlastnímu algoritmu ve zlepšení zpracování dat, je vytvoření vlastního parametru za pomoci operací nad známými atributy. Pro vytvoření vlastního příznaku můžeme využít základních matematických operací jako sčítání, násobení / dělení, ale i logaritmování a podobně. Cílem této operace je odlišení jednotlivých podskupin dat od sebe.

5.7 Odstranění korelovaných dat

Naproti tomu se v datech mohou vyskytnout příznaky, které vyhodnocení zatěžují výpočetní náročností a v porovnání s jiným atributem nepřináší žádnou informaci navíc. Takové příznaky jsou jednoduše viditelné v zobrazených grafech tak, že mají přímou závislost atributů (*respektive malý rozptyl*) tj. dají se proložit přímkou. V těchto případech pomáhá jistá expertní znalost v oboru, díky které můžeme závislé atributy vyřadit ještě před počátkem zpracování.

6. Experimentální data

K vyhotovení práce jsem dostal vyextrahovaná a signálově předzpracovaná data. Ty obsahují měření různých veličin a svodů.

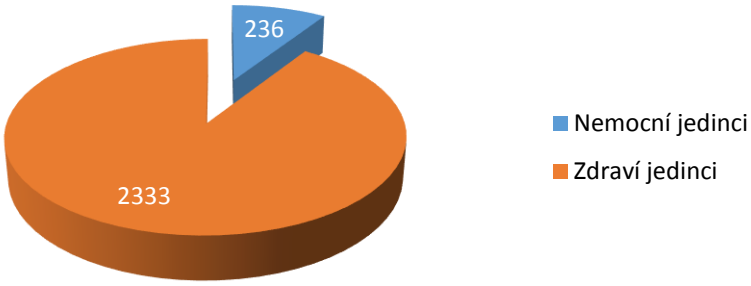
II_Q_Amplitude_uV	Amplituda Q vlny na II. Einthovenově svodu (μV)
II_Q_Duration_ms	Doba trvání Q vlny na II. Einthovenově svodu (ms)
II_Q_Position_ms	Pozice začátku Q vlny II. svodu (ms)
II_R0_Amplitude_uV	Základní amplituda první špičky R vlny II. svodu (μV)
II_R1_Amplitude_uV	Amplituda špičky R1 vlny R na II. svodu (μV)
II_R2_Amplitude_uV	Amplituda špičky R2 vlny R na II. svodu (μV)
II_R_Duration_ms	Doba trvání vlny R (ms)
II_S_Amplitude_uV	Amplituda S vlny II. svodu (μV)
II_S_Duration_ms	Doba trvání vlny S na II. svodu (ms)
II_QRS_Amplitude_uV	Velikost amplitudy QRS komplexu detekovaného na II. svodu (μV)
II_RdivQ	Podílová hodnota R vlny a Q vlny ze II. svodu
II_RdivS	Podílová hodnota R vlny a S vlny ze II. svodu
III_Q_Amplitude_uV	Amplituda Q vlny na III. Einthovenově svodu (μV)
III_Q_Duration_ms	Doba trvání Q vlny na III. Einthovenově svodu (ms)
III_Q_Position_ms	Pozice začátku Q vlny III. svodu (ms)
III_R0_Amplitude_uV	Základní amplituda první špičky R vlny III. svodu (μV)
III_R1_Amplitude_uV	Amplituda špičky R1 vlny R na III. svodu (μV)
III_R2_Amplitude_uV	Amplituda špičky R2 vlny R na III. svodu (μV)
III_R_Duration_ms	Doba trvání R vlny na III. svodu (ms)
III_S_Amplitude_uV	Velikost amplitudy S vlny III. svodu (μV)
III_S_Duration_ms	Doba trvání vlny S na III. svodu (ms)
III_QRS_Amplitude_uV	Velikost amplitudy QRS komplexu detekovaného na III. svodu (μV)
III_RdivQ	Podílová hodnota R vlny a Q vlny ze III. svodu
III_RdivS	Podílová hodnota R vlny a S vlny ze III. svodu
aVF_Q_Amplitude_uV	Amplituda Q vlny na svodu aVF (μV)
aVF_Q_Duration_ms	Doba trvání Q vlny na svodu aVF (ms)
aVF_Q_Position_ms	Pozice začátku Q vlny na svodu aVF (ms)
aVF_R0_Amplitude_uV	Základní amplituda první špičky R vlny aVF svodu (μV)
aVF_R1_Amplitude_uV	Amplituda špičky R1 vlny R na aVF svodu (μV)
aVF_R2_Amplitude_uV	Amplituda špičky R2 vlny R na aVF svodu (μV)
aVF_R_Duration_ms	Doba trvání R vlny, měřeno na aVF svodu (ms)
aVF_S_Amplitude_uV	Amplituda S vlny na svodu aVF (μV)
aVF_S_Duration_ms	Délka S vlny na aVF svodu (ms)
aVF_QRS_Amplitude_uV	amplituda QRS komplexu na svodu aVF (μV)
aVF_RdivQ	Podílová hodnota R vlny a Q vlny z aVF svodu
aVF_RdivS	Podílová hodnota R vlny a S vlny z aVF svodu
class	Binomické rozdělení pacienta do třídy (1 - nemocný, 0 - zdravý)

Dále v textu využívám vyjmenovaných zkratek, pro jejich krátký tvar.

6.1 Popis dat

Základní datový set ke zpracování jsem dostal ve formátu *arff*. Ke zpracování jsem se rozhodl využít program WEKA (*vhodný pro Bayesovské sítě*) a pro částečné předzpracování a vizualizaci dat program RapidMiner. Druhý jmenovaný programem jsem vybral, protože jsem s ním pracoval v průběhu studia a dle mého subjektivního názoru je jednodušší na ovládání.

Prvotní náhled na data nám poskytne základní rozložení zdravých a nemocných jedinců.



Obrázek 8 - Celkové rozložení dat

Celkový počet poskytnutých dat je 2596, je zde 36 přítomných atributů a 1 definiční atribut třídy. Výstupní atribut nabývá pouze binomického rozdělení (*zdravý/nemocný*). Proto je možné pro zpracování obdobných dat použít Bayesovskou síť (*viz Tabulka 4*).

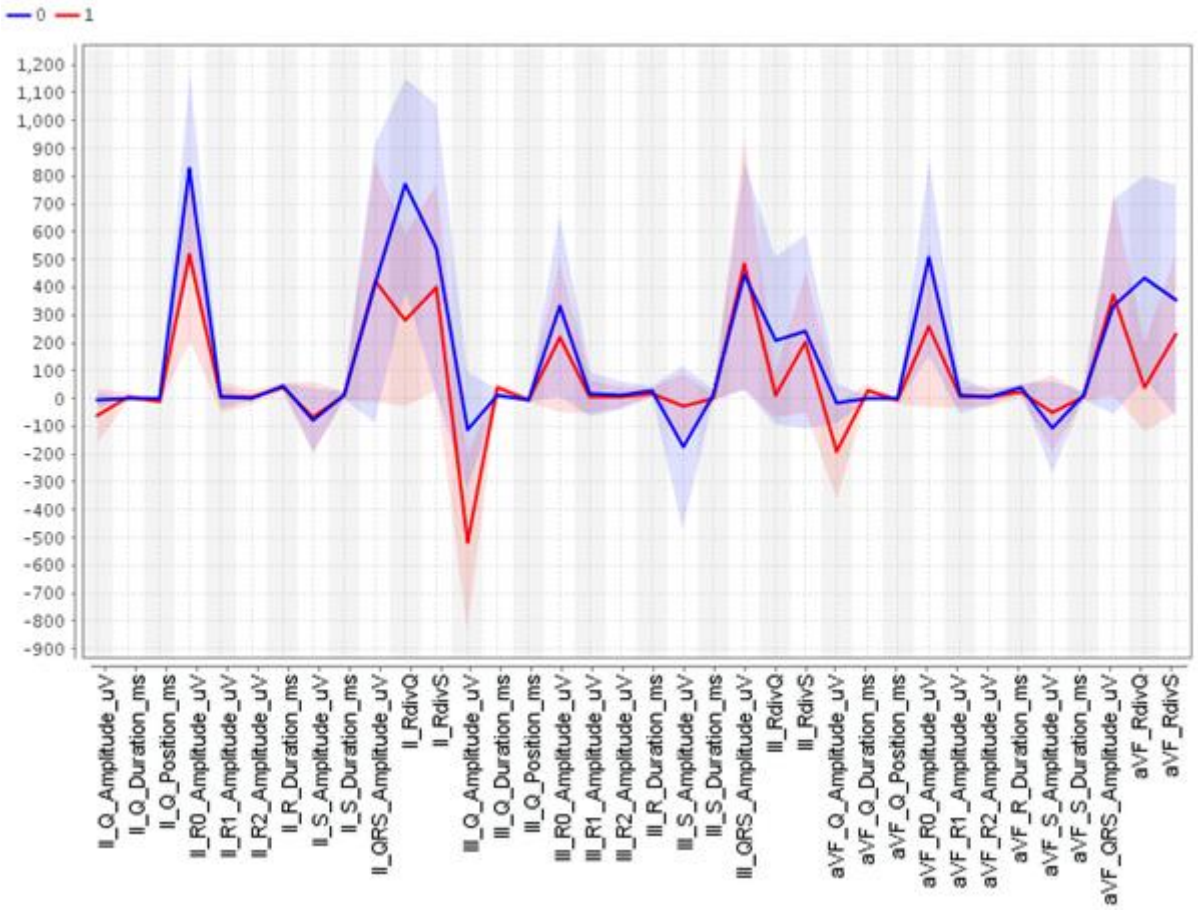
Příznaky	Odezva			
	Spojitá	Diskrétní	Nominální	Binominální
Spojitá	regrese		k-NN Naivní Bayes Bayesovská síť rozhodovací strom	Perceptron
Diskrétní				
Nominální				
Binominální				

Tabulka 4 - Možnosti použití algoritmů podle charakteru dat

6.2 Náhled na data

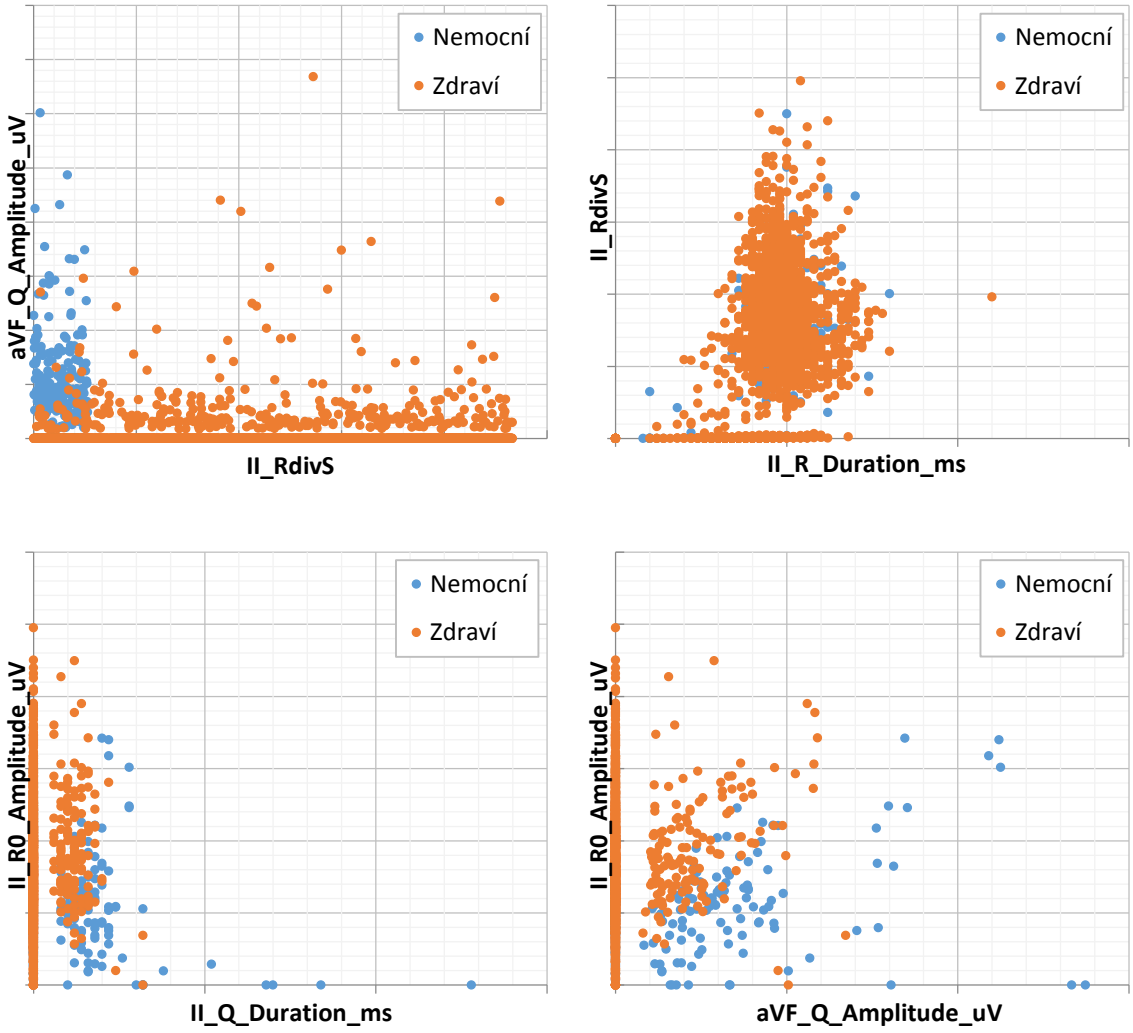
Jako základní pohled na data jsem použil graf deviací. Díky tomu si můžeme celkem jednoduše udělat představu o možných shlucích či spíše o rozložení jednotlivých atributů. Patrné je několik základních věcí. Za nejlépe rozložené atributy můžeme považovat II_RdivQ a aVF_RdivQ. Naproti tomu atribut II_Q_Duration_ms je pro obě výsledné skupiny prakticky totožný, kdy i rozptyl hodnot se vzájemně překrývá. Usuzuji, že jeho podíl na výsledku bude velmi malý.

Cílem tohoto zobrazení by ideálně měly být 2 vzájemně neprolínající se skupiny.



Obrázek 9 - Graf rozložení výsledků u jednotlivých atributů

Při bližším zkoumání dat je viditelné, že všechny měřené atributy se zde prolínají, což znamená, že v grafech nenalezneme jednoznačně definovatelné shluky.



Obrázek 10 - Příklad rozložení některých atributů $\begin{bmatrix} a, b \\ c, d \end{bmatrix}$

Z vybraných grafů si můžeme udělat představu o rozložení jednotlivých pacientů na škále příznaků. Celkový počet grafů je samozřejmě 630, což je pro vložení do této práce neúnosné a z velké části naprosto zbytečné, protože valná většina z nich nemá výpovědní hodnotu. Vybrané 4 grafy jsem zvolil z toho důvodu, že mají nejlépe čitelné shluky. Ovšem ani u jednoho nemůžeme s jistotou hranice shluků určit. V tomto ohledu vypadá velmi dobře graf *a*, který má podle mého hodnocení nejlépe viditelné hranice. Je patrné, že i tak se prvky z obou skupin nachází mimo své shluky.

Nicméně pokud nemůžeme od sebe přímo obě skupiny oddělit, můžeme použít pravděpodobnostní vyjádření nebo se pokusit vhodnou kombinací sestavit vlastní atributy, které budou požadavky splňovat.

