



**POLITECHNIKA
GDAŃSKA**

WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI I INFORMATYKI

Imię i nazwisko studenta: Adam Władziński

Nr albumu: 125483

Studia drugiego stopnia

Forma studiów: stacjonarne

Kierunek studiów: Elektronika i telekomunikacja

Specjalność: Optoelektronika

PRACA DYPLOMOWA MAGISTERSKA

Potwierdzenie przyjęcia pracy	
Opiekun pracy	Kierownik Katedry
<i>podpis</i>	<i>podpis</i>
dr inż. Marcin Gnyba	

Data oddania pracy do dziekanatu:



**POLITECHNIKA
GDAŃSKA**

WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI I INFORMATYKI



OŚWIADCZENIE dotyczące pracy dyplomowej zatytułowanej:

Wykorzystanie metody kalibracji wielowariantowej do przetwarzania danych pomiarowych w spektroskopii laserowej

Imię i nazwisko studenta: Adam Władziński

Data i miejsce urodzenia: 03.12.1990, Gdynia

Nr albumu: 125483

Wydział: Wydział Elektroniki, Telekomunikacji i Informatyki

Kierunek: elektronika i telekomunikacja

Poziom kształcenia: drugi

Forma studiów: stacjonarne

Świadomy(a) odpowiedzialności karnej z tytułu naruszenia przepisów ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (Dz. U. 2018 poz. 1191 z późn. zm.) i konsekwencji dyscyplinarnych określonych w ustawie z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce (Dz. U. 2018 poz. 1668 z późn. zm.),¹ a także odpowiedzialności cywilnoprawnej oświadczam, że przedkładana praca dyplomowa została opracowana przeze mnie samodzielnie.

Niniejsza praca dyplomowa nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadaniem tytułu zawodowego.

Wszystkie informacje umieszczone w ww. pracy dyplomowej, uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami zgodnie z art. 34 ustawy o prawie autorskim i prawach pokrewnych.

Potwierdzam zgodność niniejszej wersji pracy dyplomowej z załączoną wersją elektroniczną.

STRESZCZENIE

Istotnym problemem w diagnostyce biomedycznej jest szybkość wykonywania analiz, która determinuje szybkość pełnej diagnostyki i moment dostępności celowanego leczenia. Jest też ważna w innych zbliżonych zastosowaniach, np. przy identyfikacji śladów krwi w kryminalistyce. Praca dyplomowa dotyczy zagadnień analizy oraz klasyfikacji złożonych widm Ramana próbek organicznych, w szczególności krwi ludzkiej i zwierzęcej. Wykorzystując dwa algorytmy przetwarzania wektorów: liniową analizę głównych składowych (w skrócie PCA, *Principal Components Analysis*) oraz nieliniową maszynę wektorów nośnych SVM (*Support Vector Machine*) podjęto próbę klasyfikacji widm ramanowskich pochodzenia organicznego. Próbkami płynnej krwi przechowywanymi w lodówce. W pracy udało się dokonać klasyfikacji krwi pochodzącej od różnych dawców. Udało się rozróżnić krew pochodzenia ludzkiego oraz zwierzęcego, jak również krew, która zawierała w sobie roztwór leku. Opracowano algorytm, który rozwiązuje podstawowy problem zbyt dużej ilości wymiarów danych pomiarowych w przypadku widm pochodzących z matryc detekcyjnych o wysokiej rozdzielczości (rozdzielczość jest wielkością ilości cech danego wektora). Problem zbyt dużej ilości wymiarów jest szczególnie odczuwany w przypadku pracy ze środowiskiem Matlab. Środowisko posiada ograniczenia programistyczne, dotyczące wielkości przetwarzania danych oraz skomplikowania obliczeń na złożonych wektorach. Przeszkody w postaci tych ograniczeń zostały pokonane dzięki rozwiązaniom w architekturze bazy danych widm pomierzonych oraz algorytmowi ich przetwarzania. Baza danych jest rozwiązaniem hybrydowym, łączącym hierarchiczny model dostępu do danych z relacyjnym modelem macierzy do przetwarzania widm Ramana. Problem wielu wymiarów został rozwiązany dzięki wstępnej analizie danych za pomocą analizy głównych składowych (PCA). Po przeanalizowaniu widma, do dalszego przetwarzania były wybierane tylko te składowe z wektorów, które niosły najwięcej informacji. Ilość wybranych składowych została ograniczona do 256, dzięki czemu możliwym było zastosowanie przetwarzania SVM. Taka kompresja była oczywiście stratna, jednak ilość traconej informacji była poniżej 1%. Aplikacja opracowana w ramach pracy radziła sobie bez problemu po takiej kompresji. Dodatkowym atutem zmniejszenia liczby wektorów za pomocą PCA, była możliwość porównywania danych pochodzących z różnych systemów pomiarowych. W starszych rozwiązaniach wykorzystywano kamery o innej rozdzielczości, co utrudniało analizę bez unifikacji danych. Wybierane wektory przechowywały najwięcej informacji, były więc najbardziej odpowiednie do analizy SVM.

Słowa kluczowe: spektroskopia ramanowska, Raman, analiza krwi, maszyna wektorów nośnych, analiza głównych składowych, PCA, SVM, stanowisko pomiarowe spektroskopii Ramana

Dziedzina nauki i techniki, zgodnie z wymogami OECD: Nauka o komputerach, informatyka i bioinformatyka

ABSTRACT

An important problem in biomedical diagnostics is the speed of analysis, which determines the speed of full diagnostics and the moment of availability of the targeted treatment. It is also important in other similar areas, e.g. identification of blood traces in forensics. The thesis concerns the issues of analysis and classification of complex Raman spectra of organic samples, especially human and animal blood. Using two vector processing algorithms: linear analysis of the principal components (PCA, *Principal Components Analysis*) and non-linear Support Vector Machine (SVM), an attempt was made to classify Raman spectra of organic origin. Organic samples were liquid blood samples stored in a refrigerator. In this work, we were able to classify blood from different donors. A distinction was made between the blood of human and animal origin and blood that contained a drug solution. An algorithm was developed that solves the basic problem of too many dimensions of measurement data in the case of spectra from high-resolution detection matrices (resolution is the size of the characteristics of a given vector). The problem of too many dimensions is particularly noticeable when working with the Matlab environment. The environment has programming limitations, concerning the size of data processing and the complexity of calculations on complex vectors. Obstacles in the form of these limitations have been overcome by solutions in the architecture of the database of measured spectra and the algorithm of their processing. The database is a hybrid solution, combining a hierarchical model of access to data with a relational model of a matrix for processing Raman spectra. The problem of many dimensions has been solved by a preliminary analysis of the data using the principal components analysis (PCA). After analyzing the spectrum, only those vector components that carried the most information were selected for further processing. The number of selected components was limited to 256, making SVM processing possible. Such compression was loss-making, but the amount of lost information was less than 1%. The application developed as part of the work was able to cope with such compression without any problems. An additional advantage of reducing the number of vectors using PCA was the ability to compare data from different measurement systems. Older solutions used cameras with different resolutions, which made it difficult to analyze without data unification. The selected vectors stored the most information, so they were the most suitable for SVM analysis.

