Univerzita Jana Evangelisty Purkyně, Ústí nad Labem

Počítačové zpracování signálu KI/PZS

Seminární práce

Výpočet tepové frekvence z EKG signálu

Detekce anomálií v EKG signálech

ZS 2024/25

Martin Kučera

Osobní číslo: F22125

Zadání 1

Ve zdrojové databázi najdete celkem 18 měření EKG signálu pro různé věkové skupiny. Signál obsahuje různé anomálie a nemusí být vždy centralizován podle vodorovné osy. EKG signál obsahuje dominantní peaky, které se nazývají R vrcholy. Vzdálenost těchto vrcholů určuje dobu mezi jednotlivými tepy. Počet tepů za minutu je tedy počet R vrcholů v signálu o délce jedné minuty. Navrhněte algoritmus, který bude automaticky detekovat počet R vrcholů v EKG signálech a prezentujte tepovou frekvenci při jednotlivých jízdách/měřeních. Vás algoritmus následně otestujte na databázi MIT-BIH https://physionet.org/content/nsrdb/1.0.0/ a prezentujte jeho úspěšnost vzhledem k anotovaným datům z databáze.

Zadání 2

Ve zdrojové databázi najdete celkem 18 měření EKG obsahující úplné (3 signály) nebo částečné anotace událostí (P,T vlny a QRS komplex). Záznamy EKg obsahují i části, které jsou porušeny vlivem anomálií (vnější rušení, manipulace s pacientem apod.). Navrhněte způsob, jak detekovat tyto úseky a prezentujte statistiku výskytu úseků v měřeních.

Postup – zadání 1

Nejdříve je potřeba načíst data tak, abychom s nimi mohli pracovat. K tomu nám slouží funkce „load_data“, která načítá soubory databáze a vrací vzorky signálu EKG a seznam polí.

Práh signálu

Pro odhad prahu signálu používáme funkci „estimate_threshold“, která na základě zadaného percentilu (v našem případě 95%) odhaduje práh signálu.

Filtr vysokých frekvencí

Pro filtraci signálu využíváme funkci „high_pass_filter“, která provede vysokofrekvenční filtraci signálu a vrátí nám vyfiltrovaný signál EKG.

Odstranění baseline wander

Baseline wander způsobuje nežádoucí vertikální posunutí v signálu, což může komplikovat identifikaci prvků, jako jsou R píky. Pro odstranění tohoto nežádoucího prvku využijeme metodu „remove_baseline_wander_fft“, která využívá Fourierovy transformace.

Nalezení R píků

Pro nalezení R píků v EKG signálu využijeme funkce „find_Rpeaks“, která hledá R píky v EKG signálu s dynamicky se měnící velikostí okna, což je klíčový prvek pří hledání R píků. Výstupem je seznam indexů R píků.