Jako odrazový můstek pro kombinaci vlastních atributů jsem zvolil graf rozložení výstupů (*Obrázek 9*), kde jsem se snažil zkombinovat atributy tak, aby se nemocní a zdraví jedinci dostali do samostatných množin, které se neprotínají. Nejvhodnější se tedy zdají atributy II_RdivQ a aVF_RdivQ, které jsem zmínil dříve. Bohužel nově testované atributy rozdělení ještě více zhoršily. Proto jsem se nakonec rozhodl tuto metodu nevyužít.

6.3 Zpracování kompletního setu dat

6.3.1 HillClimber, plný datový soubor

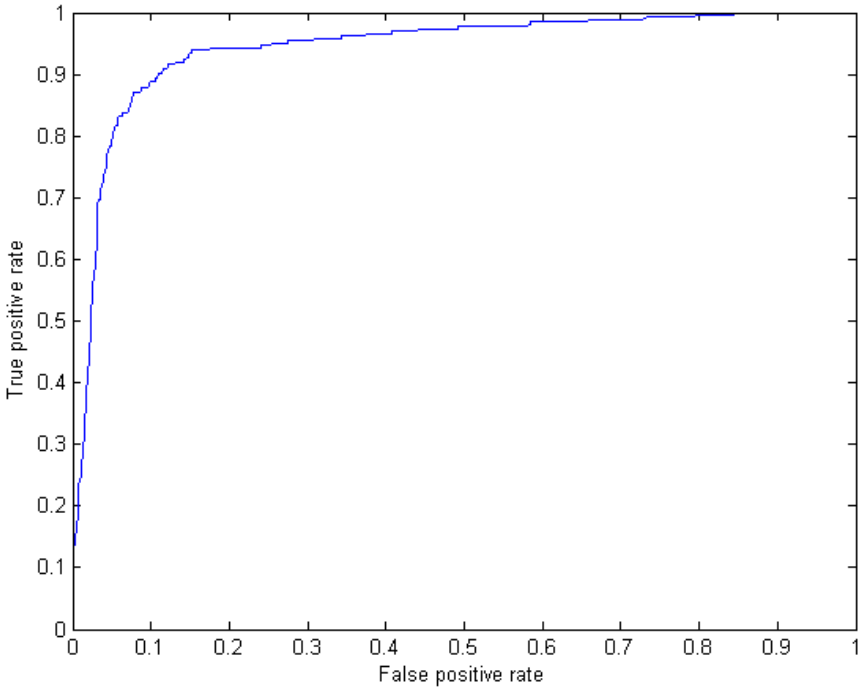
U každého z následujících zpracování jsem se pokoušel nastavovat hodnoty experimentálně. Výstupů bylo pro tuto práci až zbytečně mnoho, proto jsem se rozhodl využít pouze nejlepší výstupy z dané kategorie. Následují tak měření všech probraných vyhledávacích algoritmů a zpracování pomocí naivního Bayese, s kterým měření BN porovnávám. Na přiloženém CD jsou pak k dispozici i ostatní výstupy, které jsem během experimentů vytvořil. Exportované soubory se dají načíst rovněž v přiloženém programu WEKA (*Sekce: Explorer > Classify > Result list > pravým tlačítkem myši a Load Model*).

Estimator:	Simple Estimator	Alfa = 0,1
Vyhledávací algoritmus:	HillClimber	
Použití AD stromu:	Ne	
	Inicializace jako Naivní Bayes	Ano
	Maximální počet předků	1
	Skórovací funkce	Entropie
Vstupní počet atributů:	37	
Testovací nastavení:	Křížová validace, 10x	

---

klasifikováno: nemocný	**klasifikováno: zdravý**	
231	**32**	Reálně nemocný
213	**2120**	Reálně zdravý

Počet správných klasifikací:	**90,56** %	(2.351)
Počet chybných klasifikací:	**9,44** %	(245)
Průměrná plocha pod ROC křivkou:	**0,944**	
Falešně pozitivní hodnocení:	**12,2 %** pro nemocné jedince	
	9,1 % pro zdravé jedince	
Senzitivita:	**87,83 %**	
Specificita:	**90,87 %**	

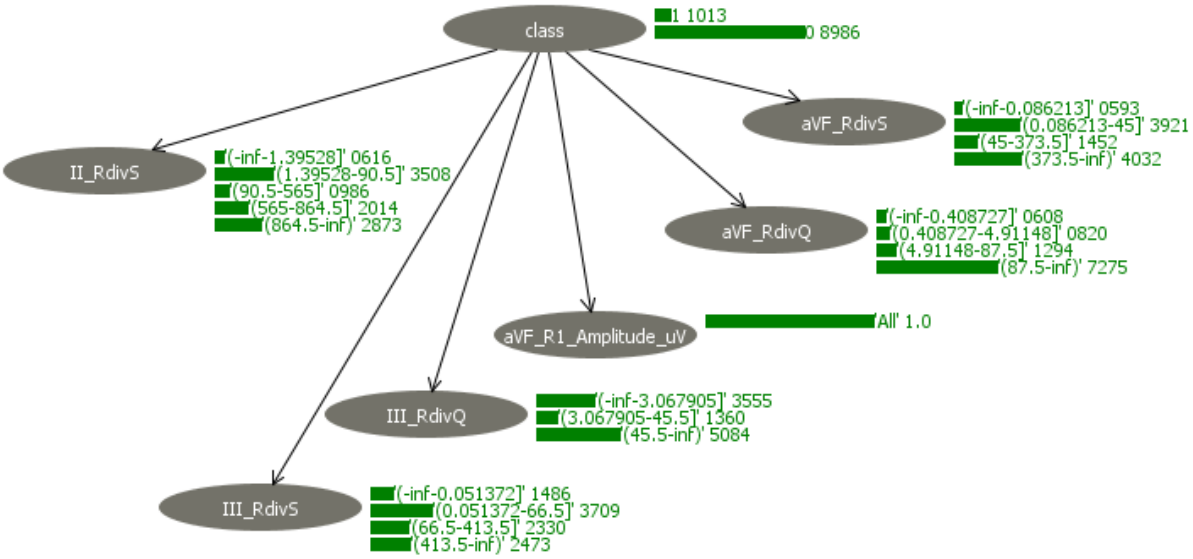


Obrázek 11 - ROC křivka, HillClimber na plných datech

ROC křivka má poměrně dobrou charakteristiku. Ideální stav je samozřejmě jednotkový skok, toho ale v praxi není možné dosáhnout. Přesto se křivka tomuto ideálu přibližuje. Je to vidět i na hodnotě obsahu plochy pod křivkou, která zabírá 94,4 % jednotkového obsahu.

Poněkud překvapující byl ale fakt, že algoritmus vyhodnotil jako nejefektivnější zapojení, kterým byl inicializován – tedy formu naivního Bayese. Jedná se o nejprimitivnější sestavení sítě, které předpokládá, že každá z vlastností má přímý vliv na rozhodující atribut a mezi sebou se hodnoty neovlivňují. Výstupní graf sítě jsem se rozhodl v tomto případě ořezat, protože se jedná o jednoúrovňovou stromovou strukturu. Zajímavější mi připadalo zobrazení včetně pravděpodobnostních tabulek.

Na uvedeném příkladu můžeme vidět, že některé z atributů mají více podskupin, podle kterých se rozhodují o přiřazení finálního rozhodnutí. Podskupiny se tvoří pomocí hledání shluků dat v daném parametru. Je tedy možné (*jak ukazuje Obrázek 12*), že atribut III_RdivS je rozdělen na 4 skupiny a naproti tomu atribut aVF_R1_Amplitude_uV má tak veliký rozptyl hodnot, že není možné provést rozdělení.



Obrázek 12 - Mapa BN - ořezaný Naivní Bayes

Poměrně zajímavé bylo poznání, že při složitějších zapojeních, která měla více úrovní a vzájemných závislostí, stoupala přesnost klasifikace. V jednom případě jsem se dostal až na hodnotu 94,38 %, ale spolu s přesností rostla i chybovost II. druhu. Nastala tak problémová situace, kdy posíláme nemocného pacienta domů. Přitom se také snižoval obsah plochy pod ROC křivkou. Na tomto prvním experimentu mohu demonstrovat vyšší důležitost snižování False-positive chyb než je zvyšování celkové správnosti klasifikace sítě. Samozřejmě vyšší správnost rozhodování je žádoucí, není však primární.

6.3.2 Repeated HillClimber, plný datový soubor

Ač se jedná o stejný algoritmus, můžeme dosáhnout lepších výstupů díky opakovanému spouštění z randomizovaných umístění. Vzniká tu vyšší šance na opuštění lokálního extrému. Tato šance se zvyšuje spolu s počtem opakování. Samozřejmě tím přímo úměrně narůstá i doba nutná ke zpracování.

Estimator:	Simple Estimator	Alfa = 0,1
Vyhledávací algoritmus:	Repeated HillClimber	
Použití AD stromu:	Ne	
Počet opakování:	10x	
	Inicializace jako Naivní Bayes	Ano
	Maximální počet předků	2
	Skórovací funkce	Entropie
	Seed (*pseudonáhodná funkce*)	2
Vstupní počet atributů:	37	
Testovací nastavení:	Křížová validace, 10x	

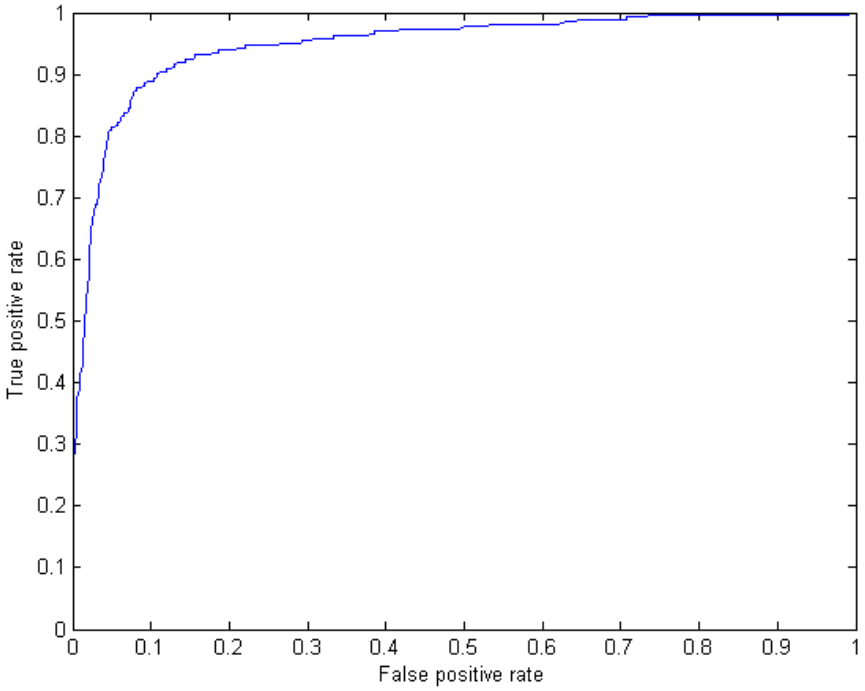
---

klasifikováno: nemocný	**klasifikováno: zdravý**	
230	**33**	Reálně nemocný
182	**2151**	Reálně zdravý

Počet správných klasifikací:	**91,72** %	(2.381)
Počet chybných klasifikací:	**8,28** %	(215)
Průměrná plocha pod ROC křivkou:	**0,948**	
Falešně pozitivní hodnocení:	**12,5 %** pro nemocné jedince	
	7,8 % pro zdravé jedince	
Senzitivita:	**87,45 %**	
Specificita:	**92,20 %**	

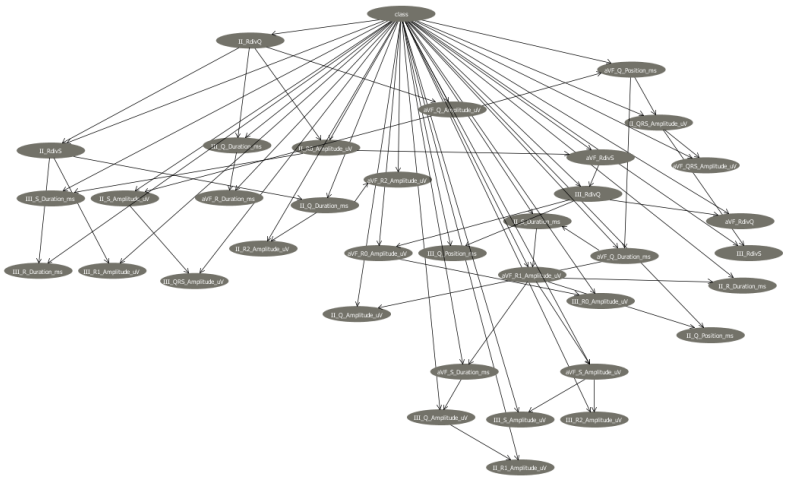
Z výsledků je patrné, že v porovnání s experimentem 6.3.1 je v tomto případě o něco horší falešně pozitivní ohodnocení nemocných jedinců, nicméně ubylo chybně klasifikovaných zdravých jedinců. Díky tomu také vzrostla celková správnost klasifikátoru.

Důležitým prvkem v tomto procesu byla tzv. hodnota *random seed*, která určuje způsob výpočtu pseudonáhodného generátoru. Při zvyšování počtu opakování jsem nepozoroval rozdíly ve výstupech a zaznamenal jsem zlepšení ROC křivky.

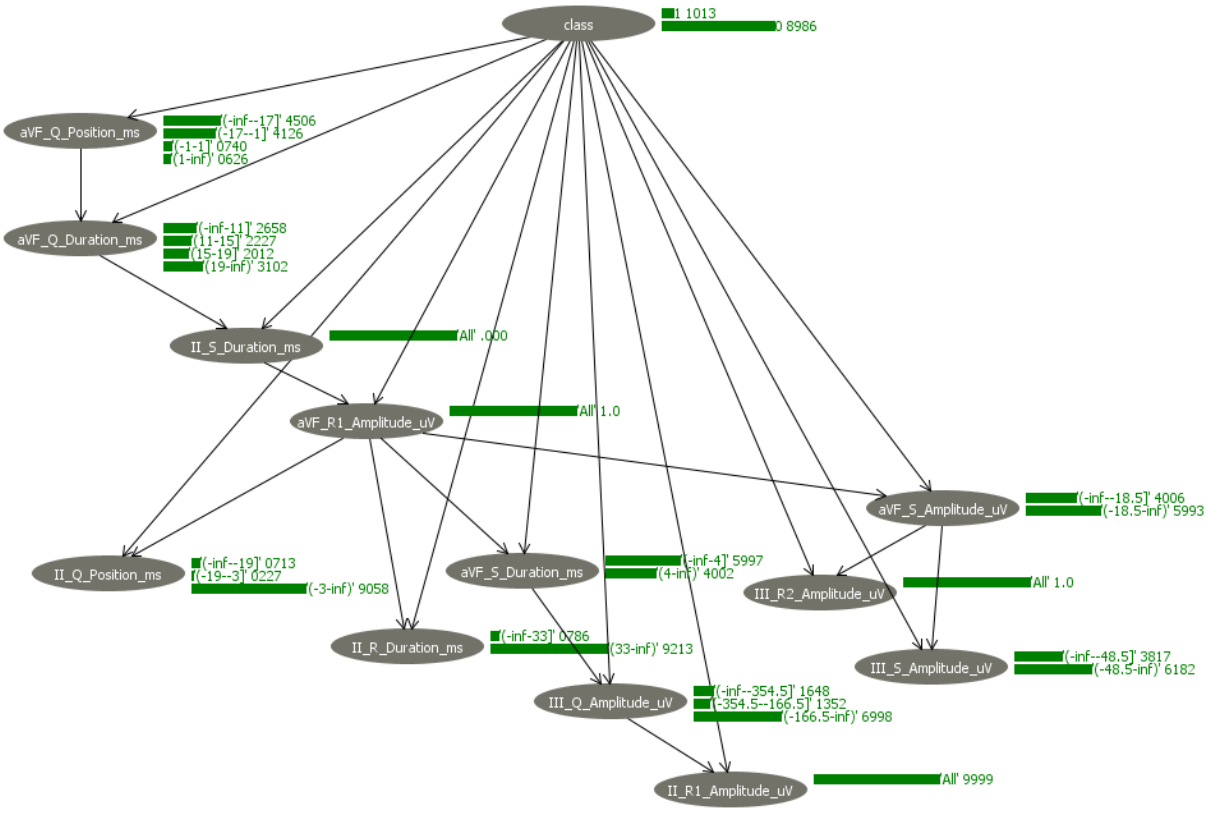


Obrázek 13 - ROC křivka, Repeated HillClimber a plný datový soubor

Pro ukázku přikládám celkovou mapu sítě (*Obrázek 14*). Je vidět, že složitost této sítě je velmi vysoká a algoritmus uplatnil všechny poskytnuté atributy, což nemusí být vždy žádoucí. Hloubka této sítě je v největším místě 7. Rozhodl jsem se opět udělat výsek této sítě, včetně pravděpodobnostních skupin, právě v místě největšího zanoření. Na obrázku můžeme vidět několik atributů, které opět nemají rozdělení do podskupin, protože algoritmus nebyl schopen tyto části od sebe odlišit. Otázkou však je, zdali jsou tyto atributy pro výpočet potřebné.