Keywords: Raman spectroscopy, Raman, blood analysis, support vector machine, principal component analysis, PCA, SVM, Raman spectroscopy measuring station

WYKAZ SKRÓTÓW

$\bar{\nu}_0, \nu_0, \lambda_0$	–	liczba falowa, częstotliwość i długość fali wzbudzającej
$\bar{\nu}_{st}, \nu_{st}, \lambda_{st}$	–	liczba falowa, częstotliwość i długość fali rozproszenia Ramanowskiego stokesowskiego
$\bar{\nu}_{ast}, \nu_{ast}, \lambda_{ast}$	–	liczba falowa, częstotliwość i długość fali rozproszenia Ramanowskiego antystokesowskiego
c	–	prędkość światła w próżni
$h\nu_0$	–	energia fali padającej
$M(E_i)$	–	energia cząsteczki
ν_{if}	-	jest to częstość przejścia pomiędzy poziomem energetycznym E_h oraz poziomem E_i
I_0	–	natężenie promieniowania wzbudzającego
\hat{c}	–	koncentracja oscylatorów, dla danego pasma ramanowskiego
l	–	długość obszaru aktywizacji rozpraszania ramanowskiego
a	–	prawdopodobieństwo przejścia molekuly pomiędzy stanami oscylacji i
a	–	polaryzowalność centrum oscylatora
q_i	–	wychylenie oscylatora ze stanu równowagi
I_{st}	–	natężenie sygnału widma ramanowskiego w paśmie stokesowskim
I_{ast}	–	natężenie sygnału widma ramanowskiego w paśmie antystokesowskim
ν_R	–	częstotliwość drgania normalnego układu atomowego(η^*c)
Q	–	amplituda drgań normalnych
E_w	–	amplituda pola elektrycznego fali wzbudzającej
PCA	-	principal component analysis
SVM	-	Support Vector Machine

SPIS TREŚCI

1. WSTĘP	8
2. PLAN I CEL PRACY	10
2.1. Praca z danymi pochodzącymi bezpośrednio ze stanowiska pomiarowego.....	10
2.2. Rozwiązanie problemów związanych z przygotowaniem i przetworzeniem danych pomiarowych.....	11
2.3. Klasyfikowanie danych pomiarowych	11
2.4. Pomiary wykonane do analizy	11
3. PODSTAWY TEORETYCZNE	13
3.1. Rozpraszanie Rayleigha oraz Ramana	13
3.2. Rozpraszanie promieniowania	13
3.3. Przesunięcie oscylacyjne w rozpraszaniu Ramana	14
3.4. Promieniowanie Ramanowskie w ujęciu mechaniki klasycznej	16
3.5. Badanie widm ramanowskich substancji organicznych	18
3.6. Przykładowe widma ramanowskie uzyskane w pomiarach	20
3.7. Identyfikacja problemu	25
4. OPIS UKŁADU POMIAROWEGO ORAZ BADANYCH PRÓBEK.....	26
4.1. Układ pomiarowy	26
4.2. Badane próbki (krew)	28
5. ALGORYTMY STOSOWANE DO PRZETWARZANIA DANYCH	30
5.1. Zastosowanie algorytmów przetwarzania danych.....	30
5.2. Algorytm PCA.....	30
5.3. Algorytm SVM.....	31
5.4. LSSVM (Least Squares Support Vector Machines)	32
6. DANE DO KLASYFIKACJI.....	35
6.1. Dane do strojenia algorytmu.....	35
6.2. Jakość danych	35
6.3. Cechy danych.....	35
6.4. Pochodzenie danych.....	36
7. DODATKOWE ROZWIĄZANIA PROGRAMISTYCZNE W PRACY MAGISTERSKIEJ	37
7.1. Baza danych.....	37
7.2. Hierarchiczny element systemu hybrydowego składowania danych	37

7.3. Relacyjny element systemu hybrydowego składowania danych.....	38
8. WYNIKI	39
9. DYSKUSJA I WNIOSKI	44
10. WYKAZ LITERATURY	46
11. ZAŁĄCZNIKI - KOD ŹRÓDŁOWY	49
11.1. Kod algorytmu tworzącego element relacyjny bazy danych	49
11.2. Kod algorytmu przetwarzającego wstępne skompresowane dane, oraz tworzenie bazy danych do klasyfikacji	59
11.3. Kod źródłowy funkcji kompresującej PCA.....	69
11.4. Kod funkcji wykreślającej i obliczającej SVM	80

1. WSTĘP

Medycyna, biotechnologia, nauki kryminalistyczne potrzebują w coraz większym zakresie szybkiej diagnostyki. Próbkę biologiczną wystawioną poza organizm żywy szybko się degenerują, często nieodwracalnie. W wyżej wymienionych dziedzinach inżynierii potrzeba metody badania, która będzie mogła być zastosowana w bardzo krótkim czasie po pozyskaniu próbki. Przykładowo, podawanie leku lub narkozy wymaga monitorowania ilości leku we krwi pacjenta w czasie rzeczywistym. W przypadku śledztwa, próbki biologiczne pobrane na miejscu zdarzenia, powinny zostać przebadane w kilka godzin po pobraniu. Czas w jakim musi zostać wykonana analiza jest kluczowym czynnikiem. Odpowiednio krótki czas może umożliwić znalezienie winnego przestępstwa. Zgodnie z policyjnymi statystykami największa szansa na ujęcie winnego następuje w ciągu 24 godzin od popełnienia przestępstwa.

Badanie obiektów o charakterze biologicznym naraża na wiele trudności - skomplikowanie pod względem składu, niszczenie pod wpływem środowiska (atmosfery, promieniowania UV), czy mnogość ich rodzajów. Pomiar tak skomplikowanych próbek oraz ich analiza jest możliwa z wykorzystaniem nowoczesnych metod analizy matematycznej. Próbkę możemy badać pod wieloma kątami. Dla krwi możemy porównywać jej skład makro i mikro elementarny, gęstość, organizm pochodzenia, stan skupienia.

Wymienione w pierwszym akapicie przykładowe zastosowania wymagają spełnienia określonych warunków. Podstawowym i decydującym czynnikiem, jakim musi się badanie charakteryzować jest czas pomiaru (powinien on nastąpić w jak najkrótszym czasie od momentu pobrania próbki). Drugim czynnikiem jest nieinwazyjność pomiaru (nie powinien powodować uszkodzenia próbki). Ostatnią bardzo ważną cechą pomiarów jest ich jakość. W przypadku pracy z materiałem biologicznym jest to kryterium niezwykle wrażliwe. Zły wynik, w najlepszym przypadku, może spowodować opóźnienie w analizie do czasu znalezienia wyniku poprawnego, co może z kolei stać się niemożliwe, gdyż do tego czasu próbka materiału biologicznego może stracić swoje właściwości i ulec bezpowrotnemu zniszczeniu. W najgorszych przypadkach niepoprawny wynik może zaważyć na czymś zdrowiu bądź życiu.

Metody, które spełniają określone wyżej warunki pomiaru są bardzo ograniczone. Do tychże należą analizy czynnikowe (ich wadą jest degradacja próbki poddanej badaniu, niemożliwe jest wykonanie innych analiz), metody pomiaru optycznego w tym: metody UV czy rozpraszania spektroskopowego [1]. Z listy kilku metod, ta praca dyplomowa poświęca uwagę metodom pomiaru optycznego, w szczególności metodom oceny próbek w różnych widmach promieniowania elektromagnetycznego.

Podstawową znaną metodą oceny składu próbki jest metoda badania absorpcji. Metoda ta polega na napromieniowaniu próbki światłem koherentnym, takim jak światło lasera. Wykorzystuje

ona sprzęt pozwalający rejestrować promieniowanie, jest to dla przykładu podstawowy spektrometr lub kamera z szerokim kątem widzenia. Otrzymane wyniki badań są jednak niepełne, ponieważ bardzo trudno jest uzyskać informację o składzie procentowym substancji organicznej. Metoda, która pozwala na przebadanie próbek pod kątem ich procentowego składu to metoda pomiaru widma ramanowskiego próbek pobudzonej promieniowaniem laserowym. Widmo pochodzi bezpośrednio od próbki i jest kierowane na układ optyczny spektrometru. Pasma wykorzystywane w metodach omówionych w niniejszej pracy to najczęściej pasmo stokesowskie oraz pasmo rozpraszania Rayleigha. Metoda optyczna, polegająca na badaniu próbek organicznych za pomocą pomiaru promieniowania rozpraszania tych próbek, jest w ocenie autora pracy najodpowiedniejszą metodą, którą można wybrać z dostępnych metod czynnikowych i optycznych do takiego badania. Spełnia ona kryteria wymienione wyżej, ma wysokie szanse na rozwiązanie problemu pomiaru jaki został przedstawiony.