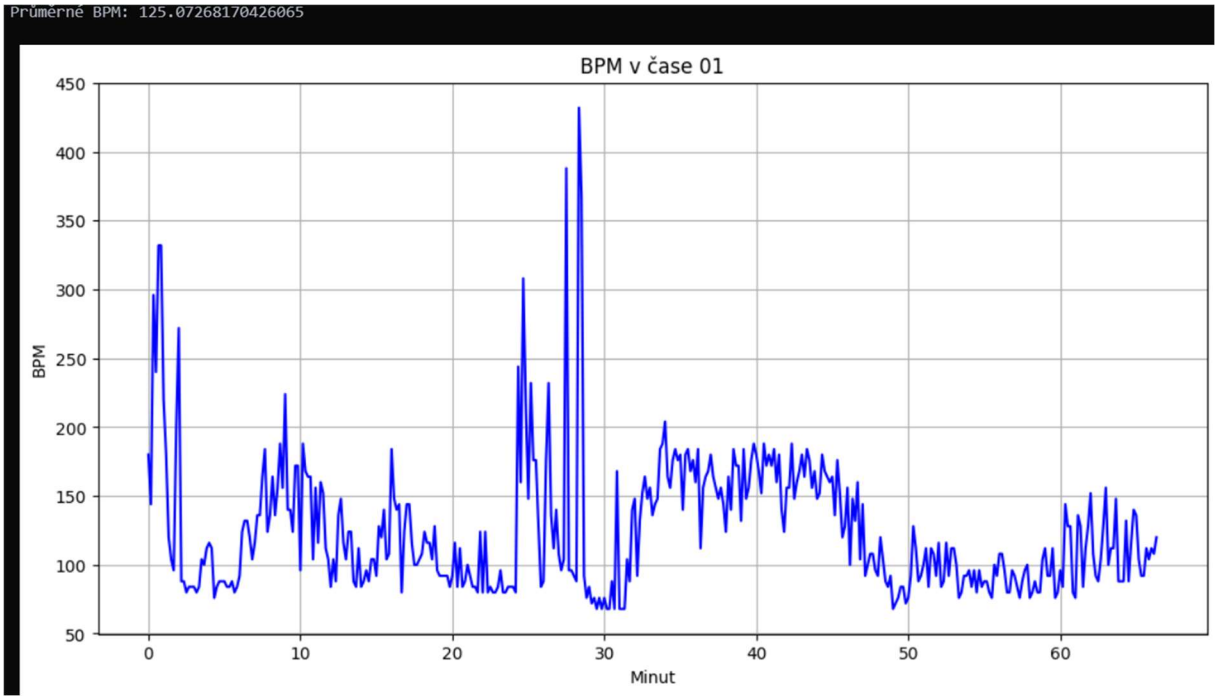
Počet R píků v oknech

Pro nápočet R píků využijeme funkci „count_rpeaks_in_windows“ počítá počet R píků v každých 15 sekundách. Výstupem je seznam počtu R píků v jednotlivých oknech.

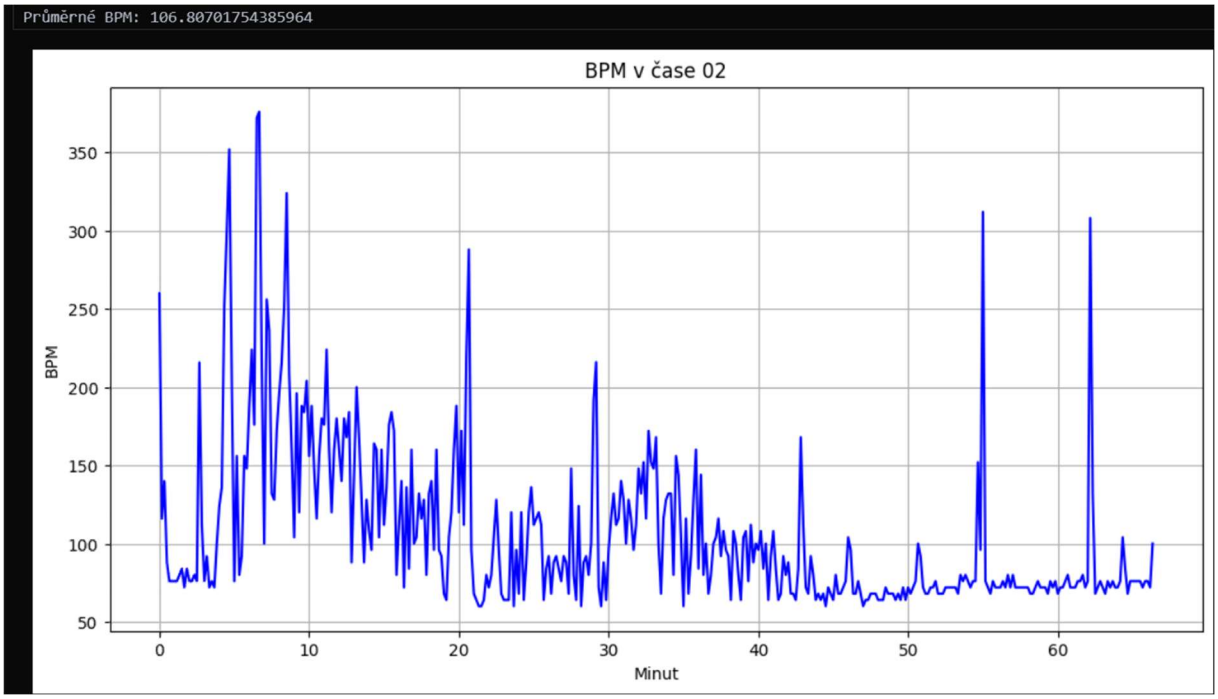
Výsledky

Grafy 1-3 zobrazují křivku průměrného srdečního tepu v čase.

