Klasyfikacja zespołów QRS z użyciem metody SVM RBF

Joanna Lebica, Piotr Olbrot

2017

Spis treści

1 Wstęp

Od wielu lat, tematyka automatycznej klasyfikacji elektrokardiogramu jest wnikliwie badana przez wiele zespołów. Wynika to z faktu, że pozwala uzyskać użyteczne informacje dotyczące rytmu oraz funkcjonowania serca. Analiza sygnału elektrokardiograficznego jest uważana za skuteczną metodę diagnostyczną w stanach chorobowych układu sercowonaczyniowego.

W ostatnich latach stosuje się coraz bardziej wyrafinowane metody klasyfikacji, do których zaliczają się m.in. która integrują w sobie elementy logiki rozmytej, sieci neuronowych, ukrytych modeli Markowa, transformacji falkowej oraz maszyny wektorów nośnych.

Jednym z kluczowych aspektów projektu systemu automatycznej klasyfikacji jest właśnie dobór metody klasyfikacji. Metoda SVM jest warta uwagi, ponieważ oparta jest na maksymalizacji obszaru pomiędzy hiperpłaszczyznami rozdzielającymi dane z dwóch grup. Znalazła ona zastosowanie w dziedzinach rozpoznawania obiektów 3-D, obrazowania medycznego i kompresji obrazów. [1]

2 Zarys proponowanego rozwiązania

Rozwiązanie stworzone w ramach projektu składa się z następujących elementów

- Pobranie sygnału EKG z bazy danych MIT-BIH Arrhythmia Database
- Wstępne przetworzenie sygnału polegające na filtracji pasmowoprzepustowej oraz odjęciu linii izoelektrycznej
- Wyznaczenie przybliżonych miejsc występowania zespołów QRS na podstawie pliku .atr dołączonego każdego z badań
- Dla każdego z zespołów, które zostały sklasyfikowane jako normalny - N lub komorowy - V wyznaczenie cech sygnału
- Stworzenie macierzy zawierającej w kolejnych wierszach zestawy cech dotyczącej każdego z zespołów QRS, w ostatniej kolumnie informacja o poprawnym przyporządkowaniu (wartość +1 dla N, wartość -1 dla V)
- Stworzenie zbioru uczącego i testowego
- Uczenie modelu maszyny wektorów nośnych przy użyciu algorytmu sequential minimal optimization
- Walidacja modelu z użyciem zbioru testowego

3 Obliczenie cech kompleksów QRS

Podawanie do algorytmu uczącego całego sygnału bądź też interesujących fragmentów byłoby niepraktyczne z racji dużej złożoności obliczeniowej takiego rozwiązania. Znalezienie prostych do obliczenia cech zespołów QRS, które jednocześnie są w stanie pogrupować dane wejściowe według rodzaju ewolucji serca byłoby najbardziej korzystną sytuacją. [3]

3.1 Matematyczny sposób obliczenia cech

Do analizy wybrano 5 prostych do obliczenia parametrów, do których obliczenia wykorzystuje się fragmenty sygnału uznane jako należące do zespołu QRS. [5] Te parametry to:

- Stosunek pola do obwodu (współczynnik Malinowskiej)

$$p_1 = \frac{\sum_{k=1}^{N} s[k]}{\sum_{k=2}^{N} s[k] - s[k-1]} \tag{1}$$

- Wartość międzyszczytowa

$$p_2 = \frac{\max_{k \in <1,N>} s[k]}{\min_{k \in <1,N>} s[k]} \tag{2}$$

- Procent próbek sygnału, które są ujemne

$$p_3 = 100\% * \frac{\sum_{k=1}^{N} u[k]}{N} \tag{3}$$

Gdzie:

$$u[k] = \begin{cases} 1 & \text{gdy } s[k] < 0 \\ 0 & \text{gdy } s[k] \geqslant 0 \end{cases}$$

- Stosunek maksymalnej prędkości do maksymalnej amplitudy

$$p_4 = \frac{\max_{k \in <3,N>} s[k] - s[k-2] + 2s[k-1]}{|\max_{k \in <1,N>} s[k] - \min_{k \in <1,N>} s[k]|} \tag{4}$$