W czasie badań dwa pierwsze warunki wymienione w poprzednim akapicie zostały spełnione (warunek czasu pomiaru oraz jego nieinwazyjności). Pomiar wykonywany za pomocą kamery CCD, która została użyta w badaniu, był bardzo szybki, a sama metoda po dobraniu odpowiedniej (bezpiecznej) mocy lasera nie niszczy próbek. Pojemnik, w którym badane były próbki, pozwalał dodatkowo opóźnić ich niszczenie. Metoda rokuje także w kwestiach mobilności, ponieważ kamera oraz laser mogą być odpowiednio zminiaturyzowane tak, aby badanie mogło się odbywać poza laboratorium, na przykład na miejscu przestępstwa lub w pomieszczeniu, gdzie podawany jest dożylnie lek (na przykład w domu pacjenta). Podstawowym problemem, który pojawił się w czasie badań było uzyskanie odpowiedniej jakości pomiarów.

Teoretyczne rozważania umożliwiają wykorzystanie spektroskopii ramanowskiej do analizy skomplikowanych związków organicznych, niestety realizacja praktyczna wymagająca inżynierskich przybliżeń nie jest łatwa w implementacji. Pojawiają się dwa problemy związane bezpośrednio z metodą pomiaru widma Ramanowskiego. Próbki wykazują duże właściwości świecenia po pobudzeniu laserem, tak zwaną fluorescencję, która będzie omówiona w kolejnych rozdziałach pracy. Fluorescencja jest zjawiskiem nieliniowym co bardzo utrudnia interpretację wyników. Drugim problemem okazało się skomplikowanie widm ramanowskich. Materiał pochodzenia biologicznego, szczególnie w przypadku tkanek, jest najczęściej mieszaniną wielu substancji, z których każda ma inne widmo rozpraszania ramanowskiego. Wszystkie widma przy pomiarze mieszaniny nakładają się na siebie, co bardzo utrudnia rozpoznanie poszczególnych składników. Bardzo trudne jest poznanie składu chemicznego na podstawie wyników badań. Próbki udało się od siebie odróżnić.

2. PLAN I CEL PRACY

Niniejsza praca opisuje zadania, które zostały zrealizowane podczas jej realizacji. Na całość projektu dyplomowego składają się 4 etapy, które są opisane w dalszej części. Każdy z wymienionych etapów potrzebował przygotowania podstawy teoretycznej, która była fundamentem niezbędnym do dalszych badań i rozwijania projektu. Podstawową wiedzą potrzebną do realizacji były informacje i teorie związane z kluczowymi zagadnieniami mechaniki kwantowej. Często można było zastosować przybliżenia wynikające z mechaniki klasycznej, jednak okoliczności w jakich można było je zastosować zostały zaczerpnięte z załączonej w bibliografii literaturze. Studia nad teoretycznymi aspektami badanych i analizowanych zjawisk oraz analiza fundamentu matematycznego, który stał za rozwiązaniami metod klasyfikacyjnych były niezbędne, aby poznać charakter zjawisk i rozpocząć ich cyfrową analizę. Studia nad teorią pochodzenia zjawisk pozwoliły lepiej je zrozumieć. Niemożliwym było by opracowanie metody przygotowania próbek do klasyfikacji bez tego fundamentu wiedzy. Kluczowe informacje jakie budowały postawę teoretyczną niniejszej pracy dotyczyły głównie niskoenergetycznego rozpraszania Rayleigha i Ramana.

Bardzo ważnym aspektem teoretycznym, wręcz fundamentalnym na jakim opierała się niniejsza praca, była teoria stojąca za maszyną działającą jak klasyfikator, maszyną wektorów nośnych oraz analizą głównych składowych. Wtyczka, która była wykorzystywana podczas badań w niniejszej pracy, była przeniesioną koncepcją teoretyczną tych dwóch metod klasyfikacyjnych do środowiska Matlab. Pozwala ona na porównywanie macierzy z danymi, jak również pozwala klasyfikować próbki w takiej macierzy zawarte. Podczas wykorzystywania Toolbox-a LSSVM podczas analizy danych, okazało się, że ma ona wiele programistycznych uproszczeń, jak choćby ograniczenie wymiarów do przetwarzania do 256. Wynika to pewnie z ograniczeń mocy obliczeniowej, lub innych cech charakterystycznych dla środowiska Matlab (ograniczenie wymiarów macierzy przy przekształceniach).

2.1. Praca z danymi pochodzącymi bezpośrednio ze stanowiska pomiarowego

Stanowisko pomiarowe miało niebagatelny wpływ na dane. Podczas przygotowania pracy pod tytułem "Implementacja kalibracji wielowariantowej do widm ramanowskich", gdzie głównym zadaniem była implementacja próbek oraz przygotowanie stanowiska i wykonanie pomiarów, zdobyto wiedzę pozwalającą na lepszą interpretację widm i napotkanych problemów, aby móc wyselekcjonować właściwe widma, oraz mieć pewność, że zdarzające się błędy nie wynikają z wadliwości pomiarów. W widmach znajdują się również różnego rodzaju zakłócenia, omówione w osobnym rozdziale, które należało brać pod uwagę przy planowaniu eksperymentów z klasyfikatorem maszyny wektorów nośnych, w skrócie SVM(Support Vector Machine).

2.2. Rozwiązanie problemów związanych z przygotowaniem i przetworzeniem danych pomiarowych

Dane pochodzące z aparatury pomiarowej były danymi surowymi przedstawionymi w formacie .asc. Dane można było zaimportować do środowiska Matlab do gotowej macierzy. Pierwsza kolumna danych pomiarowych zawierała numer piksela, z którego pochodził pomiar, kolejne kolumny to były serie odczytów kamery. W komórkach kolumn było zapisane pomierzone natężenie rozpraszania ramanowskiego, przedstawione w postaci wartości całkowitej tego natężenia. Długości fali miały swoje odwzorowanie w pikselach. Przygotowanie danych polegało na wykonaniu operacji transponowania macierzy z pomiarem. Operacja była wymagana do poprawnego działania algorytmu maszyny wektorów nośnych oraz analizy głównych składowych (SVM oraz PCA), który w wierszu potrzebował informacji zawartych w jednym pomiarze. Kolejnym krokiem było przygotowanie tabel z pomiarami jednego typu, w których część danych była używana jako model trenujący, część zaś jako zbiór do rozpoznania.

2.3. Klasyfikowanie danych pomiarowych

Klasyfikowanie danych wejściowych składało się z kilku etapów. Z przygotowanych do klasyfikacji danych należało wybrać te, które pochodziły ze starego lub z nowego systemu. Pliki różniły się diametralnie. Część próbek pochodzi z jednego długiego pomiaru, część z serii bardzo krótkich pomiarów. Utworzenie bazy danych z gotowymi próbkami do przetwarzania umożliwiło wstępną analizę i klasyfikowanie próbek. Celem pierwszych operacji było przetwarzanie jak najbardziej surowych danych bez kompresji PCA. Brak poprawnego klasyfikowania danych oraz duże problemy wydajnościowe wymusiły rozwiązania dodatkowe. Zastosowano kompresję algorytmem PCA, polegającą na wstępnym przetwarzaniu i upraszczaniu zbyt dużej ilości wymiarów w próbkach. Ilość wymiarów to 1024, programistyczne ograniczenia TOOLBOX-a LSSVM to 256 wymiarów. Wstępne przetwarzanie polegało na analizie PCA, która wskazywała najbardziej kluczowe wektory. W kolejnym kroku z pełnego widma ramanowskiego zapisanego w bazie danych wybrano do analizy tylko wektory o najbardziej kluczowym znaczeniu dla przetwarzania w klasyfikatorze. Kolejnym etapem było przetwarzanie widm za pomocą algorytmu SVM. Część wybranych danych służyła do strojenia modelu klasyfikatora. Część służyła do sprawdzenia jego skuteczności.