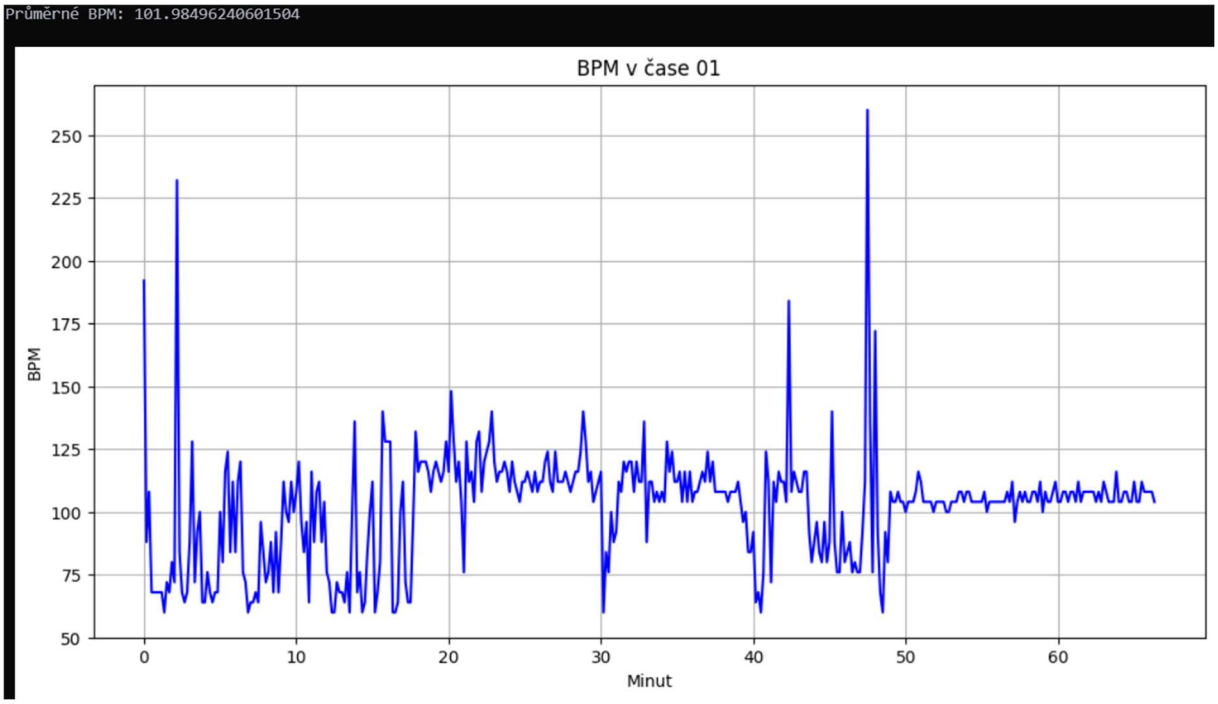
Tabulka obsahuje procentuální úspěšnost algoritmu nad test daty.