- Stosunek liczby próbek sygnału, których prędkość przekracza 40% maksymalnej prędkości obserwowanej w sygnale

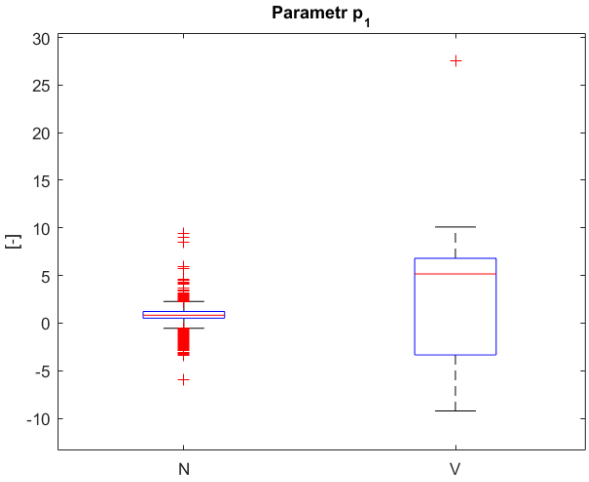
$$p_5 = \frac{\sum_{k=1}^{N} g[k]}{N} \tag{5}$$

Gdzie:

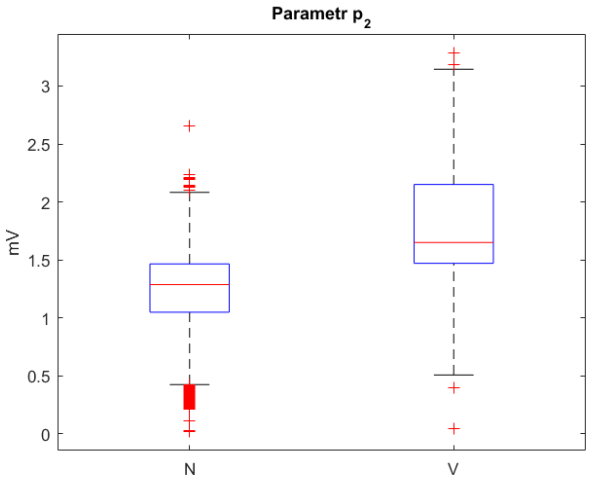
$$g[k] = \begin{cases} 1 & \text{gdy } |s[k] - s[k-1]| > 0.4 \max_{m \in <2,N>} |s[m] - s[m-1]| \\ 0 & \text{gdy } |s[k] - s[k-1]| \leqslant 0.4 \max_{m \in <2,N>} |s[m] - s[m-1]| \end{cases}$$

3.2 Klasyfikacja przy użyciu obliczonych cech

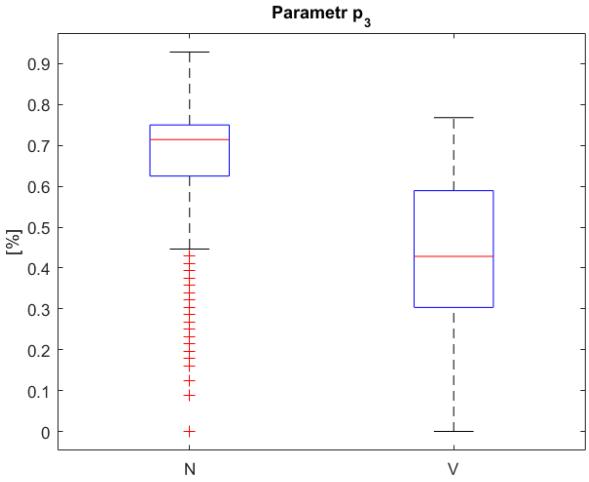
W celu wstępnej oceny wybranych cech, wybrano losowo 5 sygnałów z bazy danych (100.dat, 107.dat, 114.dat, 200.dat, 208.dat). Wyodrębiono z nich 9306 zespołów QRS, które sklasyfikowano jako normalne (N) bądź komorowe (V). Poniżej przedstawione są wykresy pudełkowe pokazujące zakres zmienności kolejnych cech dla obu grup.