2.4. Pomiary wykonane do analizy

Baza danych z pomiarami zawiera kilkadziesiąt rekordów z zapisanymi próbkami, podzielonymi w zależności od pochodzenia i mieszaniny. Zawiera również dane służące do kalibracji, dzięki którym można mieć pewność, że algorytm zastosowany do próby klasyfikowania próbek miał logiczną spójność i działał poprawnie. Do analizy pomiarów wybrane zostały dwie metody, liniowa i nieliniowa, omówione później w stosownym rozdziale. Próbki po pomiarach musiały zostać

przeanalizowane pod kątem defektów. Próbki potrafiły zdegradować pod wpływem atmosfery, tak jak w przypadku krwi (zmienić stan skupienia z ciekłego na stały, a praca dotyczyła krwi w stanie ciekłym) albo pod wpływem czynników środowiskowych, tak jak w przypadku szumu promieniowania kosmicznego lub nieszczelności w obudowie. Błąd w pomiarach mógł też być powodowany również przez próbki nie umieszczone na środku miejsca pomiarowego lub niedokładnie wypreparowane próbki.

3. PODSTAWY TEORETYCZNE

3.1. Rozpraszanie Rayleigha oraz Ramana

Promieniowanie elektromagnetyczne padające na molekuly jest przez nie częściowo rozpraszane, na przykład przez oddziaływanie z drgającymi wiązaniami molekularnymi., na Na rozpraszanie składa się między innymi rozpraszanie Rayleigha, a także stokesowskie, oraz antystokesowskie rozpraszanie Ramana. Literatura naukowa przedmiotu jest bogata i zawiera pełen opis teoretyczny zagadnienia. Przykłady takich publikacji zostały przedstawione w bibliografii, niektóre z nich to: [2] [3] [4] [5] [6]. Niniejsza praca dotyczy praktycznego wykorzystania obserwowanego efektu rozpraszania. Podstawy teoretyczne są niezbędne do zbadania wyników badań i praktycznego zastosowania zjawiska.

Spektroskopia ramanowska zajmuje się badaniem i rejestracją widma promieniowania. Promieniowanie Ramanowskie jest rozproszone nieelastycznie [7]. Promieniowanie jest rozproszone, ponieważ zachodzi oddziaływanie pomiędzy indukowanymi w układzie dipolami, a padającą wiązką promieniowania elektromagnetycznego monochromatycznego.

Odkrycie promieniowania nieelastycznego Ramanowskiego, było przyczyną nagrodzenia hinduskiego fizyka Chandrasekhara Venkata Ramana [5]. W czasie badań, które były wykonywane podczas przygotowywania niniejszej pracy, wzory matematyczne oraz abstrakcyjne konstrukcje teoretyczne, zostały zestawione z praktyką. Silny nacisk został położony na aspekt realizacji i weryfikacji teoretycznych założeń badawczych przedstawiony przed przygotowaniem niniejszej publikacji.

Aparat matematyczny oraz zagadnienia teoretyczne mimo, że nie stanowią podstawy zainteresowań pracy badawczej, są niezbędne, aby prawidłowo planować i interpretować wyniki eksperymentów. Po krótko zostaną tutaj omówione najważniejsze elementy konstrukcji teoretycznej, które pozwoliły w odpowiedni sposób zinterpretować wyniki oraz wyciągnąć na ich podstawie odpowiednie wnioski.

3.2. Rozpraszanie promieniowania

Rozpraszanie promieniowania elektromagnetycznego, zwłaszcza w paśmie widzialnym, jest łatwo obserwowalne gołym okiem. Rozpraszaniem widzianym w ten sposób jest niebieski kolor nieba. Czerwony kolor jest rozpraszany przez mieszaninę gazów atmosfery, przez co do ludzkiego oka trafia więcej niebieskiego światła. Zjawisko to możemy wyjaśnić posługując się uproszczonym modelem dostrzegalnego dla oka zbioru molekuł do którego dociera promieniowanie, które po przejściu przez ten zbiór trafia do obserwatora. Promieniowanie składa się z fotonów, które przenoszą energię. Część tej energii jest pochłonięta przez zbiór „w zderzeniach molekuł w postaci ich energii translacyjnej” [5].

W wyniku tego zjawiska fotony nie biegną już do wiązki biegnącej w kierunku obserwatora. Z zasady zachowania energii możemy łatwo wydedukować, że bilans energetyczny jest niekompletny, pozostaje jej składowa zaabsorbowana przez cząsteczkę. Energia ta zostaje wyemitowana w postaci fotonów, jednak w dowolnym kierunku przestrzeni, nie zawsze zgodnym z kierunkiem wiązki padającego promieniowania. Obecność w wiązce promieniowania nielicznych fotonów wyemitowanych przez zbiór molekuł tłumaczy dlaczego obserwator jest w stanie zauważyć zmniejszoną ilość energii w promieniowaniu, które do niego dociera. Możemy więc stwierdzić, że parametry promieniowania jakie obserwujemy będą różne w zależności od kąta obserwacji. Będąc w linii promieniowania padającego będziemy obserwować ubytek promieniowania. Zmieniając zaś kąt obserwacji będziemy obserwować promieniowanie uzupełnione o promieniowanie zaabsorbowane, jednak wyemitowane ponownie przez molekułę z pewnym opóźnieniem.

Zjawisko omówione powyżej nazywa się rozpraszaniem promieniowania. Rozpraszanie możemy podzielić na różne typy w zależności od długości fali: radiowe, mikrofalowe, podczerwieni (IR), obszaru widzialnego i UV, rotacyjne EPR, oscylacyjne (Ramana), elektronowe Ramana [5]. Można wyróżnić 3 typy rozpraszania oscylacyjnego – Rayleigha, stokesowskie oraz antystokesowskie [5]. Jeżeli rozpraszanie ma tę samą częstość ν_0 jak promieniowanie padające jest nazywane rozpraszaniem Rayleigha [7]. Rozpraszanie występuje ~~obok~~ pomiędzy molekułami. Materia molekuł zostaje wzbudzona do wyższego poziomu energetycznego, który dzieli się na rzeczywisty i wirtualny. Po wzbudzeniu molekuły wracają do stanu równowagi, czyli do stanu sprzed wzbudzenia. Wiąże się to jak wspomniano wcześniej z emisją fali o takiej samej częstości jak promieniowanie wzbudzające, czyli wiązki padającego promieniowania. Przechodząc od ogółu do szczegółu, promieniowanie padające wzbudza układ do poziomu energetycznego, który jest oscylacyjny. Poziom ma inną wartość od poziomu początkowego i występuje tylko dla zjawiska Ramana. Dla obserwatora opisanego zjawiska, obserwuje się przesunięcie częstotliwościowe (wyrażone w jednostkach cm^{-1}

Układ zostaje wzbudzony i zaczyna oscylować dzięki energii, która wynosi $\hbar\omega_v \approx 1000 \text{ cm}^{-1}$, dla T równego 300K. kT jest równe około 200 cm^{-1} . Pasma anty-stokesowskie jest pomniejszone dodatkowo o czynnik $\exp\left(\frac{-\hbar\omega_v}{kT}\right)$, który sprawia, że jest mniejsze od pasma stokesowskiego o kilka rzędów wielkości [5]. Praktyka pokazuje, że w badaniach głównie wykorzystywane jest pasmo stokesowskie. Stoi za taką praktyką dość istotny fakt. Wyższy poziom sygnału jest w paśmie o niższej częstotliwości. W literaturze przedmiotu istnieje wiele prac, które zajmują się analizą porównawczą dla pasma stokesowskiego i antystokesowskiego, oraz relacji pomiędzy nimi [7].