Obrázek 14 - Orientační mapa složitosti Bayesovské sítě



Obrázek 15 - Výsek BN, Repeate HillClimber na plných datech

6.3.3 LookAhead HillClimbing, plný datový soubor

Tento vylepšený horolezecký algoritmus prochází možné cesty dopředu a postupně tak tipuje nejvhodnější cestu. Díky velikosti předem vyhledávaných kroků dokáže přeskočit lokální extrém, ale se zvyšujícím se počtem kroků se exponenciálně zvyšuje výpočetní náročnost. Algoritmus totiž prochází všechny možnosti a hledá nejlevnější cestu.

Estimator:	Simple Estimator	Alfa = 0,1
Vyhledávací algoritmus:	LookAhead HillClimber	
Použití AD stromu:	Ne	
	Inicializace jako Naivní Bayes	Ano
	Maximální počet předků	2
	Skórovací funkce	Entropie
	Počet dopředných kroků	1
Vstupní počet atributů:	37	
Testovací nastavení:	Křížová validace, 10x	

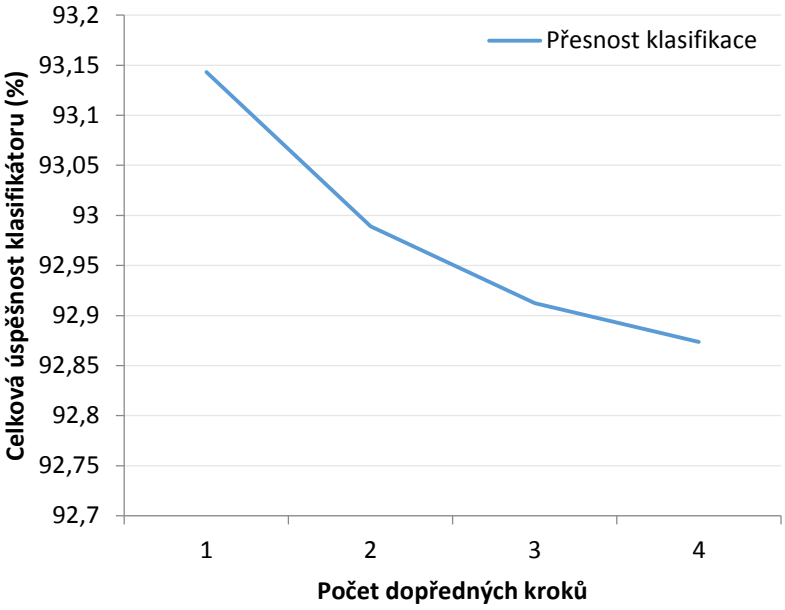
---

klasifikováno: nemocný	**klasifikováno: zdravý**	
220	**43**	Reálně nemocný
135	**2198**	Reálně zdravý

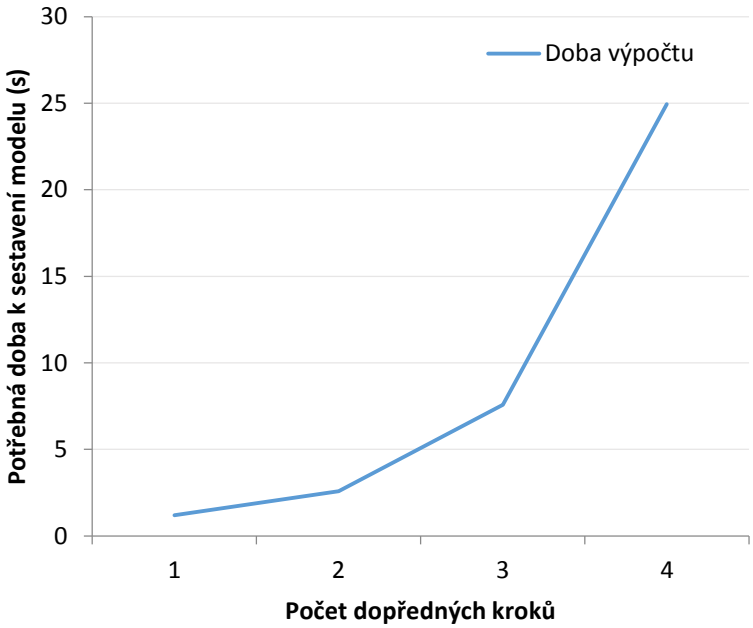
Počet správných klasifikací:	**93,14** %	(2.418)
Počet chybných klasifikací:	**6,86** %	(178)
Průměrná plocha pod ROC křivkou:	**0,954**	
Falešně pozitivní hodnocení:	**16,3 %** pro nemocné jedince	
	5,8 % pro zdravé jedince	
Senzitivita:	**83,65 %**	
Specificita:	**94,21 %**	

Výstup této sítě je prozatím nejlepší v celkové správnosti hodnocení. Je zde ale vysoká míra chybovosti nemocných jedinců a tedy i nízká senzitivita. Proto není tento algoritmus vhodný pro toto použití i přes zatím nejlepší hodnocení klasifikace.

Jako velmi zajímavý závěr hodnotím rostoucí nepřesnost sítě spolu s rostoucím počtem předem vyhledávaných kroků, ač bych čekal opak. Rostoucí změny jsem detekoval pouze na chybném ohodnocení reálně zdravých pacientů, což se ovšem promítá i do celkové přesnosti klasifikátoru (*Obrázek 16*). Vývoj výpočetní náročnosti je zobrazen níže (*Obrázek 17*).

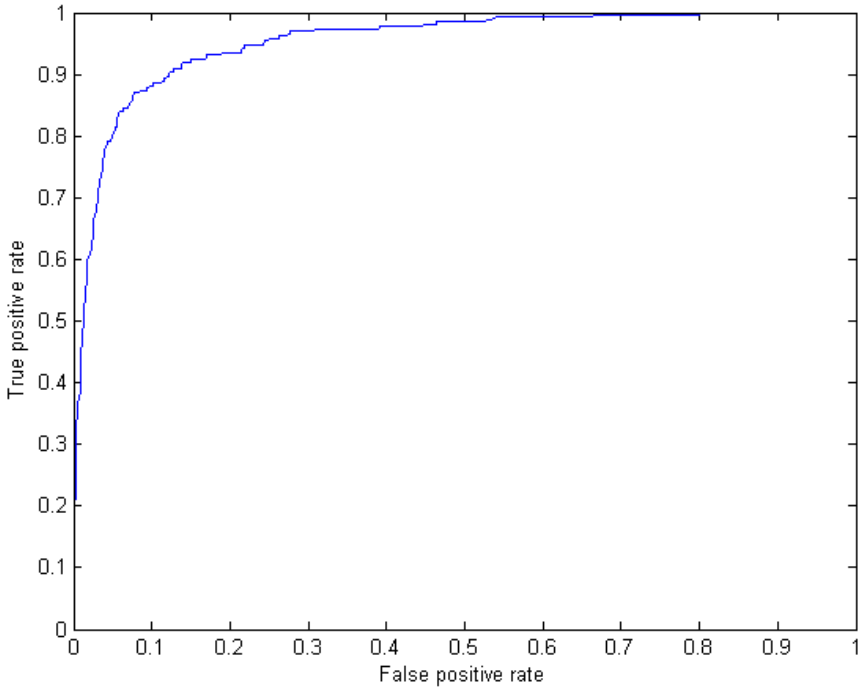


Obrázek 16 - Závislost přesnosti klasifikace na počtu dopředných kroků

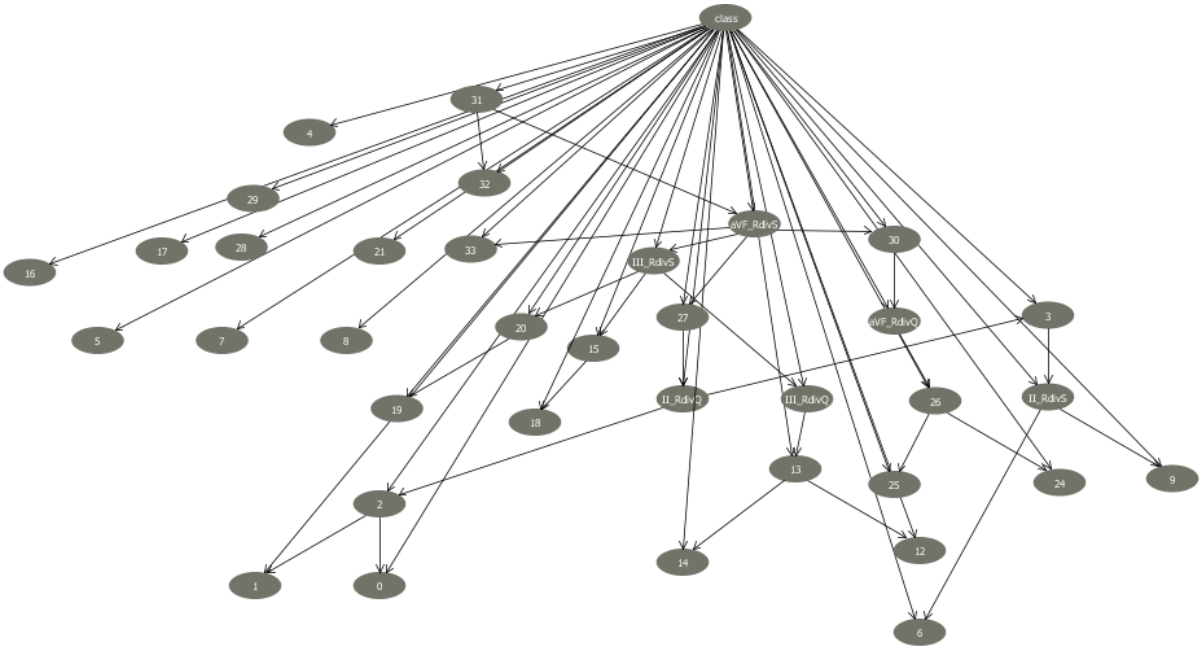


Obrázek 17 - Graf vzrůstající doby, potřebné k sestavení modelu v závislosti na počtu dopředných kroků

Síť je opět vysoce složitá s hloubkou zanoření 8. Vložený graf je tedy pouze orientační (*kompletní mapa sítě ve formátu XML je k dispozici na CD, pod názvem network.xml*).



Obrázek 18 - Vizualizace ROC křivky pro LAGD HillClimber, plný datový soubor



Obrázek 19 - Orientační mapa složitosti BN, LAGD HillClimber, plný datový soubor

6.3.4 K2 algoritmus, plný datový soubor

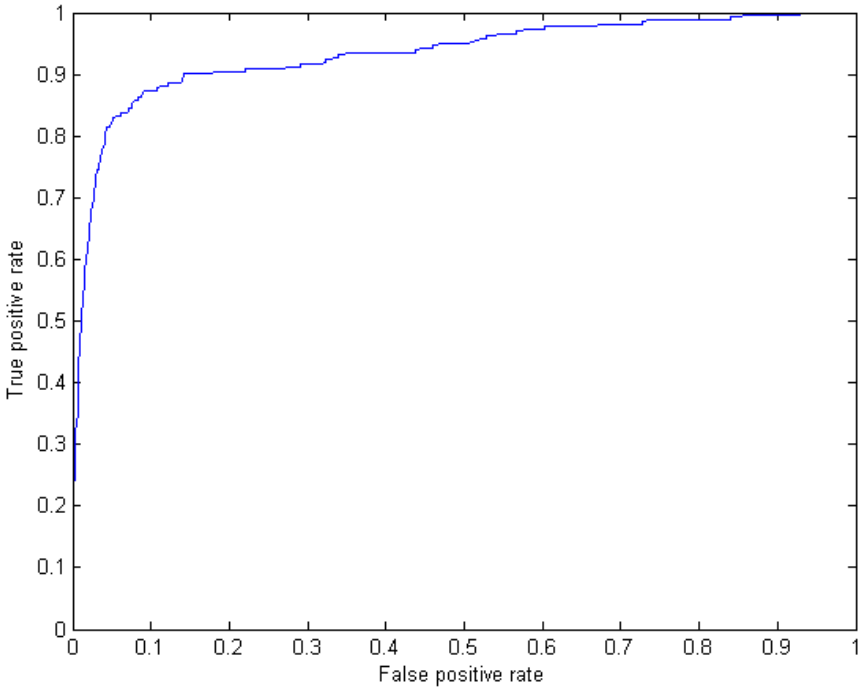
Estimator:	Simple Estimator	Alfa = 0,1
Vyhledávací algoritmus:	K2	
Použití AD stromu:	Ne	
	Inicializace jako Naivní Bayes	Ne
	Maximální počet předků	1
	Náhodné řazení	Ne
	Skórovací funkce	Bayesovské skóre
Vstupní počet atributů:	37	
Testovací nastavení:	Křížová validace, 10x	