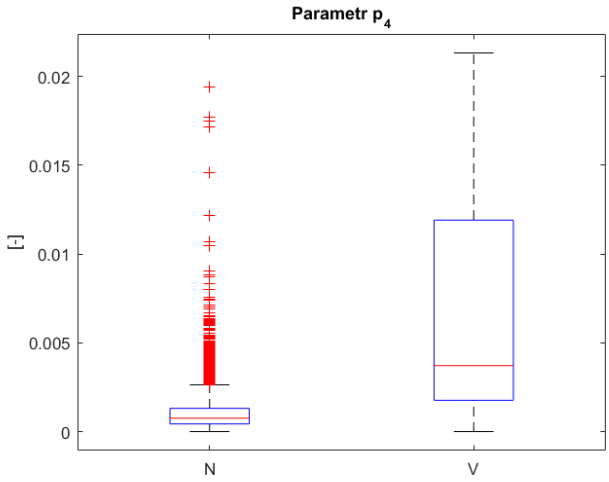
Rysunek 1: Parametr p_1 dla obu grup



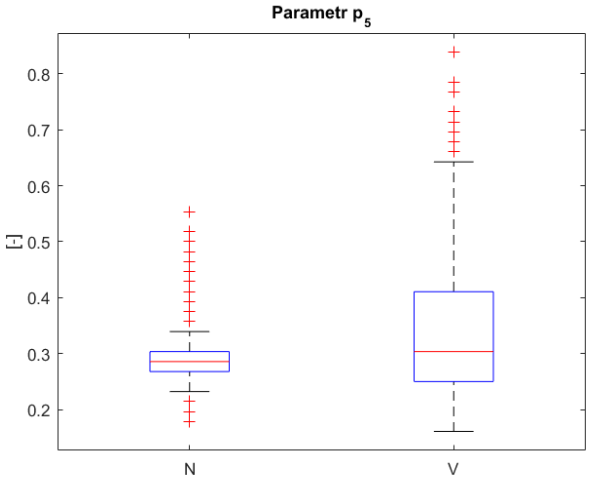
Rysunek 2: Parametr p_2 dla obu grup



Rysunek 3: Parametr p_3 dla obu grup

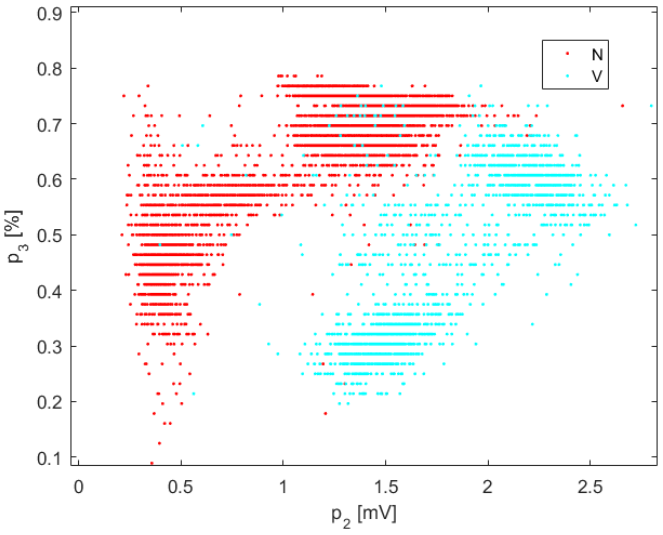


Rysunek 4: Parametr p_4 dla obu grup



Rysunek 5: Parametr p_5 dla obu grup

Jak można zauważyć niektóre cechy (przede wszystkim p_2 i p_3) dobrze dokonują rozróżnienia. W celu potwierdzenia tej obserwacji stworzono wykres obserwacji w przestrzeni p_2 - p_3.



Rysunek 6: Zbiór testowy na płaszczyźnie p_2-p_3

Pokazuje on, że klasyfikacja już w przestrzeni tych dwóch cech może dawać satysfakcjonujące rezultaty. Do algorytmu uczącego podano jednak macierz zawierającą wszystkie 5 cech zespołów QRS.