3.3. Przesunięcie oscylacyjne w rozpraszaniu Ramana

Po opracowaniu kwantowo-mechanicznej teorii rozpraszania przez Dirac-a [5], która była najbardziej konsekwentna w swoich założeniach, rozpoczęto poszukiwania doświadczalnego

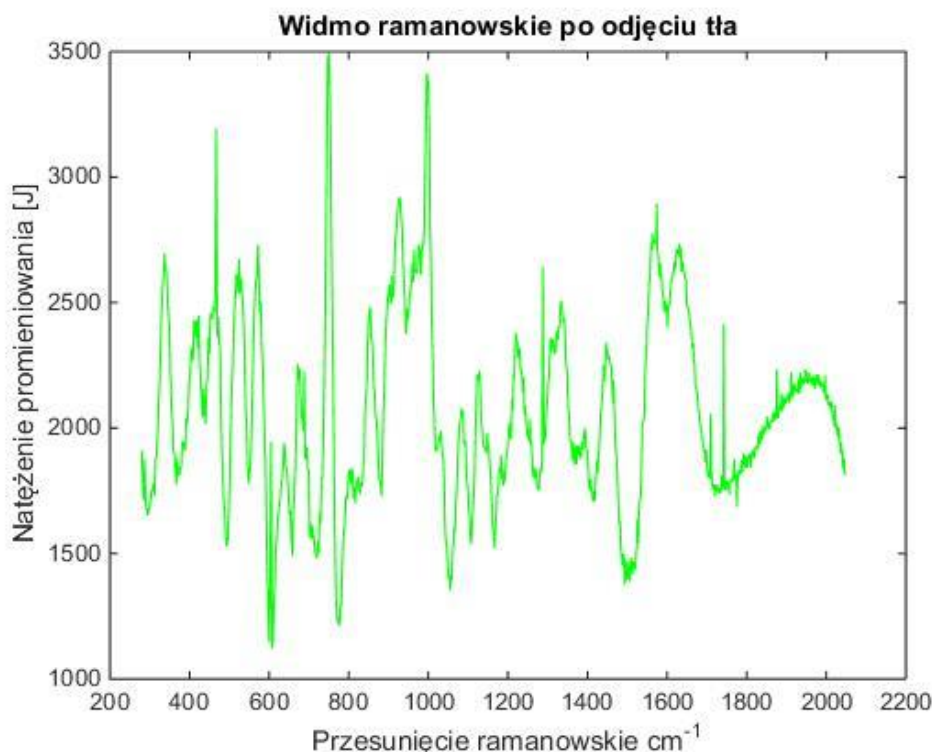
potwierdzenia przewidywań mogących potwierdzić podstawy mechaniki kwantowej [5]. Przewidywania dotyczyły ubytku energetycznego w fali elektromagnetycznej docierającej do obserwatora. Ta energia powinna zostać zamieniona na drgania oscylacyjne w molekułę, przez którą promieniowanie przechodzi. W krótkim czasie udało się zaplanować i przeprowadzić eksperymenty potwierdzające przewidywania, ponieważ dwa niezależne zespoły potwierdziły istnienie tego zjawiska w kryształach kwarcu, odkryto wiązek pomiędzy rozpraszaniem, a częstotliwością drgań kryształu. Odkrycie zaowocowało nagrodą Nobla m.in. dla Ramana, który eksperymentalnie potwierdził jego istnienie w benzenie.

Częstotliwość drgania oscylatora może być większa nieznacznie od częstości padania. W takim wypadku promieniowanie nazywa się promieniowaniem antystokesowskim. Jeżeli częstotliwość jest mniejsza to jest to promieniowanie stokesowskie. Pasma stokesowskie otrzymało taką nazwę ze względu na analogię z regułą Stokesa obowiązującą we fluorescencji [5]. Reguła mówi, że długość fali promieniowania fluorescencyjnego jest zawsze większa od długości promieniowania, które to wzbudzenie powoduje. W wypadku, kiedy molekuła była wcześniej wzbudzona (przed zejściem kolejnego rozpraszania), z dużym prawdopodobieństwem możemy przypuszczać, że oddziaływanie przeniesie ją na podstawowy poziom oscylacyjny [5]. Jeżeli dokonamy pomiaru, otrzymamy w widmie ramanowskie pasmo częstości drgań, jednak przesunięte w przeciwną stronę niż pasmo stokesowskie. Nazwa dla tego promieniowania to promieniowanie antystokesowskie. Nazwa bierze się ze sprzeczności z regułą przesunięcia Stokesa. We wzorach 3.1 oraz 3.2 opisano jak wyliczyć częstotliwość drgań oraz jak się mają do częstotliwości promieniowania padającego pasma stokesowskie oraz antystokesowskie.

$$\bar{\nu} = \bar{\nu}_0 - \bar{\nu}_{st} = \frac{\nu_0}{c} - \frac{\nu_{st}}{c} = \frac{1}{\lambda_0} - \frac{1}{\lambda_{st}} \quad (3.1)$$

$$\bar{\nu} = \bar{\nu}_{ast} - \bar{\nu}_0 = \frac{\nu_{ast}}{c} - \frac{\nu_0}{c} = \frac{1}{\lambda_{ast}} - \frac{1}{\lambda_0} \quad (3.2)$$

Promieniowanie Ramana jest unikalne dla każdego rodzaju wiązań atomowych oraz dla każdego atomu w zależności od jego poziomu energetycznego. Właśnie ta właściwość tutaj omówiona jest głównym powodem, dlaczego promieniowanie ramanowskie jest nazywane odciskiem palca zbioru molekuł danej substancji. Stan fizyczny w jakim znajduje się substancja predestynuje poziom lub rodzaj widma promieniowania Ramana. Stan fizyczny obserwowany w tym promieniowaniu może pochodzić od drgań termicznych, drgań całej cząsteczki lub od drgania wiązania. Powoduje to możliwość rozpoznania substancji po zbadaniu rodzaju jej drgań. Przykładowe widmo ramanowskie w pasmie stokesowskim zostało przedstawione na wykresie 1.