Obrázek 1 - bmp signálu 100001



Obrázek 2 - bmp signálu 100002



Obrázek 3 - bmp signálu 103001

Algoritmus se otestoval na datech MIT-BIH https://physionet.org/content/nsrdb/1.0.0/. Viz tabulka s výsledky:

Výsledky nad test daty:

	File	Správně	Špatně	Total	Úspěšnost (%)
0	16265	1141	176	1317	86.636295
1	16272	797	200	997	79.939819
2	16273	1006	237	1243	80.933226
3	16420	905	249	1154	78.422877
4	16483	930	241	1171	79.419300
5	16539	1028	67	1095	93.881279
6	16773	822	569	1391	59.094177
7	16786	754	210	964	78.215768
8	16795	756	303	1059	71.388102
9	17052	859	259	1118	76.833631
10	17453	930	104	1034	89.941973
11	18177	1214	279	1493	81.312793
12	18184	918	283	1201	76.436303
13	19088	977	404	1381	70.745836
14	19090	1034	114	1148	90.069686
15	19093	771	190	961	80.228928
16	19140	995	268	1263	78.780681
17	19830	1263	161	1424	88.693820

Úspěšnost: 79.85%

Obrázek 4 - úspěšnost algoritmu nad test daty

Závěr

V grafech 1-3 vyčteme, že interval je reálný a odpovídá skutečnosti. Ovšem graf obsahuje nepřesnosti, kde hodnoty hodně oscilují za velmi krátký okamžik (např. 2. Graf).

Postup – zadání 2

Už u prvního zadání jsem zjistil, že struktura quality dat a načtení není stejné jako pro testovací data, tuto skutečnost budu muset zohlednit a správně načítat veškeré soubory. Klíčové bude správně načítat anotace a identifikovat význam jednotlivých sloupců, ale postup načítání je relativně stejný.

Volba správné metody detekce anomálií

Pro detekci jsem se rozhodl nejdříve použít všechny možné metody pro filtraci signálu, jmenovitě jsem použil:

- Fourierova transformace (analýza frekvenčního spektra signálu)
- Klouzavý průměr a odchylka (vyhlazení signálu a detekce odlehlých hodnot)
- IQR (interkvartilové rozpětí pro identifikaci extrémních hodnot)
- Detekce vrcholů (nalezení lokálních maxim a minim v signálu)
- Prahová hodnota amplitudy (filtrování na základě předem stanovené úrovně signálu)
- Z-score (normalizace signálu a detekce odlehlých hodnot na základě standardní odchylky)

Dodatečně lze použít ještě butter_lowpass a lowpass_filter, což jsou funkce pro filtrování signálu pomocí nízkofrekvenčního filtru.

Načtení anotací a identifikace abnormálních úseků

V anotacích jsem identifikoval příslušné pole pro anomálie, které použiji pro zobrazení v grafu a porovnání úspěšnosti s mou metodou detekce.

Experimentace při volbě prahových hodnot

V ideálním případě by se nikdy neměla volit statická prahová hodnota, ale dynamická. Bohužel jsem neměl s dynamickými hodnotami vysokou úspěšnost detekce. Viz. graf v sekce VÝSLEDKY.

```
# Dynamické prahování na základě průměru a standardní odchylky
threshold_freq = mean_freq + 2 * np.std(fft_magnitude)
anomalies = np.where(fft_magnitude > threshold_freq)[0]
return anomalies
```

Obrázek 5 - dynamické prahování

```
# Použití Fourierovy transformace k detekci anomálií
anomalies_fft = detect_anomalies_fft(signal, threshold_freq)
```