4 Metoda maszyny wektorów nośnych (SVM)

4.1 Podstawowe założenia metody

Metoda wektorów nośnych dokonuje binarnej (na dwa pozbiory) klasyfikacji z nadzorem. Zbiór uczący składa się z N wektorów

$$x_i \in R^d (i = 1, 2, 3...) \tag{6}$$

z d-wymiarowej przestrzeni cech X. Z każdym wektorem stowarzyszona jest wartość

$$y_i \in \{-1, +1\} \tag{7}$$

Metoda SVM w najprostszym (liniowym) wydaniu szuka hiperpłaszczyzny rozdzielającej dwie klasy w przestrzeni X tak, aby hiperpłaszczyzna była najbardziej oddalona od obserwacji zbioru uczącego (maksymalny odstęp).

W przypadku, gdy zbiory nie są liniowo separowalne, obie klasy są wcześniej mapowane do innej przestrzeni cech (o wyższej wymiarowości). Więcej o tym w kolejnym podrozdziale.

Decyzja dotycząca klasyfikacji polega na obliczeniu wartości funkcji dyskryminacyjnej $f(x)$.

$$f(x) = \omega^* \Phi(x) + b^* \tag{8}$$

Optymalna hiperpłaszczyzna jest zdefiniowana przez wektor wag

$$\omega^* \in R^{d'} \tag{9}$$

i bias

$$b^* \in R \tag{10}$$

Optymalizacja klasyfikatora polega na minimalizacji błędu (tj. sumy kar za źle sklasyfikowane obserwacje) oraz maksymalizacji szerokości odległości między hiperpłaszczyzną rozdzielającą a obserwacjami. Przez odpowiedni dobór parametrów można większą wagę postawić na jedną z tych zależności.

4.2 Radial Basis Function

Radial Basis Function (lub radialna funkcja bazowa) jest funkcją zależącą jedynie od odległości od pewnego przyjętego środka. Każda funkcja spełniająca warunek

$$\phi(x) = \phi(||x||) \tag{11}$$

jest funkcją radialną.

Wprowadzenie radialnej funkcji bazowej związane jest z tym, że w wielu przypadkach niemożliwy jest skuteczny podział w wyjściowej przestrzeni cech.

Funkcje bazowe usuwają tę przeszkodę przenosząc problem klasyfikacji do przestrzeni o wyższej wymiarowości, co schematycznie pokazuje poniższy rysunek.

W algorytmie wykonanym w ramach projektu wykorzystano Gaussowską radialną funkcję bazową opisaną wzorem:

$$k(\vec{x}_i, \vec{x}_j) = exp(\frac{||\vec{x}_i - \vec{x}_j||^2}{2\sigma^2}) \tag{12}$$

4.3 Wykorzystana metoda implementacji

4.3.1 Matematyczne podstawy

Zastosowana metoda optymalizacji modelu SVM to Sequentual minimal optimization w skrócie SMO. Jest to metoda rozwiązywania zadania optymalizacyjnego polegającego na znalezieniu ekstremum funkcji kwadratowej. Przy implementacji oparto się na opisie z artykułu [4].

SMO jest efektywną metodą rozwiązania problemu optymalizacji, który może być opisany następującym równaniem

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} y^i y^j \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle \tag{13}$$

Gdzie:

$$0 \leqslant \alpha_i \leqslant C, \text{dla } i = 1, 2, 3, ..., m \tag{14}$$

$$\sum_{i=1}^{m} y^{(i)} \alpha_i = 0 \tag{15}$$

Warunek znalezienia optymalnego rozwiązania (zwany warunkiem Karusha-Kuhna-Tuckera) dla tego problemu jest zdefiniowany następująco

$$\alpha_i = 0 \implies y^{(i)}(\omega^T x^{(i)} + b) \geqslant 1 \tag{16}$$

$$\alpha_i = C \implies y^{(i)}(\omega^T x^{(i)} + b) \leqslant 1 \tag{17}$$

$$0 < \alpha_i < C \implies y^{(i)}(\omega^T x^{(i)} + b) = 1 \tag{18}$$

Algorytm SMO iteruje aż do momentu, gdy te warunki zostaną spełnione przez to zapewniając jego zbieżność do optymalnego rozwiązania.