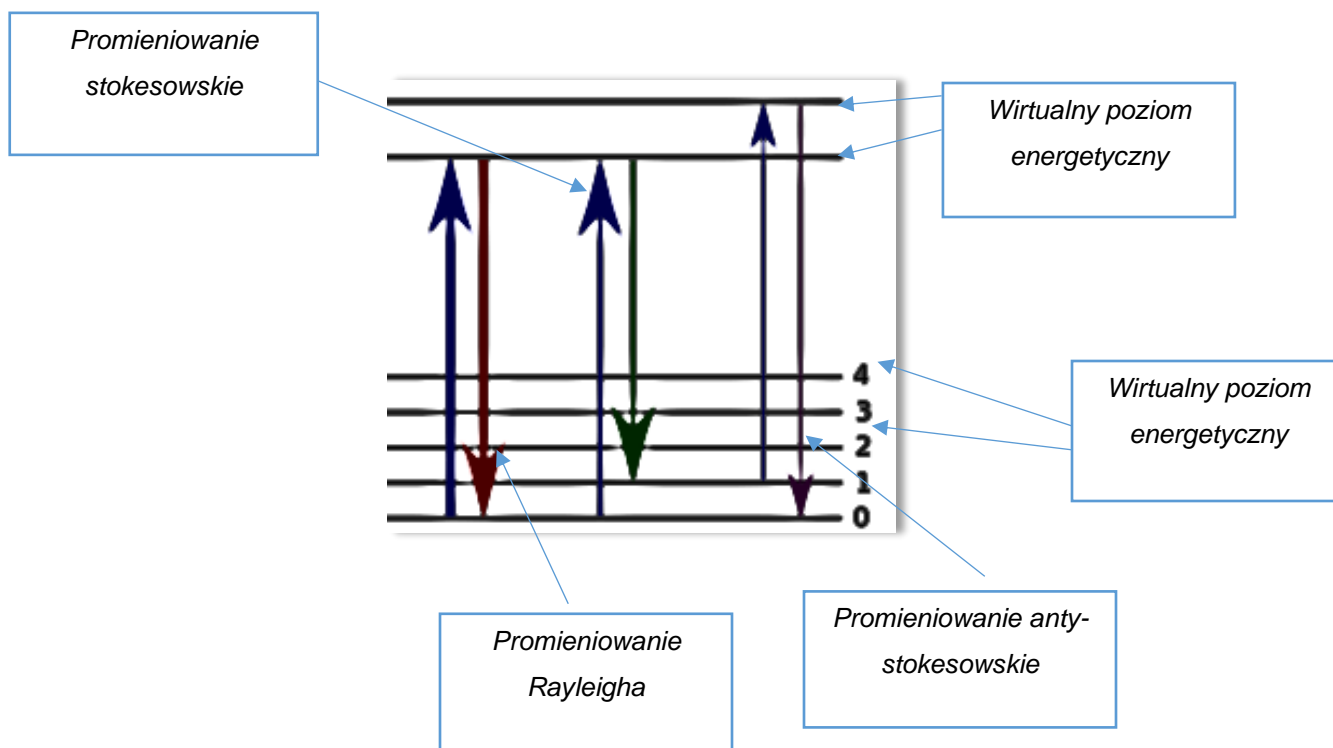


Wykres 1. Przykładowe widmo ramanowskie w pasmie stokesowskim. Badaną próbką jest krew człowieka

3.4. Promieniowanie Ramanowskie w ujęciu mechaniki klasycznej

Jeżeli zastosujemy model atomu zaproponowany przez Bohra, możemy wytłumaczyć zjawisko rozpraszania przy wykorzystaniu mechaniki klasycznej. W takim ujęciu fotony padające z wiązki promieniowania są anihilowane. Foton o zmienionej energii jest kreowany w wyniku przebiegu zjawiska rozpraszania ramanowskiego [1]. Stan energetyczny cząsteczki może się zmieniać. Stany te dzieli się na rotacyjny, oscylacyjny i rotacyjno-oscyłacyjny. W mechanice kwantowej ujęte jest to tak: „istnieje prawdopodobieństwo zmiany stanu energetycznego” [5]. Przejście pomiędzy stanami w ujęciu mechaniki klasycznej jest po prostu możliwe. Dalej rozpatrując ujęcie klasyczne, możliwa jest zmiana energii albo pozostanie w stanie energetycznym pierwotnym. Zderzenie fotonu zaś powoduje zmianę energii tego fotonu. Równanie opisujące prawo zachowania energii jest przedstawione poniżej (wzór 3.3). Schemat wzbudzenia promieniowania został przedstawiony na rysunku 3.1. We wzorze (3.3) opisane zostało przekształcenie energii cząsteczki po przejściu przez nią promieniowania. Możemy zauważyć, że bilans energetyczny równania po lewej i prawej stronie musi się zgadzać.

$$h\nu_o + M(E_i) \rightarrow h\nu_r + M(E_f) \quad (3.3)$$



Rys 3.1 Rysunek ilustruje wirtualne stany energetyczne i przejście pomiędzy nimi cząsteczek.

Natężenie Ramanowskie w wzorach.

$$I = I_0 \cdot a \cdot \hat{c} \cdot l \quad (3.4)$$

Wzór (3.4) prezentujący natężenie sygnału ramanowskiego I dla danej długości fali wzbudzającej λ_0 [5]. We wzorze występuje zmienna a , która wymaga dodatkowego komentarza. Jak wyjaśniono wcześniej w zestawie skrótów, jest to współczynnik prawdopodobieństwa przejścia molekuly pomiędzy stanami oscylacyjnymi. Wielkość ta jest uzależniona od geometrii przebiegu promieniowania w badanym obszarze. We współczynniku uwzględnione jest również odbicie na granicy ośrodków i załamanie fali elektromagnetycznej. W wyniku przeprowadzania pomiarów porównawczych współczynnik zostaje wyeliminowany, wzór staje się prostszy, a wynik łatwiejszy w analizie. Jeżeli oscylator drga, we współczynniku najbardziej kluczowa jest zależność zmian polaryzowalności alfa centrum aktywnego polaryzatora.

$$a_i \sim \left(\frac{\partial \alpha}{\partial q_i} \right)_{q=0}^2 \quad (3.5)$$

Wzór (3.5) opisuje zmiany polaryzowalności cząsteczki badanej pod kątem rozpraszania Ramana.

We wzorze, kolejno, a oznacza polaryzowalność centrum oscylatora, zaś q_i wychylenie oscylatora ze stanu równowagi. Opis oscylacyjnego zjawiska Ramana został zaproponowany przez Placzka. Opis ten jest podawany w licznych opracowaniach promieniowania Ramanowskiego [7] [2] [5].

$$p = \mu + \alpha E \quad (3.6)$$

Wzór (3.6) opisuje cząsteczkowy moment dipolowy niezbędny do znalezienia częstotliwości drgań oscylatora. We wzorze, kolejno, αE oznacza moment indukowany, μ trwały własny moment dipolowy oraz p oznacza cząsteczkowy moment dipolowy.