---

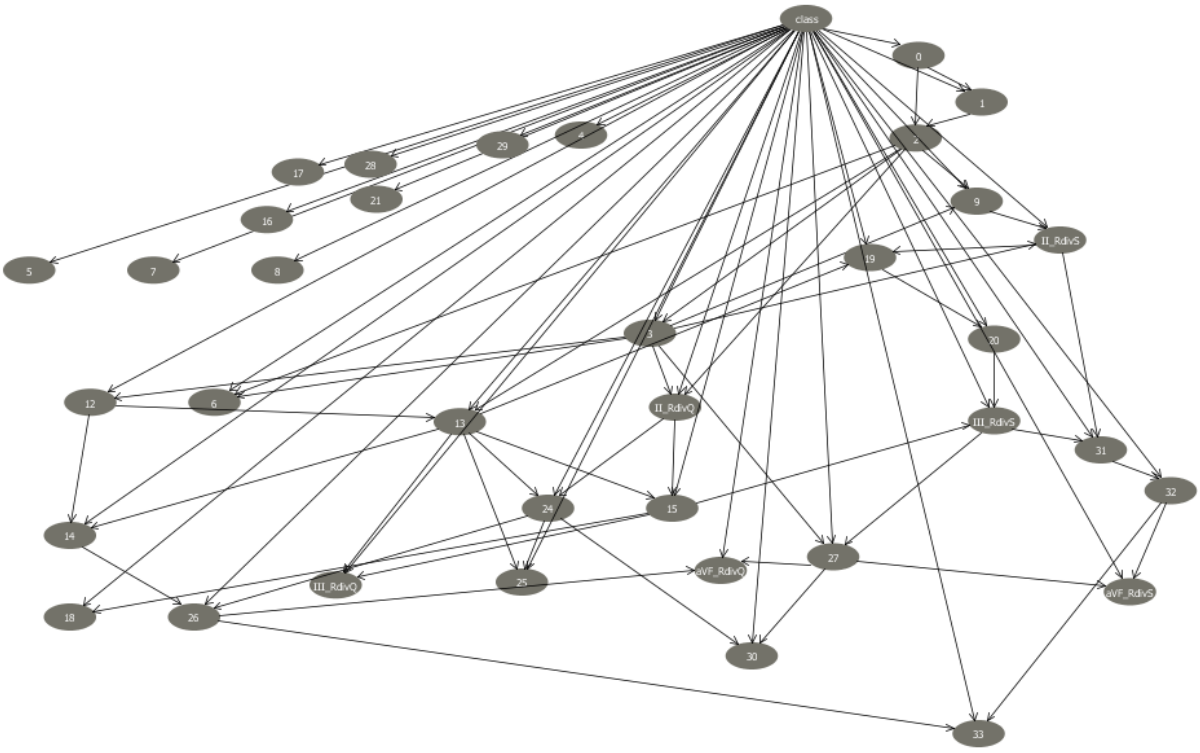
klasifikováno: nemocný	klasifikováno: zdravý	
198	**65**	Reálně nemocný
78	**2255**	Reálně zdravý

Počet správných klasifikací:	**94,49** %	(2.453)
Počet chybných klasifikací:	**5,51** %	(143)
Průměrná plocha pod ROC křivkou:	**0,934**	
Falešně pozitivní hodnocení:	**24,7 %** pro nemocné jedince	
	3,3 % pro zdravé jedince	
Senzitivita:	**75,29 %**	
Specificita:	**96,66 %**	

Síť vytvořená algoritmem K2 má velmi vysokou celkovou přesnost klasifikace, ale dosahuje jí díky nízké chybovosti u zdravých jedinců. Ze všech měření je chybovost prozatím nejnižší. Jak jsem již poznamenal v minulých kapitolách, tato síť není pro dané použití vhodná. Výhodou je velmi rychlé zpracování sítě i s vyšší složitostí. Lépe však vycházela síť sestavená z naivního Bayese, ale tu již jako výsledky prezentuji v experimentu 6.3.1, proto jsem se rozhodl zahrnout druhou nejlepší strukturu. Mapa sítě je vzhledem k velmi vysoké složitosti opět orientační. Oproti experimentu 0 je patrná vyšší provázanost mezi jednotlivými atributy.



Obrázek 20 - Vizualizace ROC křivky, K2 na plném datovém setu



Obrázek 21 - Orientační mapa BN, K2 na plném datovém setu

6.3.5 Simulated Annealing, plný datový soubor

Simulované žíhání zabralo velmi dlouhou výpočetní dobu. Je to dáno především vysokým počtem prvků, přičemž složitost výpočtu roste exponenciálně spolu s počtem příznaků. Pro urychlení výpočtu jsem se rozhodl u tohoto experimentu změnit způsob měření, data jsem rozdělil na 50% testovací / trénovací množiny a výpočty provedl pouze jednou. I tak zabral výpočet téměř 4 hodiny a byl poměrně náročný na operační paměť (*vyžadoval přibližně 21 GB RAM*).

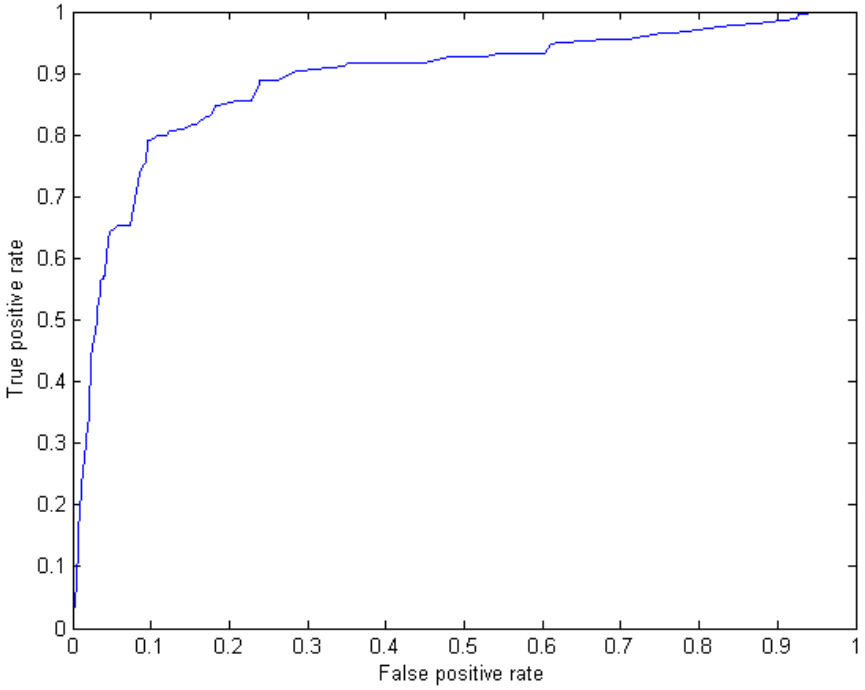
Estimator:	Simple Estimator	Alfa = 0,1
Vyhledávací algoritmus:	Simulated Annealling	
Použití AD-stromu:	Ne	
	Počáteční teplota	10,0
	Delta teploty	0,999
	Počet běhů algoritmu	10.000
	Skórovací funkce	Entropie
Vstupní počet atributů:	37	
Testovací nastavení:	Rozdělení 50% (*testovací/trénovací množina*)	

---

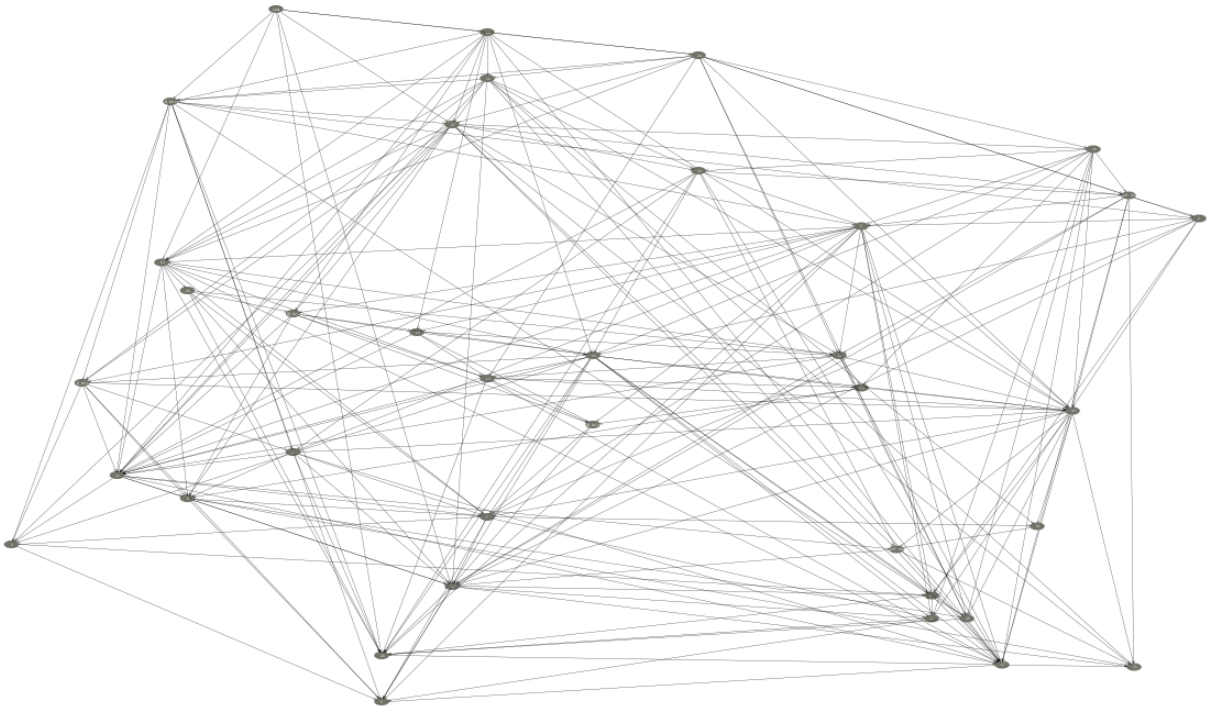
klasifikováno: nemocný	**klasifikováno: zdravý**	
95	**35**	Reálně nemocný
36	**1132**	Reálně zdravý

Počet správných klasifikací:	**94,53** %	(1.227)
Počet chybných klasifikací:	**5,47** %	(71)
Průměrná plocha pod ROC křivkou:	**0,951**	
Falešně pozitivní hodnocení:	**26,9 %** pro nemocné jedince	
	3,1 % pro zdravé jedince	
Senzitivita:	**73,08 %**	
Specificita:	**96,92 %**	

Prezentované výsledky mohou být v tomto experimentu zkresleny, protože se neprováděla křížová validace, ale jedno měření. Nicméně výsledky celkové správnosti ohodnocení a falešných poplachů u zdravých jedinců jsou z plného datového souboru jednoznačně nejlepší. Naproti tomu požadovaná hodnota chybně označených nemocných jedinců je z měření nejhorší. Proto síť s daným nastavením není vhodná k použití. Vizualizace sítě je opět orientační a v této práci se jedná o nejobsáhlejší mapu. Je zřejmé, že s vyšší složitostí nemusíme nutně dosáhnout lepších výsledků.



Obrázek 22 - Vizualizace ROC křivky, Simulované žíhání na plném datovém setu



Obrázek 23 - Vizualizace složitosti sítě vytvořené simulovaným žíháním

Z grafu (*Obrázek 23*) je patrná velmi vysoká složitost zapojení sítě. V takovém případě by bylo vhodné při generování sítě využít paralelní výpočty pro jednotlivé teploty nebo cykly. Nabízí se tak možnost pro využití distribuovaných výpočetních gridů nebo cloudových služeb. Ty pro danou úlohu disponují vhodnou výpočetní kapacitou.

6.3.6 Naivní Bayes na plném datovém setu

Pro toto zpracování jsem opět použil program WEKA. Ponechal jsem křížovou validaci a nastavil jsem parametr *useSupervisedDiscretization*, který nám určuje, zdali budou převedeny číselné atributy na nominální jednotky.

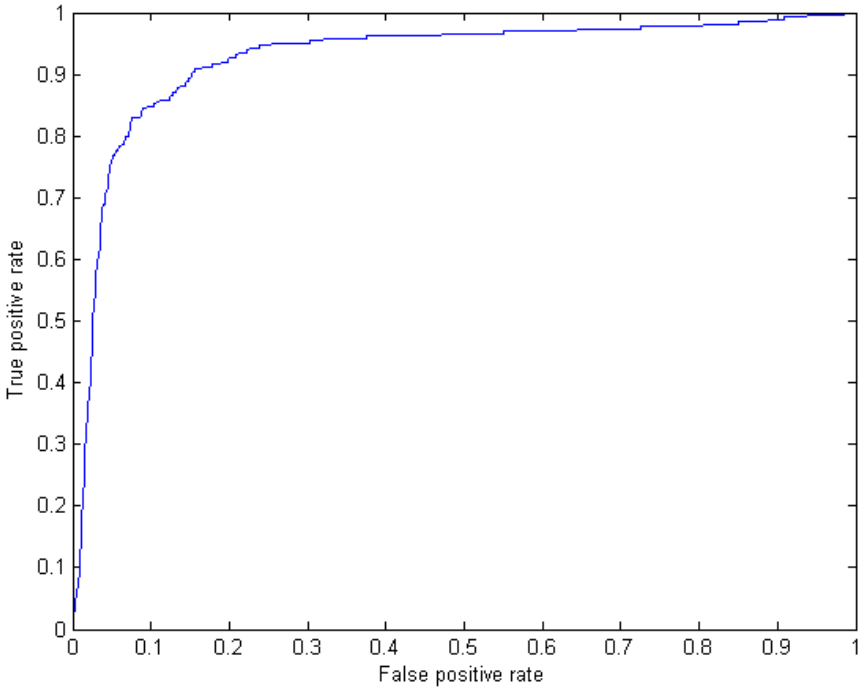
Vstupní počet atributů:	37
useSupervisedDiscretization	Ano
Testovací nastavení:	Křížová validace, 10x

Výstupy

klasifikováno: nemocný	**klasifikováno: zdravý**	
229	**34**	Reálně nemocný
283	**2050**	Reálně zdravý

Počet správných klasifikací:	**87,79** %	(2.279)
Počet chybných klasifikací:	**12,21** %	(317)
Průměrná plocha pod ROC křivkou:	**0,944**	
Falešně pozitivní hodnocení:	**12,1 %** pro nemocné jedince	
	12,9 % pro zdravé jedince	
Senzitivita:	**87,07 %**	
Specificita:	**87,87 %**	

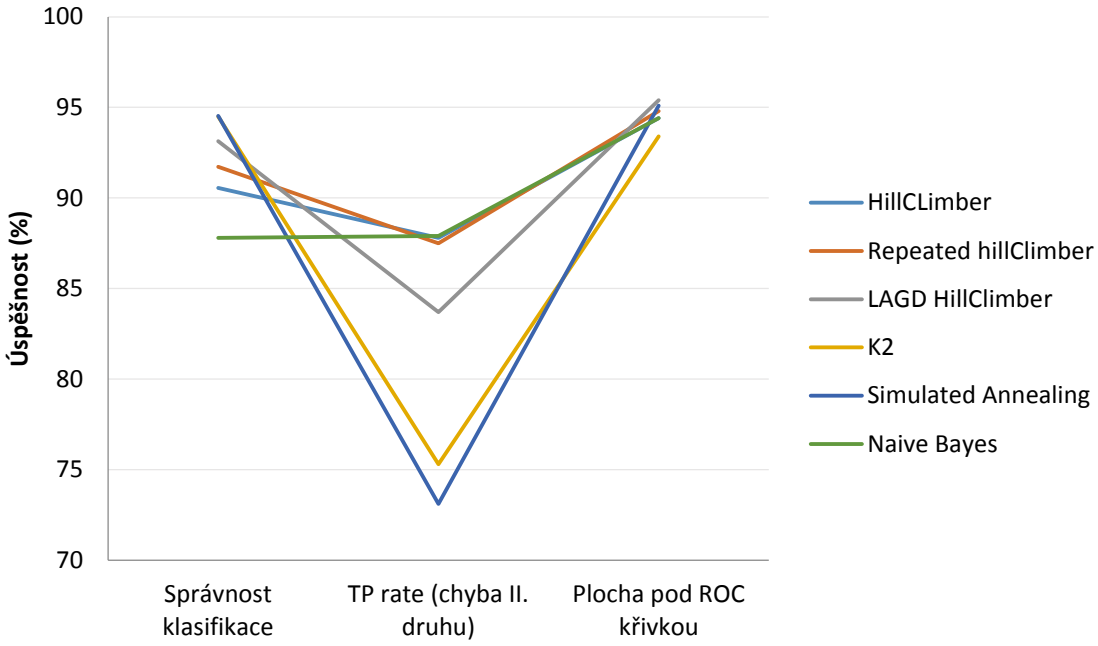
Jedná se o doplňkový experiment, u kterého je zajímavé, že ač se jedná o totožné nastavení, jako v případě 6.3.1, tak vychází rozdílné výsledky. Je to dáno procedurou nastavení pravděpodobnostních tabulek sítě.