Składa się on 3 kroków [4]:

- Wybór parametrów α W zaimplementowanym algorytmie zastosowano najprostszy wybór parametrów, tj. w pętli iteruje się przez wszystkie α_i, a następnie losuje się α_j takie, że $i \neq j$. Istnieje niewielkie prawdopodobieństwo, że któraś z par parametrów, które wprowadziłyby istotną zmianę do ostatecznego wyniku zostanie pominięta. Założono jednak, że algorytm musi wykonać przynajmniej 10 (domyślna wartość) "pustych przebiegów", aby zminimalizować takie prawdopodobieństwo. "Pusty przebieg"polega na tym, że w danej iteracji pętli zewnętrznej nie zostają wprowadzone żadne zmiany do modelu.

- Optymalizacja α_i i α_j Obliczane są ograniczenia, które musi spełniać α_j, a natępnie rozwiązywany jest ograniczony problem optymalizacyjny. Następnie obliczana jest α_j optymalna dla danego kroku, która jest następnie ogranicza jeśli posiada wartość spoza założonego zakresu. Na podstawie α_j obliczana jest nowa wartość α_i. Odpowiednie wzory dostępne są [4].

- Obliczenie biasu b Po obliczeniu α_i i α_j, wybierana jest taka wartość b, żeby warunek znalezienia optymalnego rozwiązania był spełniony. Sprawdzane jest, czy obliczone uprzednio α_i i α_j znajdują się na granicach przedziału $< 0, C >$. Jeśli nie, to dane rozwiązanie jest prawidłowe. Jeśli oba nie są na granicach przedziału wartość b jest średnią arytmetyczną b_1 i b_2 obliczoną odpowiednio dla α_i i α_j. Szczegóły: [4].

4.3.2 Implementacja w C++

W przygotowaniu programu w C++ wykorzystano bibliotekę Eigen, która okazała się szczególnie przydatna z racji wbudowanych funkcji obsługujących macierze i wektory. Najważniejsze elementy programu to:

- Klasa SVM - jest to klasa obsługująca proces uczenia modelu (metoda train), a następnie jego testowania (metoda predict), posiada szereg funkcji prywatnych funkcji pomocniczych

- Funkcja parseData - jest to funkcja odpowiadająca za wczytanie odpowiednio zdefiniowanego pliku CSV i stworzenie na jego podstawie struktury zawierającej 2 macierze (treningową, testową) i 2 wektory (poprawne przyporządkowanie dla zbioru treningowego i testowego)

- Funkcje writeDecisionToCSV i appendToOutputFile - odpowiadające za zapis do plików działania modelu

- Struktura SVMOptions - zawierający parametry działania metody maszyny wektorów nośnych oraz domyślny konstruktor przyjmujący wartości domyślne

- Struktura ParsedData - zawierający całość danych potrzebnych do uczenia i testowania modelu

Funkcja główna main zawiera kod wywołujący poszczególne elementy programu:

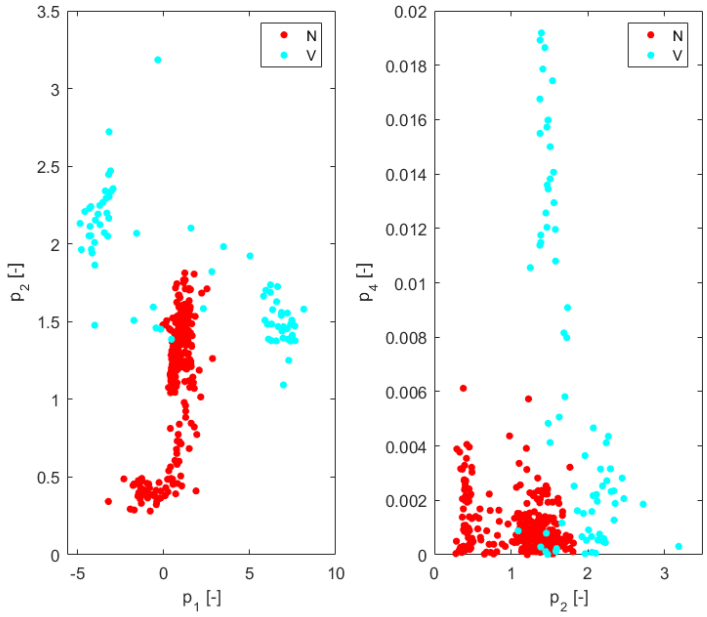
```
int main(int argc, char** argv){

// wczytaj dane z plikow, ktorych nazwe podano w konsoli
// - zalozono, ze pliki z danymi znajduja sie w tym samym
// folderze co program

ParsedData data = parseData(argv);

// utworz model, podano domyslne parametry tj. C(1.0),
// tol(0.0001), sigma(0.5), iterLimit(10000),
```

```
// passLimit(10)

SVM model(data.trainSet, data.trainSetOut, SVMOptions());

// trenuj model na danych podanych w konstruktorze
model.train();

// dokonaj klasyfikacji danych podanych w pliku
// testowym; zapisz wektor z klasyfikacja do pliku CSV o
// nazwie out.csv

VectorXd decision = model.massPredict(data.testSet);
writeDecisionToCSV(decision);

// ocen dzialanie klasyfikatora (ilosc poprawnie
// sklasyfikowanych ewolucji serca N/V) oraz czas
// uczenia i testow

evaluateModel(data.testSetOut, decision);
appendToOutputFile(model.trainTime, model.testTime);

return 0;
}
```

5 Wyniki

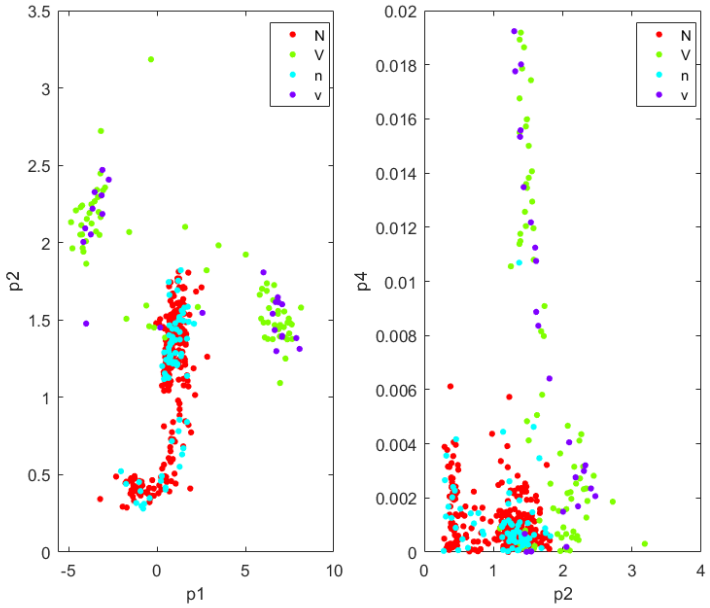
Spośród ponad 9 tysięcy obserwacji do trenowania wybrano losowo 400 (322 zespoły QRS typu N, 76 zespoły typu V, jest to naturalna proporcja w jakiej występowały te zespoły w przebiegach 5 sygnałów, które wybrano losowo z bazy danych), a do testowania 100.

Zbiór uczący pokazano poniżej, w płaszczyźnie p_1-p_2 (po lewej) i p_2-p_4 (po prawej).