3.5. Badanie widm ramanowskich substancji organicznych

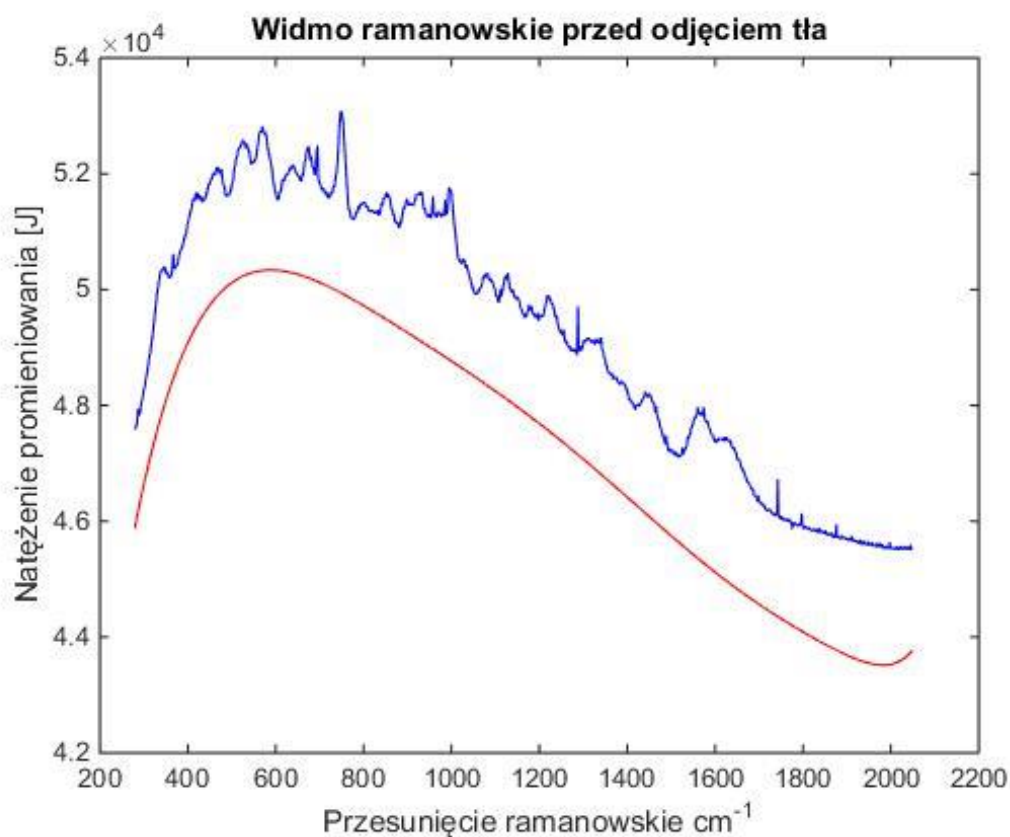
Niniejszy podrozdział skupia się na interpretacji widm ramanowskich. Badania prowadzone nad spektroskopią Ramana widm materiałów są prowadzone od dawna, na rynku są dostępne rozwiązania aplikacyjne, które pozwalają interpretować widma ramanowskie dla niektórych materiałów. Prawdziwym problemem jaki pojawia się w interpretacji widm jest problem z materiałami nietrwałymi, na przykład pochodzenia organicznego [5]. Materiały takie zmieniają swój stan skupienia, lub strukturę pod wpływem działania czynników środowiskowych, takich jak temperatura, gazy atmosferyczne czy promieniowanie UV. Również materiały o złożonej budowie, zmiennym stanie skupienia są niejednokrotnie niemożliwe do analizy. Przykładem materiału, który w zasadzie łączy wszystkie trudności jest krew. Krew ssaków jest skomplikowaną mieszaniną, w skład której wchodzi wszystkie makro i mikroelementy niezbędne do właściwego funkcjonowania organizmu. Krew nie jest jednolita w swojej strukturze. Jest delikatną organiczną substancją, która ulega degradacji w wyniku używania laserów zbyt dużej mocy do zbierania danych. Najtrudniejszą w interpretacji i badaniu właściwością krwi jest zmienny stan skupienia. Krew zasycha poddana na działanie tlenu atmosferycznego w temperaturze pokojowej. We krwi pojawiają się dodatkowe czynniki, które wpływają na wartość przesunięcia Ramana. Czynniki te są skomplikowane relacje pomiędzy substancjami. Krew jest emulsją wymieszanych cieczy o różnej gęstości. Widma

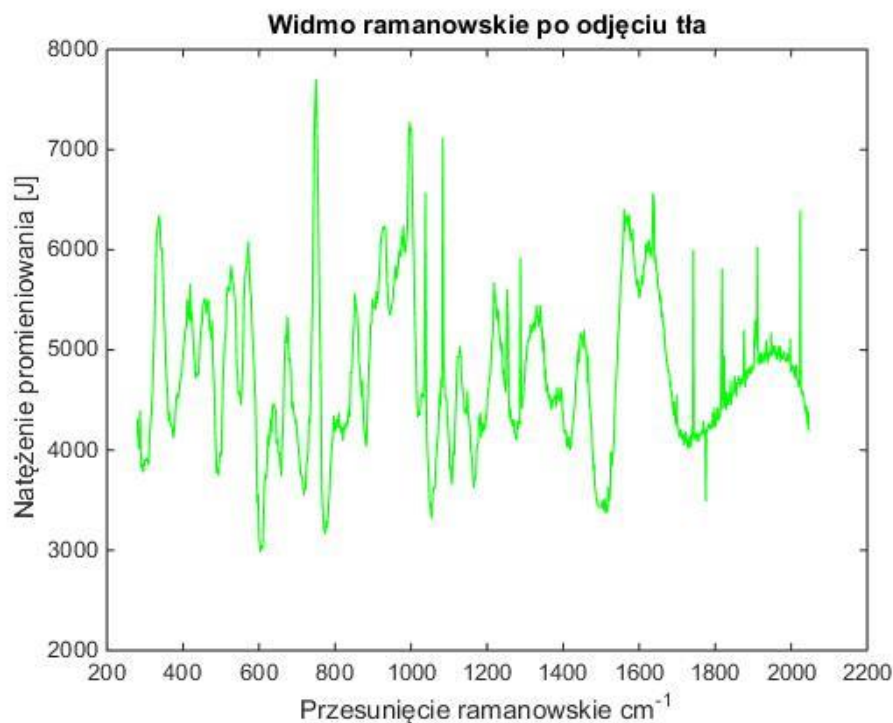
poszczególnych składników krwi dodatkowo nakładają się na siebie. Wartość przesunięcia jest zmieniana, również przez bliską odległość do innych skomplikowanych molekuł [3].

$$I_{st} \sim \left(\frac{d\alpha}{dq} \right)_{q=0}^2 Q^2 E_w^2 (\nu_0 - \nu_R)^4 \quad (3.7)$$

$$I_{ast} \sim \left(\frac{d\alpha}{dq} \right)_{q=0}^2 Q^2 E_w^2 (\nu_0 + \nu_R)^4 \quad (3.8)$$

Wzory (3.7) oraz (3.8) opisują natężenie sygnałów w pasmach stokesowskim i antystokesowskim. We wzorach, kolejno, I_{st} oznacza natężenie sygnału widma ramanowskiego w paśmie stokesowskim, I_{ast} natężenie sygnału widma ramanowskiego w paśmie antystokesowskim, ν_R to częstotliwość drgania normalnego układu atomowego, Q to amplituda drgań normalnych, zaś E_w to amplituda pola elektrycznego fali wzбудzającej.



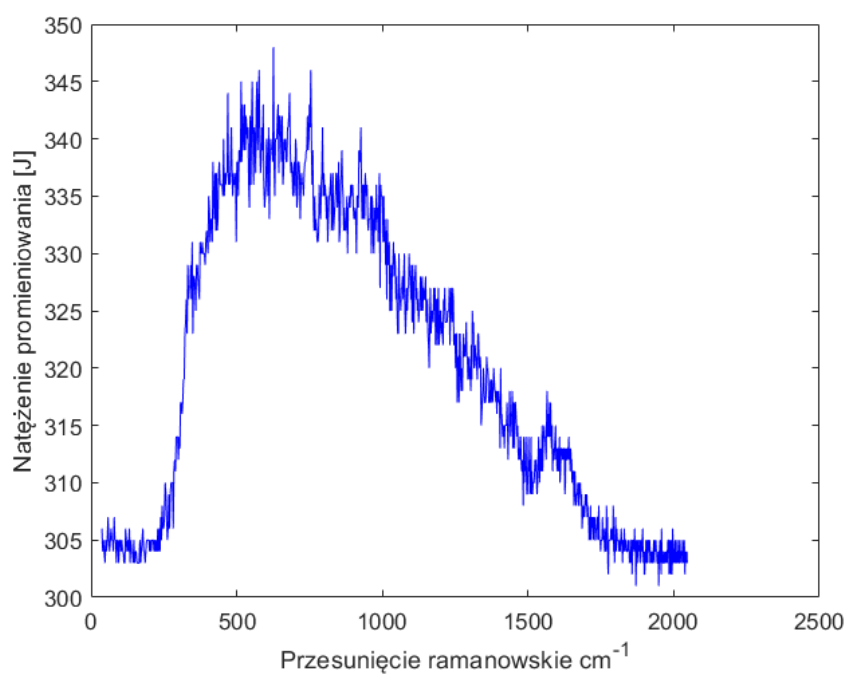


Wykres 2. i 3. Widmo glukozy z fluorescencją i odjętym tłem. Dla przesunięcia od 1800 cm^{-1} występują szумы. Są pozostałością niedokładnego odejmowania tła