Obrázek 6 - statické prahování

Metoda detekce pomocí amplitudy

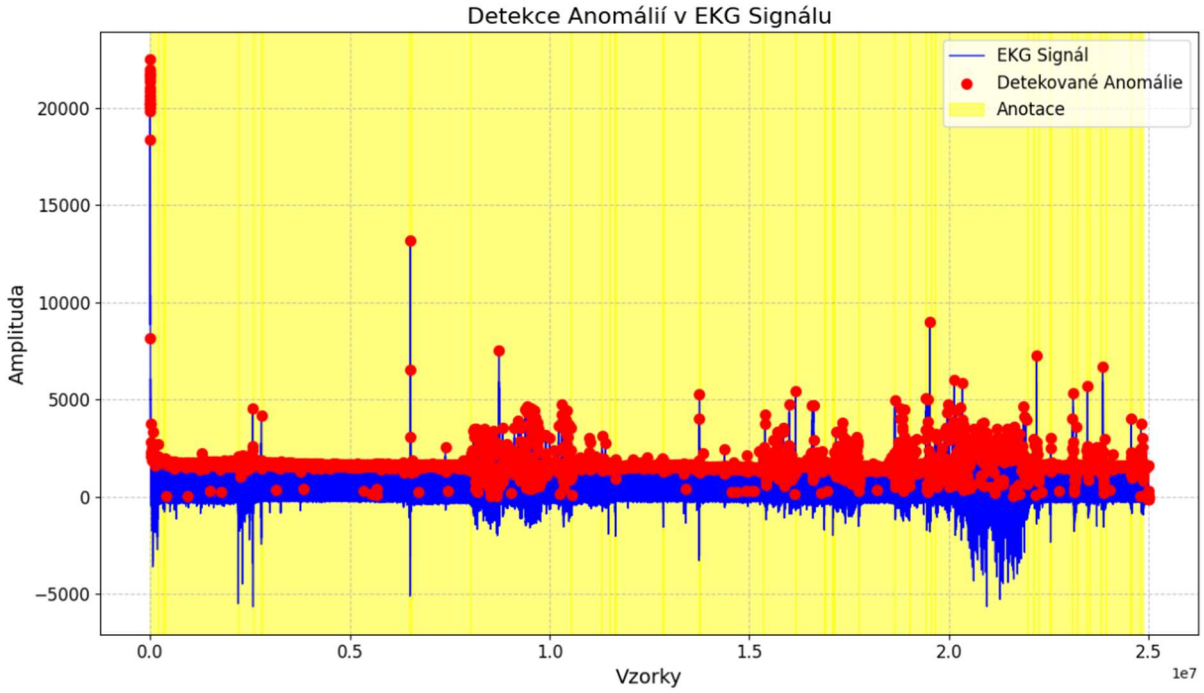
Tato metoda se osvědčila jako nejpřesnější i když je velice primitivní a proto jsem jí použil jako finální výsledek pro detekci.

```python
# Funkce pro detekci anomálií na základě amplitudy
def detect_anomalies_amplitude(signal):
    mean_amplitude = np.mean(signal)
    std_amplitude = np.std(signal)

    # Dynamické prahování
    threshold_amplitude = mean_amplitude + 3.7 * std_amplitude
    anomalies = np.where(np.abs(signal) > threshold_amplitude)[0]

    return anomalies, threshold_amplitude
```

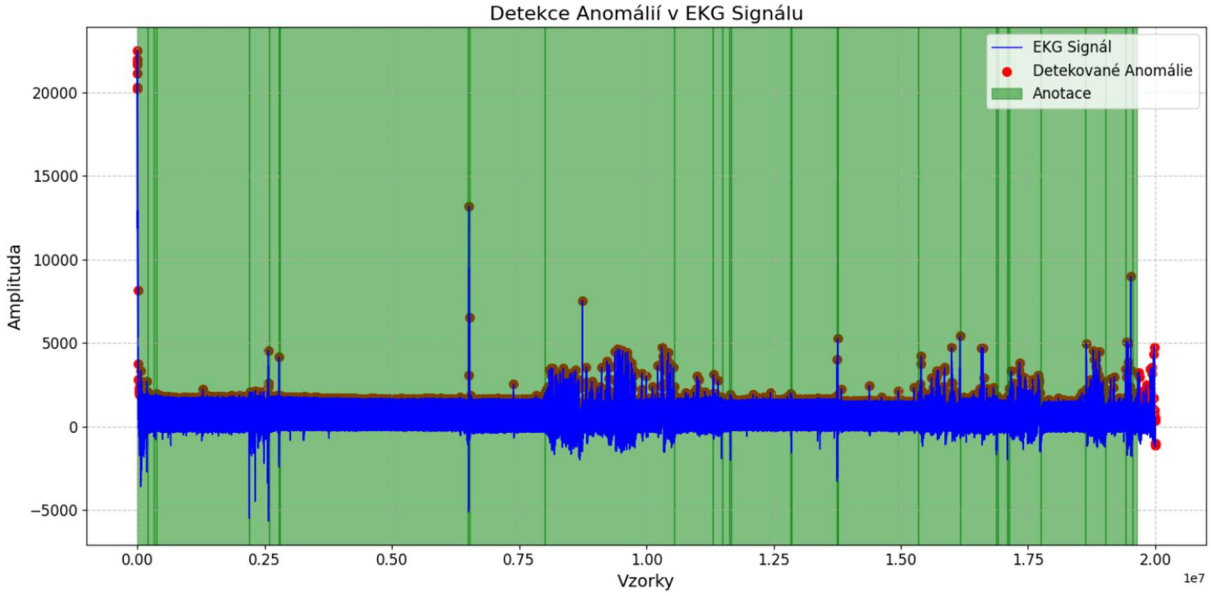
Obrázek 7 - detekce pomocí amplitudy a dynamického prahování