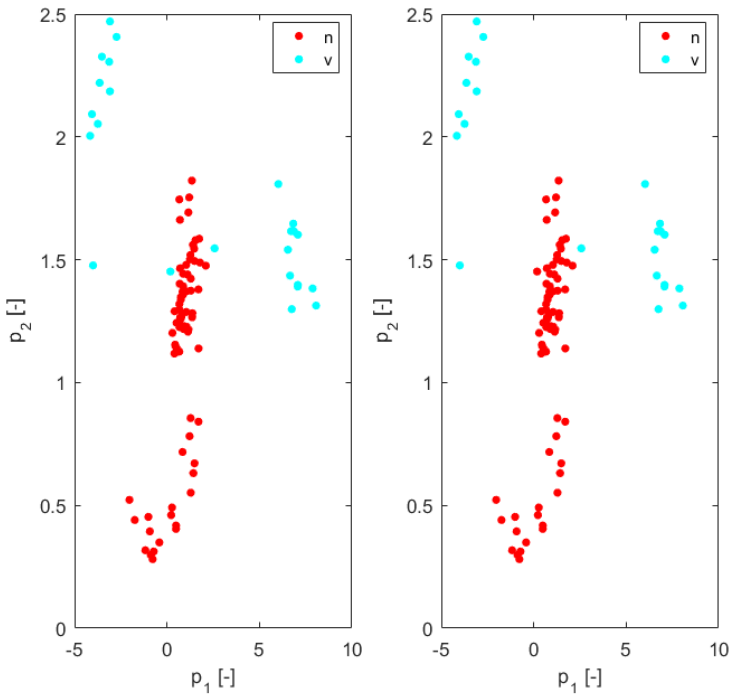
Rysunek 7: Zbiór uczący na płaszczyźnie p_1-p_2 (po lewej) i p_2-p_4 (po prawej)

Poniżej zwizualizowano zbiór testowy (n/v) i treningowy (N/V) na jednym wykresie w tych samych płaszczyznach.



Rysunek 8: Zbiór uczący i testowy na płaszczyźnie p_1-p_2 (po lewej) i p_2-p_4 (po prawej)

Zaimplementowana metoda klasyfikacji poprawnie przyporządkowała 99 spośród 100 obserwacji. Na wykresie porównawczym (po lewej poprawne przyporządkowanie, po prawej wynik działania algorytmu) oznaczono źle sklasyfikowaną obserwację.



Rysunek 9: Poprawnie sklasyfikowany zbiór uczący na płaszczyźnie p_1-p_2 (po lewej) i wynik działania algorytmu w tej samej płaszczyźnie (po prawej)

Porównanie działania algorytmu SVM RBF z algorytmami innych grup znajduje się w osobnym raporcie.

6 Wnioski

Klasyfikator zespołów QRS stworzony w ramach projektu działa satysfakcjonująco dla wybranych cech fragmentów sygnału i zaimplementowanego algorytmu SVM.

W ramach projektu wykorzystano sprawdzone, ale proste matematycznie parametry sygnału i potwierdziły one swoją zdolność do rozróżniania zespołów QRS o różnym pochodzeniu pobudzenia. W ramach projektu porównywano zespoły sklasyfikowane jako normalne (N) oraz komorowe (V). Należy przyjąć, że gdyby dokonano bardziej szczegółowego podziału działanie klasyfikatora uległoby pogorszeniu. Już w początkowej fazie projektu stwierdzono, że bardzo ciężko byłoby odróżnić zespoły normalne (N) od nadkomorowych (SV) wg klasyfikacji. [2]

Spośród wszystkich metod SVM realizowanych w projekcie metoda SVM RBF zapewnia najlepszą skuteczność rozdziału cech, która w przestrzeni, w której są zdefiniowane nie są liniowo separowalne, co stanowi niewątpliwą zaletę tego algorytmu. [1]

Czas trenowania klasyfikatora tj. ok. 20 sekund (na komputerze jednego z autorów) dla 400 obserwacji po 5 cech każda można uznać za zadowalający, ale z pewnością istnieje wiele metod dalszej optymalizacji kodu.

Literatura

[1] Bazi Yakoub, Melgani Farid *Classification of Electrocardiogram Signals With Support Vector Machines and Particle Swarm Optimization* IEEE Transactions on Information Technology in Biomedicine 12, 2008

[2] da Luza E. J. S, et al. *ECG-based Heartbeat Classification for Arrhytmia Detection: A Survey* Computer methods and programs in biomedicine, 2016

[3] Augustyniak Piotr *Przetwarzanie sygnałów elektrodiagnostycznych* Wydawnictwa AGH, Kraków, Poland, 2001

[4] Husanbir Singh Pannu *Simplified SMO Algorithm*
`http://math.unt.edu/ hsp0009/smo.pdf`

[5] Augustyniak Piotr *Optymalizacja wyboru reprezentacji zespołów skurczowych dla celów klasyfikacji zapisów holterowskich*
`http://galaxy.uci.agh.edu.pl/ august/pub/pdf/p15.pdf`