Obrázek 24 - Vizualizace ROC křivky, Naivní Bayes na plném datovém setu

6.3.7 Zhodnocení úspěšnosti jednotlivých sítí

Pro vyhodnocení úspěšností jednotlivých metod výpočtu jsem využil parametry procentuální celkové správnosti klasifikace, velikost plochy pod ROC křivkou a hodnotu True-Positive pro zdravé jedince. True-Positive veličina nám vyčísluje úspěšnost správné klasifikace zdravých jedinců. Tyto hodnoty jsem zvolil proto, že se je snažím v grafu maximalizovat.



Obrázek 25 - Graf úspěšnosti sestavených sítí

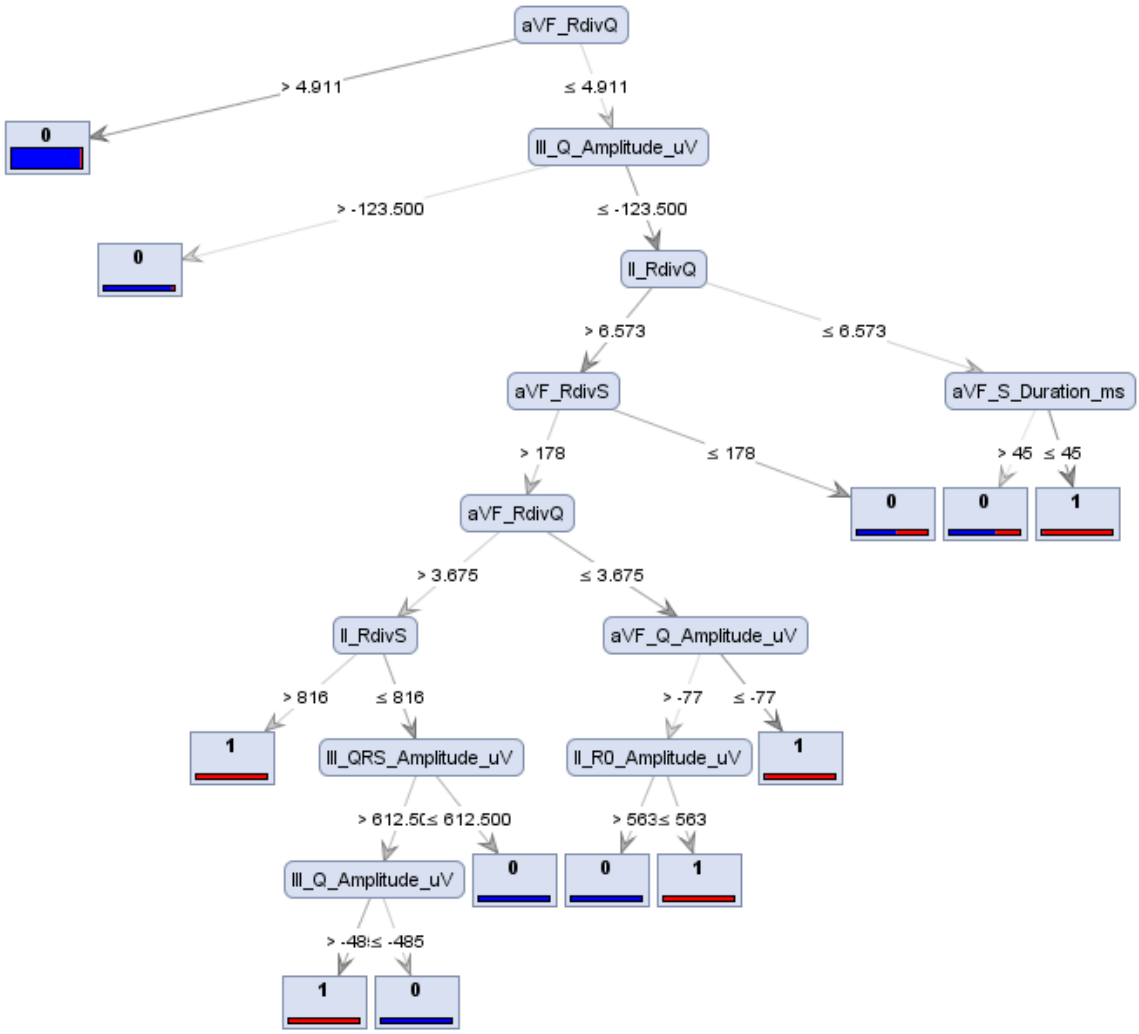
Jak z grafu (*Obrázek 25*) vyplývá, jako nejvhodnější sítě se jeví 6.3.1 a 0, tedy metody HillClimber a v prvním případě prakticky čistý naivní Bayes. Pro tak velký datový soubor je vzhledem ke své rychlosti i výhodnější.

Pro zaměření této práce je primární TP-rate, který právě u naivního Bayese vychází nejlépe. V následující kapitole se pokusím snížit počet proměnných a vynutit tak zlepšení klasifikace Bayesovských sítí.

6.4 Zpracování s vylepšeným předzpracováním dat

Vzhledem k tomu, jak vypadají a jak dlouho se počítaly předchozí experimenty, rozhodl jsem se provést změny v části předzpracování dat. Cílem bylo samozřejmě dosáhnut vyšší přesnosti rozhodování sítě.

Základním problémem je v tomto případě vysoký počet atributů, který síť značně komplikuje. Při hlubším pohledu na většinu dat můžeme konstatovat, že jsou jejich hodnoty korelované. Proto jsem se rozhodl sestavit rozhodovací strom, který by mi mohl pomoci v rozhodnutí, které atributy nesou větší informaci a které by bylo možné vypustit. Vypuštěním některých atributů sice můžeme přijít o některou ze závislostí, ale za to můžeme značně urychlit zpracování dat.



Obrázek 26 - Rozhodovací strom sestavený z kompletních dat

Použitý rozhodovací strom typu AD je schopen s daty pracovat tak, že vyhodnotí jejich entropii a z té určuje potřebnost příznaků. Do kořene vloží hodnotu s nejnižší entropií, tj. data, která mají největší míru neuspořádanosti a rozdělí veličiny na 2 skupiny. Dále postupuje, dokud se mu nepodaří separovat jednotlivé výstupní třídy. Pro rozdělení skupiny je možné využit například metody shlukování. Příznaky, které mají entropii vysokou a není tak jednoduché najít v datech podskupiny, se do vyhodnocení nepoužijí.

Díky této proceduře jsem dostal 9 příznaků, které dokáží problém s dostatečně vyhovující přesností determinovat.

6.4.1 HillClimber, omezený datový soubor

Jako první přichází metoda HillClimber, kterou jsem se rozhodl vybrat jinou než naivní Bayes. Naivní algoritmus budu na závěr testovat samostatně (*6.4.7*), nyní je mým cílem dostat složitější strukturu sítě.

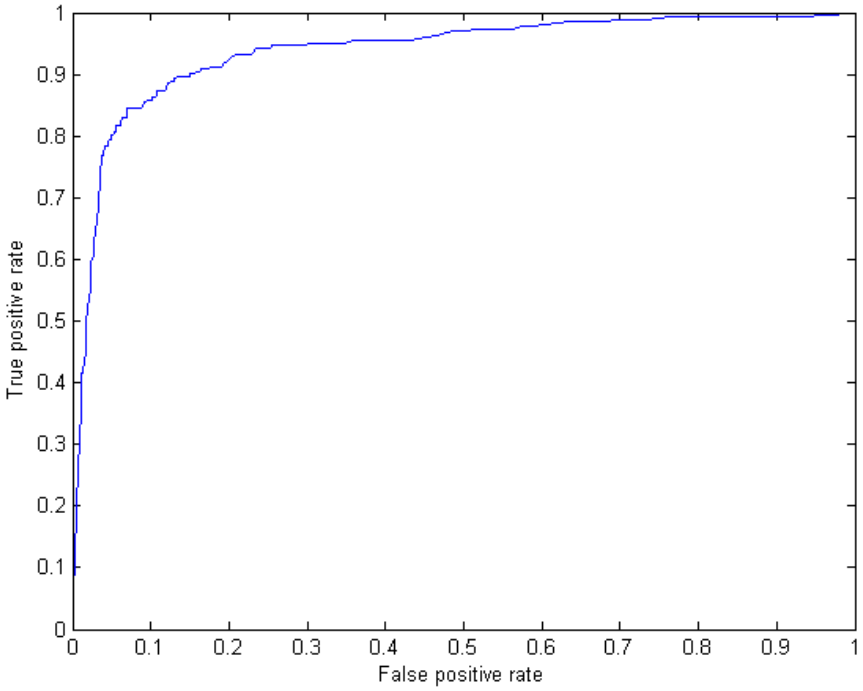
Estimator:	Simple Estimator	Alfa = 0,1
Vyhledávací algoritmus:	HillClimber	
Použití AD stromu:	Ne	
	Inicializace jako Naivní Bayes	Ano
	Maximální počet předků	2
	Skórovací funkce	Bayes
Vstupní počet atributů:	9	
Testovací nastavení:	Křížová validace, 10x	

---

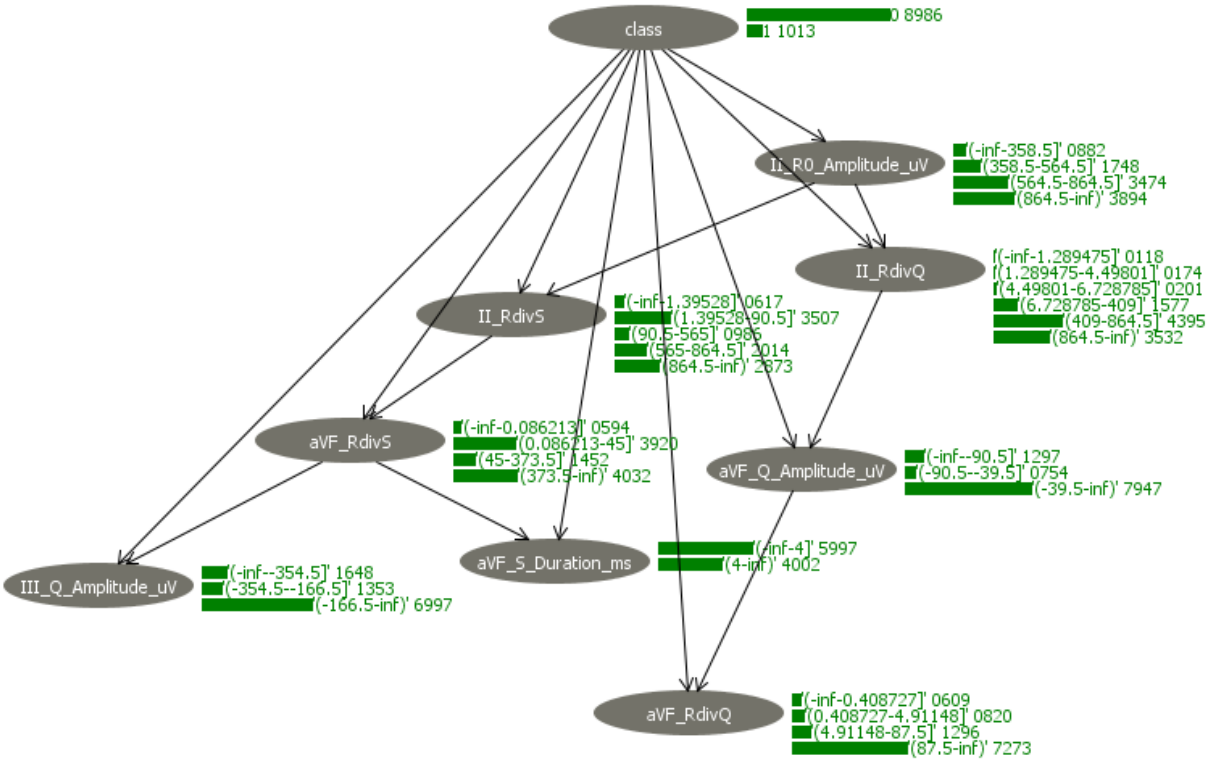
klasifikováno: nemocný	**klasifikováno: zdravý**	
207	**56**	Reálně nemocný
93	**2240**	Reálně zdravý

Počet správných klasifikací:	**94,26** %	(2.447)
Počet chybných klasifikací:	**5,74** %	(149)
Průměrná plocha pod ROC křivkou:	**0,94**	
Falešně pozitivní hodnocení:	**21,3 %** pro nemocné jedince	
	4,0 % pro zdravé jedince	
Senzitivita:	**78,71 %**	
Specificita:	**96,01 %**	

Můžeme vidět, že algoritmus HillClimber podává velmi dobré výsledky pro celkovou správnost sítě. Důvodem je velmi nízká chybovost u zdravých jedinců, která je nejlepší ze všech použitých modifikací HillClimber experimentů na plném datovém setu. Největší nevýhodou je ale vysoká chybovost u nemocných jedinců, která použití sítě znemožňuje.



Obrázek 27 - Vizualizace ROC křivky, HillClimber na částečném datovém setu



Obrázek 28 - Vizualizace mapy sítě

Jako zajímavou vlastnost bych zde zmínil, že se algoritmus při sestavování sítě rozhodl vypustit atribut III_QRS_Amplitude_uV.

6.4.2 Repeated HillClimber, omezený datový soubor

Od opakovaného HillClimberu opět očekávám zvýšení přesnosti. Parametrem je seed, který volím experimentálně. Opět vyřazuji naivního Bayese.