Výsledky

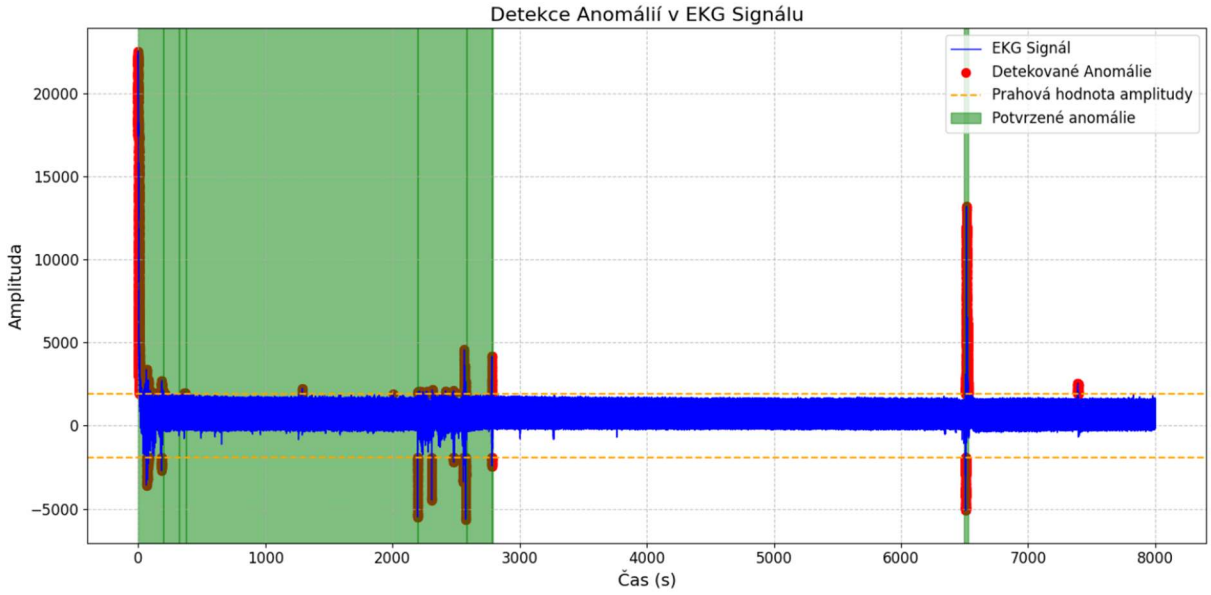
- Graf 1 zobrazuje statickou metodu prahování pomocí kombinace metod
- Graf 2 zobrazuje metodu dynamického prahování pomocí kombinace metod
- Graf 3 zobrazuje metodu detekce pomocí amplitudy a dynamického prahování



Obrázek 8 - graf 1



Obrázek 9 - graf 2



Obrázek 10 - graf 3

Závěr

Jak je z grafů patrné, tak nejpoužitelnější metodou je u grafu 3 a to pomocí dynamického prahování amplitudy. U grafů 1 a 2 byla detekce anomálií dle anotací úspěšná, ale celkový počet anomálií daleko převyšoval skutečné hodnoty.

Finální úspěšnost dle metody použité v grafu 3:

	Statistika	Hodnota
0	Počet shodných anomálií	8.000000
1	Úspěšnost detekce (%)	88.888889

Obrázek 11 - výsledek dle grafu 3