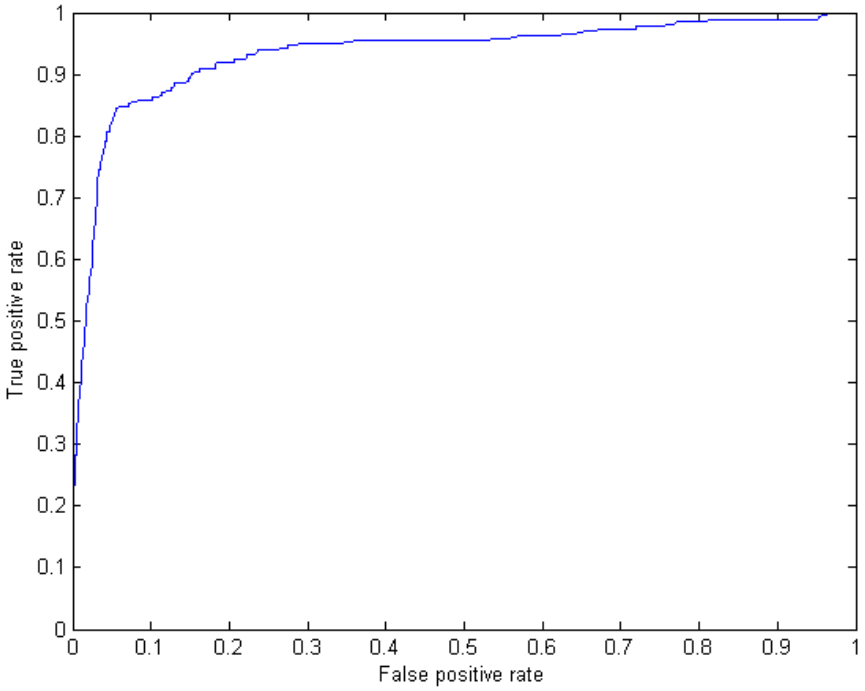
Estimator:	Simple Estimator	Alfa = 0,1
Vyhledávací algoritmus:	Repeated HillClimber	
Použití AD stromu:	Ne	
Počet opakování:	10x	
	Inicializace jako Naivní Bayes	Ano
	Maximální počet předků	2
	Skórovací funkce	Bayes
	Seed (*pseudonáhodná funkce*)	3
Vstupní počet atributů:	9	
Testovací nastavení:	Křížová validace, 10x	

---

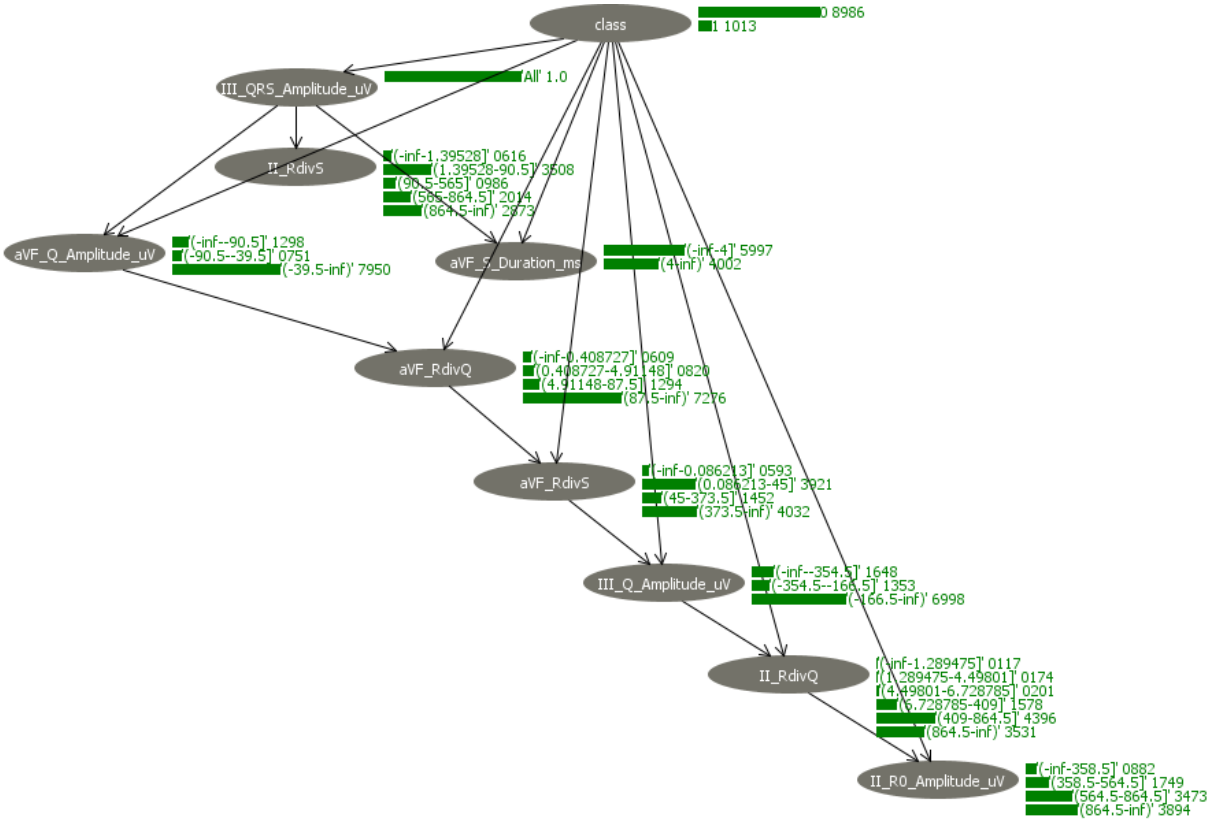
klasifikováno: nemocný	**klasifikováno: zdravý**	
223	**50**	Reálně nemocný
110	**2223**	Reálně zdravý

Počet správných klasifikací:	**93,84** %	(2.436)
Počet chybných klasifikací:	**6,16** %	(160)
Průměrná plocha pod ROC křivkou:	**0,940**	
Falešně pozitivní hodnocení:	**19,0 %** pro nemocné jedince	
	4,7 % pro zdravé jedince	
Senzitivita:	**81,68 %**	
Specificita:	**95,29 %**	

Experiment s opakovaným horolezeckým algoritmem dopadl velmi podobně jako předchozí (*6.4.1*). Došlo ke zlepšení celkové úspěšnosti klasifikace, nicméně opět vzrostla nežádoucí chyba u nemocných jedinců. Nesporným plusem (*vzhledem k 6.3.2*) je značně rychlejší zpracování a nastavení pravděpodobnostních tabulek.



Obrázek 29 - Vizualizace ROC křivky, Repeated HillClimber a omezený datový soubor



Obrázek 30 - Vizualizace mapy sítě

6.4.3 LookAhead HillClimbing, omezený datový soubor

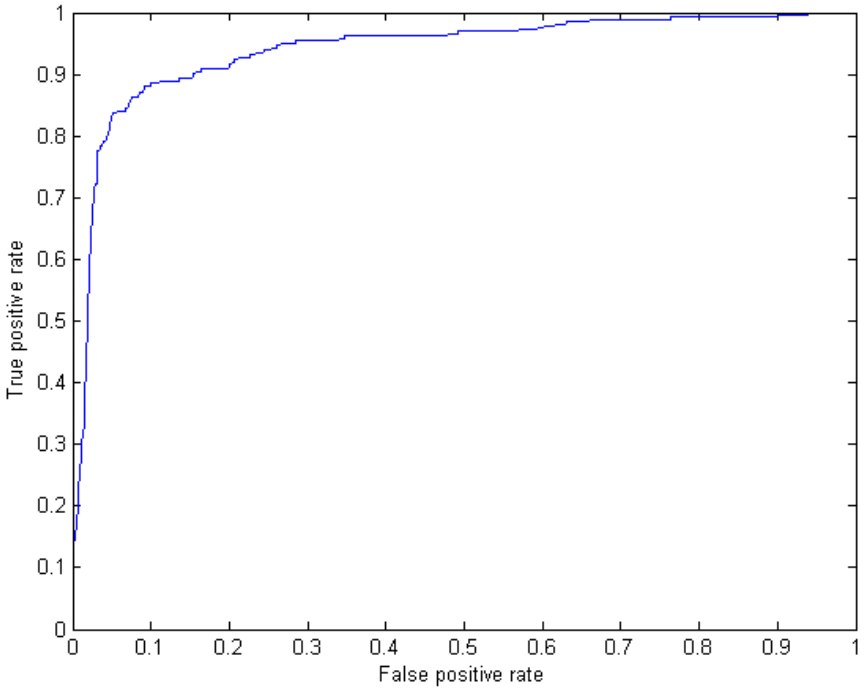
Estimator:	Simple Estimator	Alfa = 0,1
Vyhledávací algoritmus:	LookAhead HillClimber	
Použití AD stromu:	Ne	
	Inicializace jako Naivní Bayes	Ano
	Maximální počet předků	2
	Skórovací funkce	Bayes
	Počet dopředných kroků	1
Vstupní počet atributů:	9	
Testovací nastavení:	Křížová validace, 10x	

---

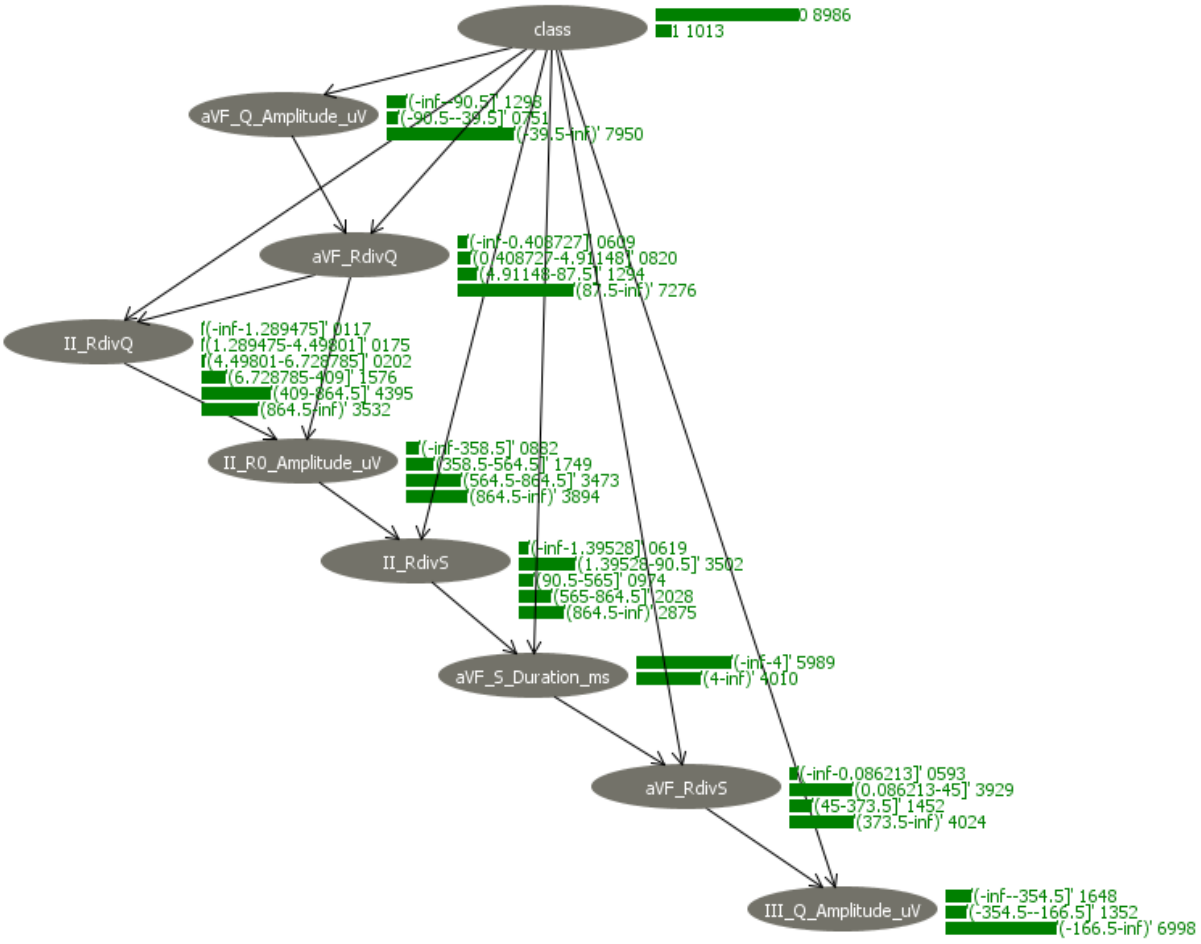
klasifikováno: nemocný	klasifikováno: zdravý	
209	54	Reálně nemocný
94	2239	Reálně zdravý

Počet správných klasifikací:	**94,30** %	(2.418)
Počet chybných klasifikací:	**5,70** %	(178)
Průměrná plocha pod ROC křivkou:	**0,941**	
Falešně pozitivní hodnocení:	**20,5 %** pro nemocné jedince	
	4,0 % pro zdravé jedince	
Senzitivita:	**79,47 %**	
Specificita:	**95,97 %**	

Při porovnání s původním LAGD HillClimbing algoritmem opět pozoruji zlepšení celkové klasifikace, ale při podobném nastavení s experimentem 0 má výstup sítě horší chybu ohodnocení nemocných pacientů. Ta se tentokrát dostává nad 20 %, síť je tedy nevhodná pro použití, ač má velmi dobrou celkovou úspěšnost. Zajímavostí této sítě je vypuštění atributu III_QRS_Amplitude_uV, podobně jako v experimentu 6.4.1.



Obrázek 31 - Vizualizace ROC křivky, LAGD HillClimber, omezený datový soubor



Obrázek 32 - Vizualizace mapy sítě

6.4.4 K2, omezený datový soubor

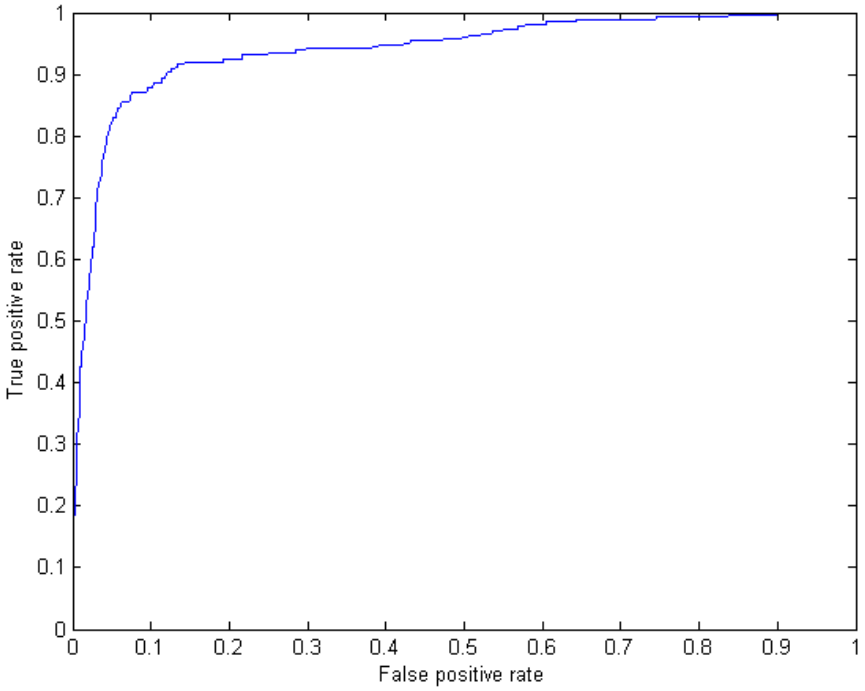
Estimator:	Simple Estimator	Alfa = 0,1
Vyhledávací algoritmus:	K2	
Použití AD stromu:	Ne	
	Inicializace jako Naivní Bayes	Ne
	Maximální počet předků	3
	Náhodné řazení	Ano
	Skórovací funkce	Bayesovské skóre
Vstupní počet atributů:	9	
Testovací nastavení:	Křížová validace, 10x	

---

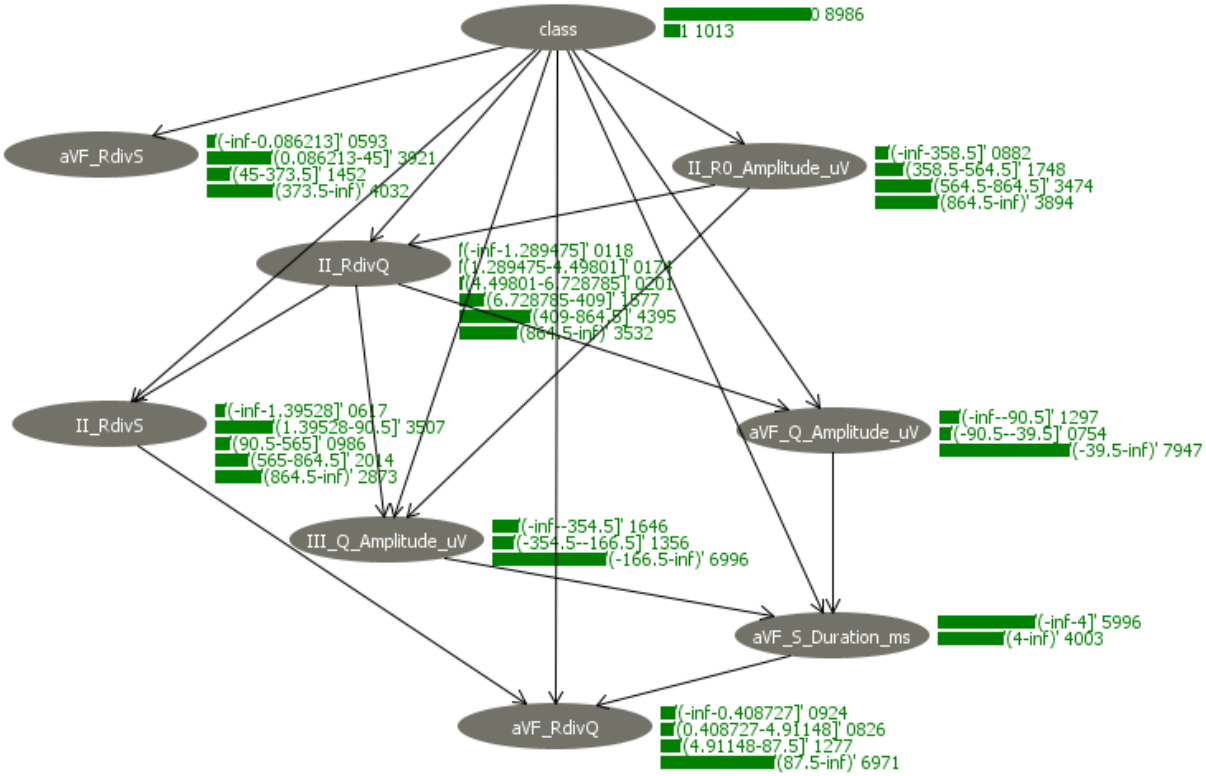
klasifikováno: nemocný	klasifikováno: zdravý	
215	48	Reálně nemocný
112	2221	Reálně zdravý

Počet správných klasifikací:	**93,84** %	(2.453)
Počet chybných klasifikací:	**6,16** %	(143)
Průměrná plocha pod ROC křivkou:	**0,941**	
Falešně pozitivní hodnocení:	**18,3 %** pro nemocné jedince	
	4,8 % pro zdravé jedince	
Senzitivita:	**81,75 %**	
Specificita:	**95,20 %**	

Při vytváření struktury sítě algoritmem K2 se omezení datového souboru osvědčilo. Došlo sice k poklesu celkové přesnosti, ale kleslo i falešně pozitivní ohodnocení nemocných jedinců o více než 6%. V tomto případě bych tedy síť zvažoval k implementaci, velkým plusem je razantní zvýšení rychlosti zpracování oproti práci nad kompletními daty. Výsledná síť je poměrně spletitá, opět došlo k vynechání atributu III_QRS_Amplitude_uV.



Obrázek 33 - Vizualizace ROC křivky, K2 a omezený datový soubor



Obrázek 34 - Vizualizace mapy sítě

6.4.5 Simulated Annealing, omezený datový soubor

U tohoto experimentu primárně očekávám velké snížení nároků na výpočet, a to jak paměťových, tak časových.

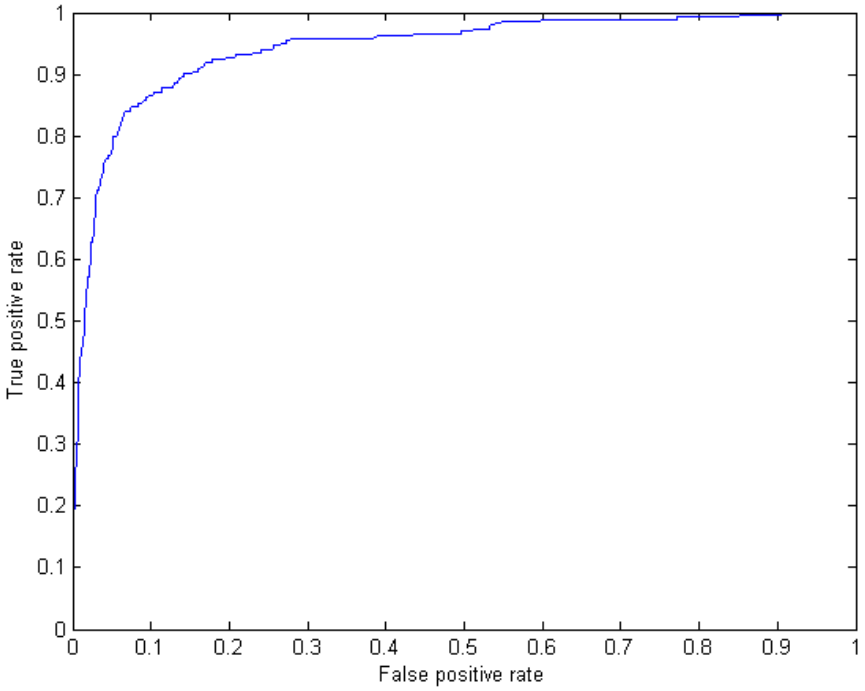
Estimator:	Simple Estimator	Alfa = 0,1
Vyhledávací algoritmus:	Simulated Annealling	
Použití AD-stromu:	Ne	
	Počáteční teplota	10,0
	Delta teploty	0,999
	Počet běhů algoritmu	10.000
	Skórovací funkce	Bayes
Vstupní počet atributů:	9	
Testovací nastavení:	Křížová validace, 10x	

---

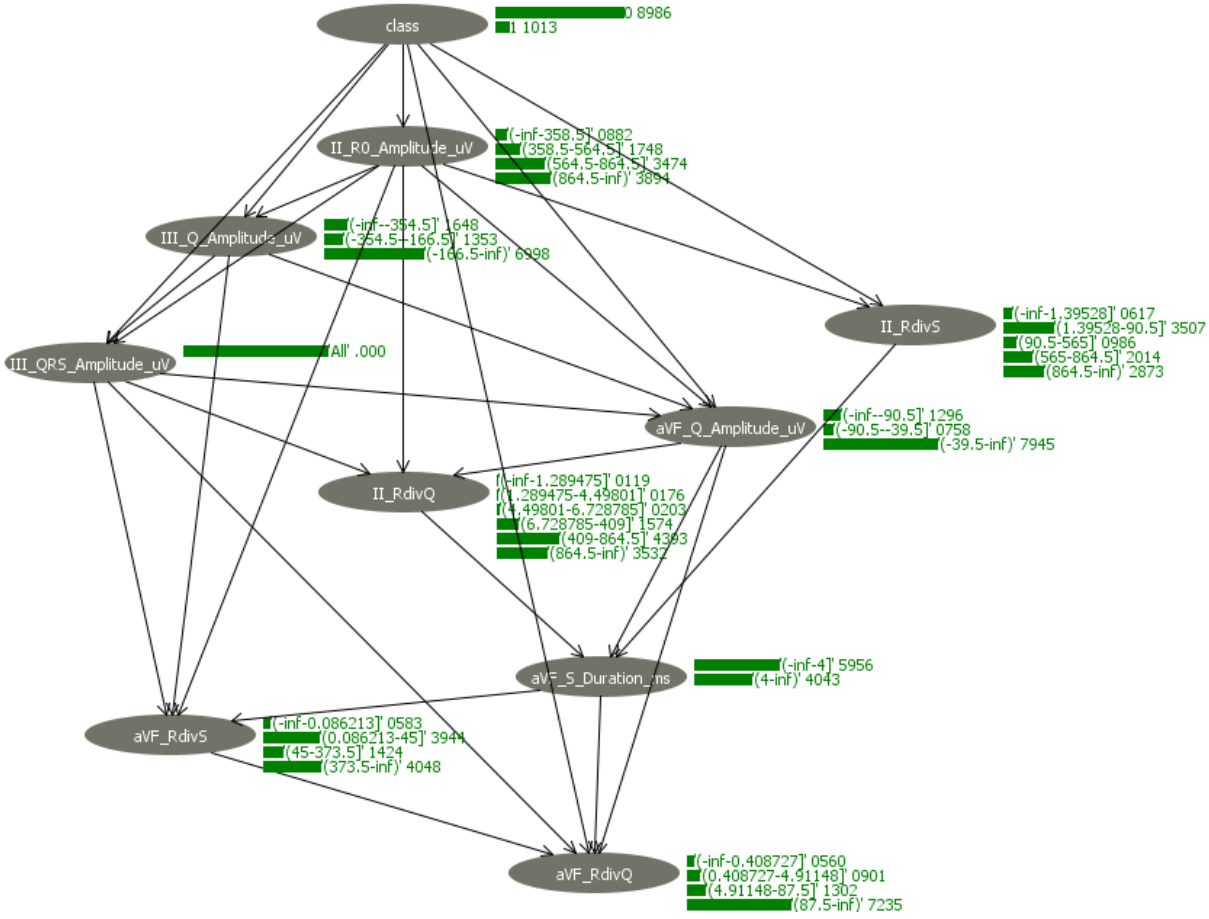
klasifikováno: nemocný	klasifikováno: zdravý	
201	**62**	Reálně nemocný
103	**2230**	Reálně zdravý

Počet správných klasifikací:	**93,64** %	(2.431)
Počet chybných klasifikací:	**6,36** %	(71)
Průměrná plocha pod ROC křivkou:	**0,943**	
Falešně pozitivní hodnocení:	**23,6 %** pro nemocné jedince	
	4,4 % pro zdravé jedince	
Senzitivita:	**76,43 %**	
Specificita:	**95,59 %**	

Dle očekávání, došlo ke zrychlení algoritmu, výpočet nyní zabral řádově 80 sekund. Bylo proto možné použití křížové validace. Snížení počtu vstupních proměnných mělo znovu pozitivní vliv na počet falešně pozitivních ohodnocení nemocných jedinců. Spolu s tím však klesla celková úspěšnost klasifikátoru přibližně o 1%.



Obrázek 35 - Vizualizace ROC křivky, Simulované žíhání na omezeném setu dat



Obrázek 36 - Vizualizace mapy sítě

6.4.6 Simulated Annealing (2), omezený datový soubor

Vzhledem k tomu, že jsem u metody simulovaného žíhání dosáhl dvou zajímavých výsledků, rozhodl jsem se zobrazit oba. SA (2) má vyšší senzitivitu, ale spolu s tím se částečně snižuje hodnota specificity. U klasifikátoru došlo ke zhoršení celkové přesnosti.

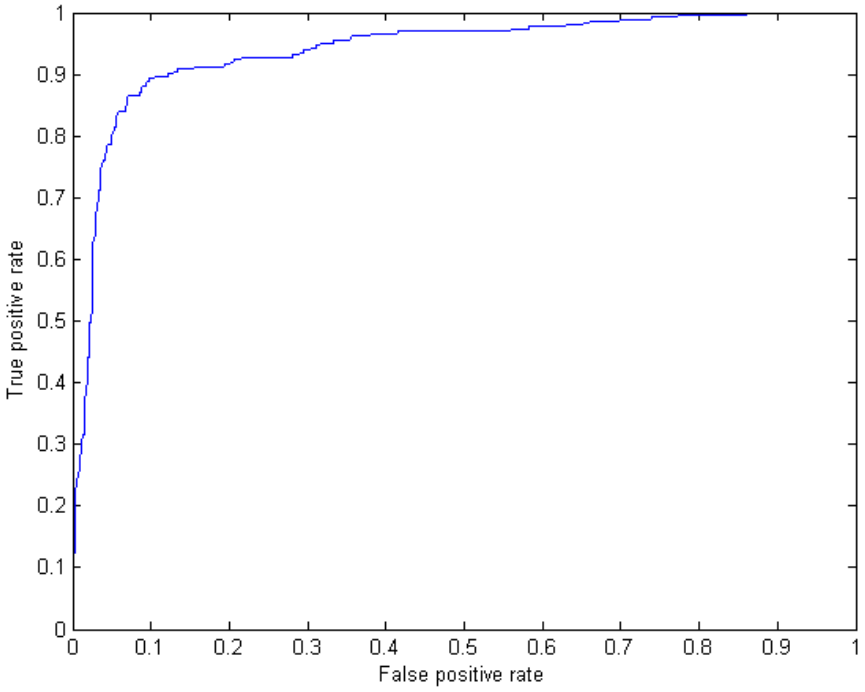
Estimator:	Simple Estimator	Alfa = 0,5
Vyhledávací algoritmus:	Simulated Annealling	
Použití AD-stromu:	Ne	
	Počáteční teplota	10,0
	Delta teploty	0,999
	Počet běhů algoritmu	10.000
	Skórovací funkce	Entropie
Vstupní počet atributů:	9	
Testovací nastavení:	Křížová validace (10x)	

Výstupy

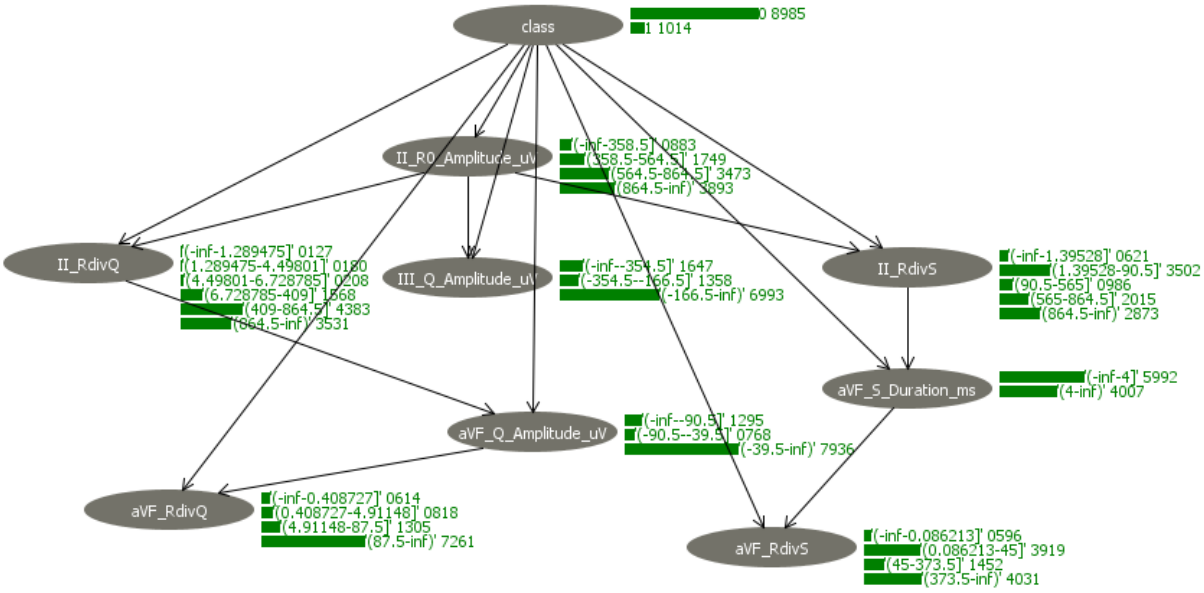
klasifikováno: nemocný	**klasifikováno: zdravý**	
217	**46**	Reálně nemocný
129	**2204**	Reálně zdravý

Počet správných klasifikací:	**93,26** %	(2.421)
Počet chybných klasifikací:	**6,74** %	(175)
Průměrná plocha pod ROC křivkou:	**0,94**	
Falešně pozitivní hodnocení:	**17,5 %** pro nemocné jedince	
	5,5 % pro zdravé jedince	
Senzitivita:	**82,51 %**	
Specificita:	**94,47 %**	

Obecně mi přijde tento klasifikátor vhodnější než 6.4.5, právě kvůli lepší klasifikaci falešně pozitivních nemocných jedinců.



Obrázek 37 - Vizualizace ROC křivky, Simulované žíhání (2), omezený datový soubor



Obrázek 38 - Vizualizace mapy sítě

6.4.7 Naivní Bayes, omezený datový soubor

Zajímavé bude porovnání všech předchozích experimentů s algoritmem naivního Bayese. Ten je také sestaven rychleji, než stejný algoritmus s plným datovým souborem.

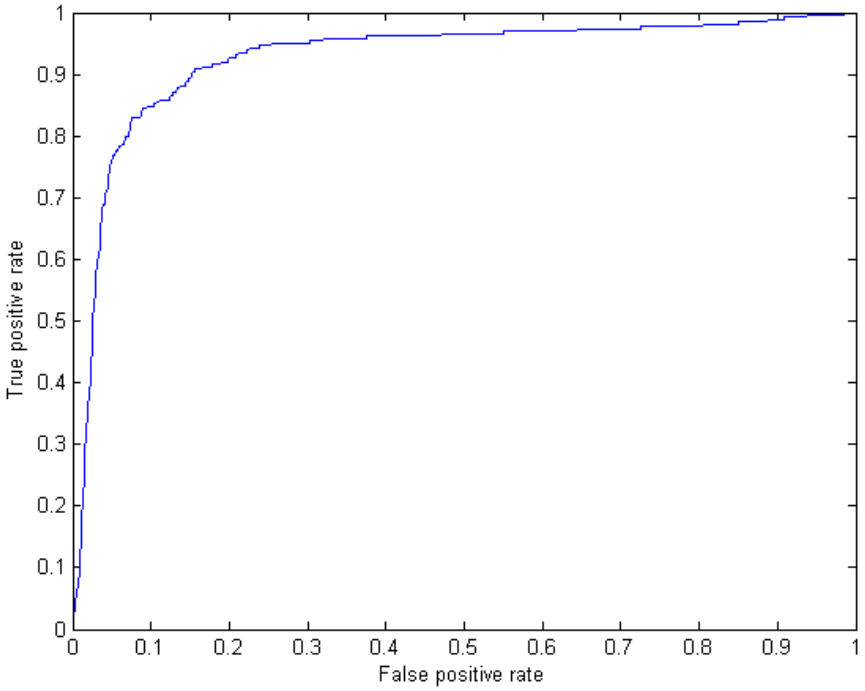
Vstupní počet atributů:	9
Testovací nastavení:	Křížová validace (10x)

Výstupy

klasifikováno: nemocný	klasifikováno: zdravý	
218	**45**	Reálně nemocný
177	**2156**	Reálně zdravý

Počet správných klasifikací:	**91,45** %	(2.374)
Počet chybných klasifikací:	**8,55** %	(222)
Průměrná plocha pod ROC křivkou:	**0,927**	
Falešně pozitivní hodnocení:	**7,6 %** pro zdravé jedince	
	17,1 % pro nemocné jedince	
Senzitivita:	**82,89 %**	
Specificita:	**92,41 %**	

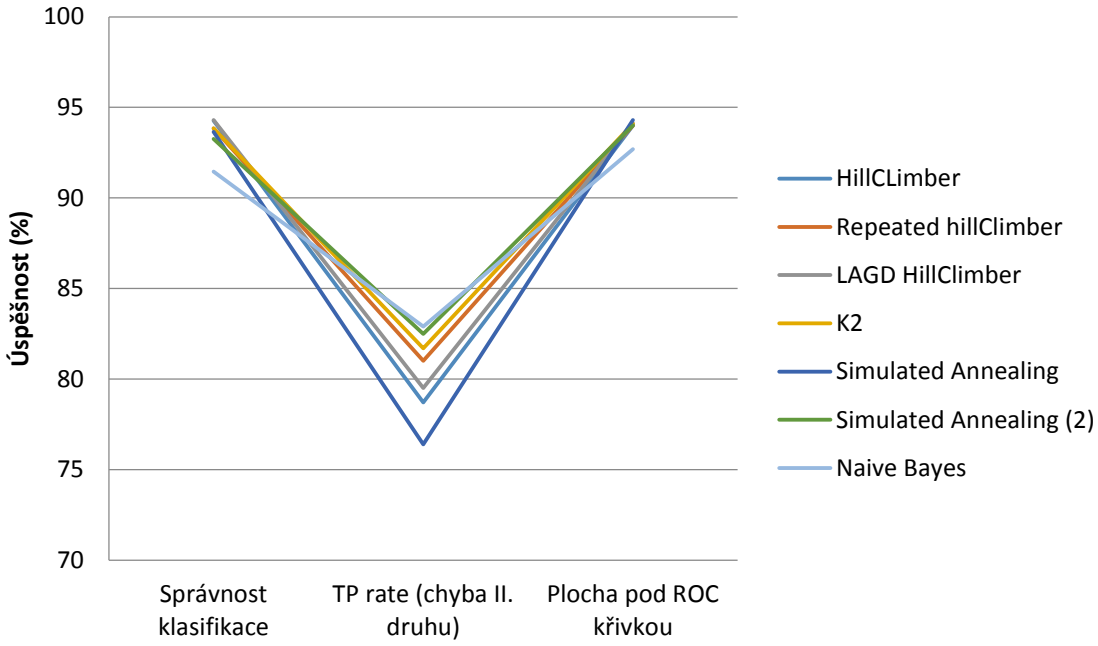
V porovnání s experimentem 0, je toto nastavení vhodnější, protože snížilo falešně pozitivní ohodnocení nemocných pacientů. I když je celkový počet správných ohodnocení nejnižší ze všech zveřejněných dat v této práci, vzhledem k hodnotě senzitivity je klasifikátor velmi dobrým řešením pro danou problematiku.



Obrázek 39 - Vizualizace ROC křivky, naivní Bayes a omezený set dat

6.4.8 Zhodnocení úspěšnosti jednotlivých sítí

Z uskutečněných experimentů je patrné, že úprava výběru atributů měla na všechny sestavené sítě vliv. Zaprvé se podařilo navýšit celkové procentuální vyjádření úspěšnosti a zadruhé se dařilo snižovat chybovost u zdravých jedinců. Poměrně nepříjemný vliv tato operace měla na chybovost požadované hodnoty u nemocných pacientů. Je nezbytné vybírat vhodnou síť tak, aby měla všechny sledované atributy vyrovnané.

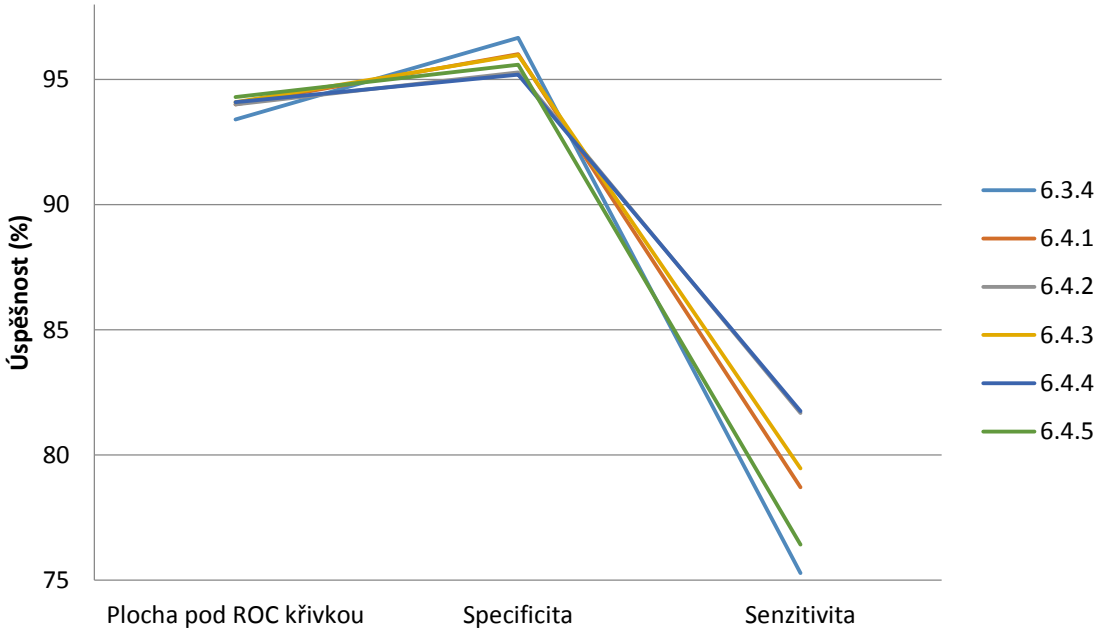


Obrázek 40 - vyhodnocení přesnosti klasifikátorů

V experimentech 0 se tomuto poznatku blížily sítě 6.4.4 a 6.4.6. Síť, sestavená pomocí algoritmu K2, dokonce předčila původní, sestavovanou nad všemi 37 atributy. U simulovaného žíhání byly výsledky mezi různými vstupy velmi podobné. Naivní Bayes sice měl dobré hodnoty True-Positive u nemocných jedinců, ale má vyšší celkovou nepřesnost klasifikace.

7. Závěr

Pro závěrečné porovnání klasifikátorů jsem se rozhodl využít hodnot specificity a senzitivity, přičemž jsem stanovil hranici pro výběr sítí na senzitivitu nad 75% a specificitu nad 95%.



Z výsledků nám plyne několik velmi zajímavých faktů. V závěrečném grafu je patrné, že podmínkám vyhovuje pouze 1 algoritmus ze zpracování plného datového souboru a naproti tomu 5 struktur z omezeného souboru. Je patrné, že některým algoritmům snížení počtu proměnných pomohlo pro zvýšení hodnoty senzitivity, byť za cenu snížení celkové přesnosti klasifikace.

Hlavní přínos předzpracování pomocí AD stromu byl především ve zvýšení rychlosti vytváření a nastavování sítí. Z provedených experimentů bych vybral jako nejvhodnější sítě 6.4.2 a 6.4.4, které mají nejlépe vyvážený poměr mezi požadovanou specificitou a senzitivitou.

Při porovnání kalibrovaných sítí spolu s algoritmy naivního Bayese je patrné, že výsledky NB dosahují nižší úspěšnosti. Ač se na první pohled zdají výsledky vyrovnanější a např. u 6.3.1 jsou falešně pozitivní ohodnocení pro obě skupiny relativně malé, tak důležitý parametr senzitivita je zde téměř nejnižší ze všech provedených experimentů.

Pro porovnání úspěšnosti výstupů z provedených experimentů jsem využil poskytnutého publikovaného článku (*zdroj [13]*). V níže uvedené tabulce je srovnání algoritmů s nejlepšími výsledky.

	SG model C4.5	SG model Ripper	6.4.4 (BN+K2)
Senzitivita:	**63 %**	**78 %**	**81 %**
Specificita:	**97 %**	**95 %**	**95 %**
Precision:	**68 %**	**66 %**	**94 %**

Tabulka 5 - Porovnání úspěšnosti nejlepších rozhodovacích algoritmů, zdroj [13]

Vzhledem k vyšší náročnosti na výpočet sítě při jejím vytváření a nastavování by bylo vhodné použít pro tyto účely paralelní zpracování, pokud to algoritmy povolují. Jak jsem zmínil v kapitole 4.3, některé algoritmy tuto eventualitu poskytují. Pro zvýšení přesnosti je nutné provádět výpočty opakovaně. V této situaci můžeme využít paralelizace. Myslím si, že tuto operaci by bylo vhodné přenechat na cloudové služby či grid. Přirozeně by bylo nutné řešení bezpečnosti, ačkoliv zpracovávaná data nemusí být jmenovitě párovaná na konkrétní osobu, což už jistý druh základní bezpečnosti přináší. Výpočty pro zpřesnění sítě je možné provádět na pozadí jednou za určitou dobu. Výsledná síť i s nastavenými pravděpodobnostními modely by se mohla distribuovat zpět do zdravotnických zařízení. Pak už by nebyly nutné silné výpočetní stroje, pouze se ohodnotí naměřená data pacienta a vyhodnotí se pravděpodobnostní model. To může dát podklad pro vyžádání dalšího, přesnějšího vyšetření.

8. Reference

[1] B. Lerner and R. Malka: *Investigation of the K2 algorithm in learning Bayesian network classifiers*, Applied Artificial Intelligence, 25: 1, 74 — 96, 2011

[2] D. Heckerman: *A tutorial on learning with Bayesian networks*. Technical Report MSR-TR-95–06, Microsoft Research, 1995

[3] V. Mařík, O. Štěpánková, J. Lažanský: *Umělá Inteligence (2)*, Akademia – nakladatelství Akademie věd ČR, 1997

[4] L. Özdamar, M. Demirhan: *Experiments with new stochastic global optimization search techniques*, Computers and Operations Research, 27:841-865, 2000.

[5] Matoušek K.: *Bayesovské sítě, Informační a znalostní systémy*, (https://cw.felk.cvut.cz/wiki/_media/courses/a5m33izs/bayesovske_site.pdf)

[6] I. Zelinka: *Evoluční výpočetní techniky*, BEN, Praha, 2009 (http://arg.vsb.cz/data/Vyuka/09%20BIV%20Evoluce%20-%20Algoritmy.pdf)

[7] K. Murphy: *How to use the Bayes Net Toolbox*, The University Of British Columbia, 2007 (http://www.cs.ubc.ca/~murphyk/Software/BNT/usage.html#K2)

[8] J. Malmivuo, R. Plonsey, Bioelectromagnetism: Principles and Applications of Bioelectric and Biomagnetic Fields. Oxford University Press, USA, 1 ed., 1995. 15.3, 15.4, 15,5, 15.7, 15.8

[9] K. Murphy: *Dynamic Bayesian Networks: Representation, Inference and Learning*, UC Berkley, Computer Science Division 2002

[10] Obrázky elektrod využity z: http://inset.cz/INSHOP/scripts/set.asp?level=1837

[11] R. Bouckaert: *Weka documentation - Bayesian Network Classifiers in Weka for Version 3-5-8*, The University of Waikato, 2008 (http://garr.dl.sourceforge.net/project/weka/documentation/3.5.x/BayesianNetClassifieBa-3-5-8.pdf)

[12] R. Jiroušek: *Metody reprezentace a zpracování znalostí v umělé inteligenci* (http://staff.utia.cas.cz/vomlel/r.pdf)

[13] J. Spilka, V. Chudáček, J. Kužílek, L. Lhotská, M. Hanuliak: *Detection of Inferior Myocardial Infarction: A Comparison of Various Decision*, Cumputing in Cardiology, 37:273 – 276, 2010

[14] R. E. Neapolitan: *Learning Bayesian Networks*, Prentice Hall, 2003

[15] J. R. Hampton: *EKG stručně, jasně, přehledně (překlad 7. vydání)*, Grada, 2013

[16] Úvodní citát převzat z (http://allfamousquotes.com/thomas-bayes/216967)

9. Příloha A – Obsah přiloženého CD

- diplomova_prace.pdf – *elektronická kopie diplomové práce*
- plot.png – *plný grafický výstup rozložení všech atributů*

- vystupy_mereni/full_dataset – *složka obsahující výstupy z experimentů 6.3.x*
- vystupy_mereni/cutted_dataset – *složka obsahující výstupy z experimentů 6.4.x*
- tests – *složka obsahující neúspěšné experimenty s generováním vlastních atributů*

- software/rapidminermulti-core-weka-extension.jar – *rozšíření programu Weka pro program RapidMiner*
- software/weka-3-6-10jre-x64.exe – *použitá verze programu WEKA (x64)*