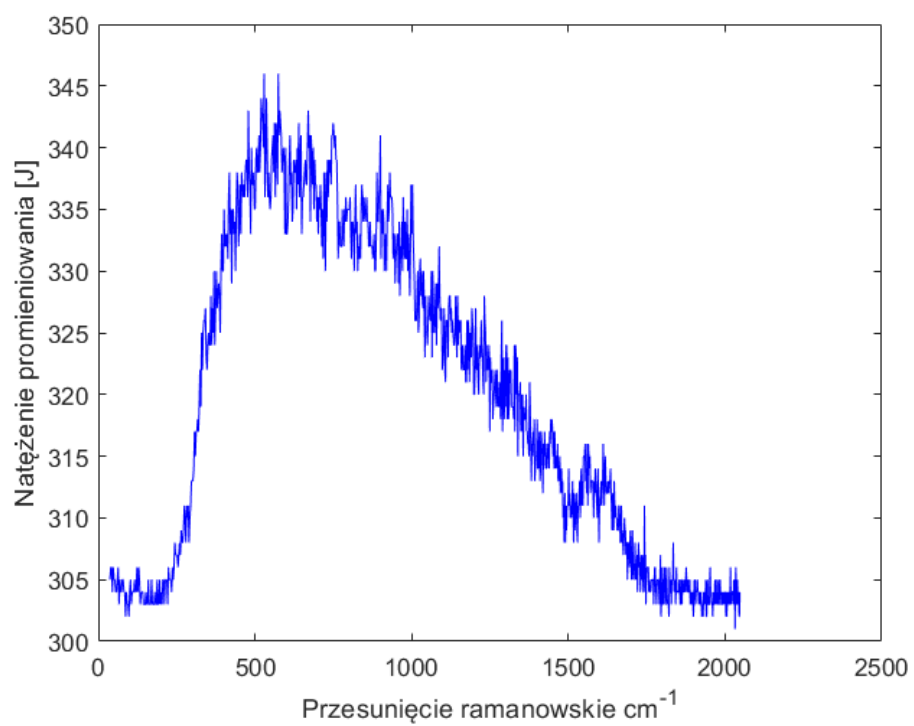
Szersze pasmo widma ramanowskiego jest uzyskiwane dzięki promieniowaniu elektromagnetycznemu o jak najwęższym paśmie. Idealnym źródłem wzbudzającym jest światło laserowe. Duża koherentność wiązki powoduje niezakłócenie pomiarów, dodatkowo łatwo można odfiltrowywać deformacje powstające w detektorze. Światło laserowe padające na próbki ma niską intensywność linii widmowych.

3.6. Przykładowe widma ramanowskie uzyskane w pomiarach

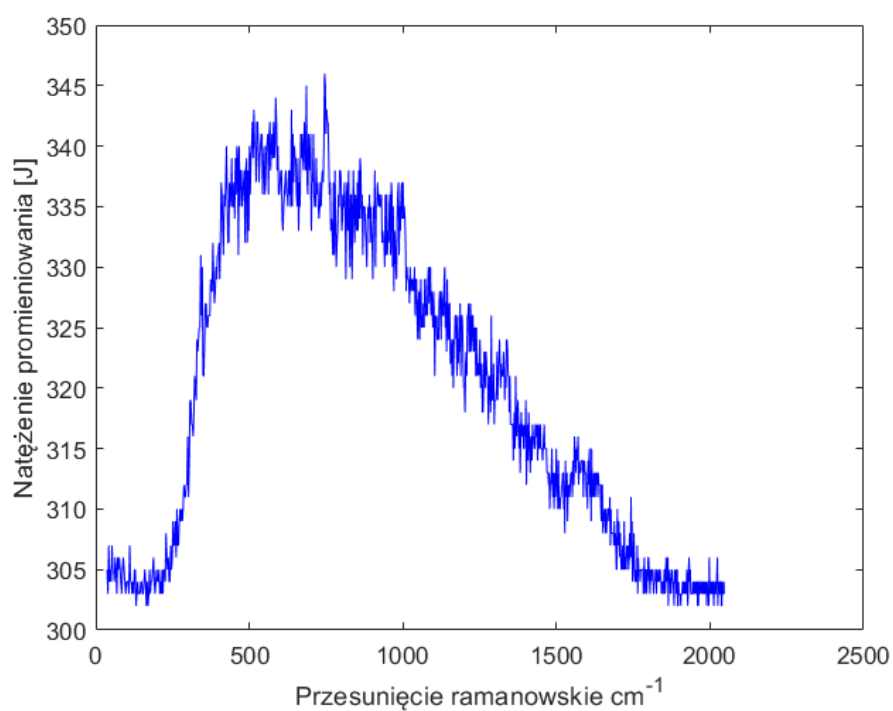
Wykresy od 4 do 11 przedstawiają pomierzone widma ramanowskie w pasmie stokesowskim dla różnych substancji organicznych (alkoholu, glukozy z hydroxapatytem, krwi psiej, krwi ludzkiej, propofolu). Na wykresach widać, że poszczególne widma różnią się w subtelny sposób jednak algorytm klasyfikujący doskonale radzi sobie z rozróżnianiem. Dane przedstawione na wykresach są po kompresji za pomocą algorytmu PCA.



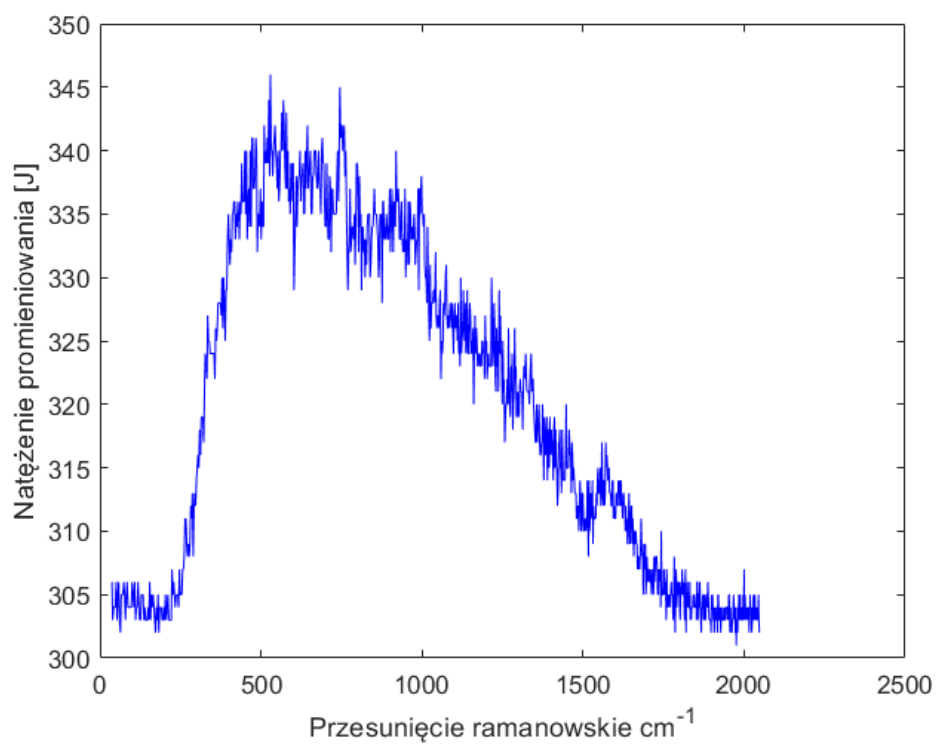
Wykres 4. Widmo ramanowskie pustej podstawki



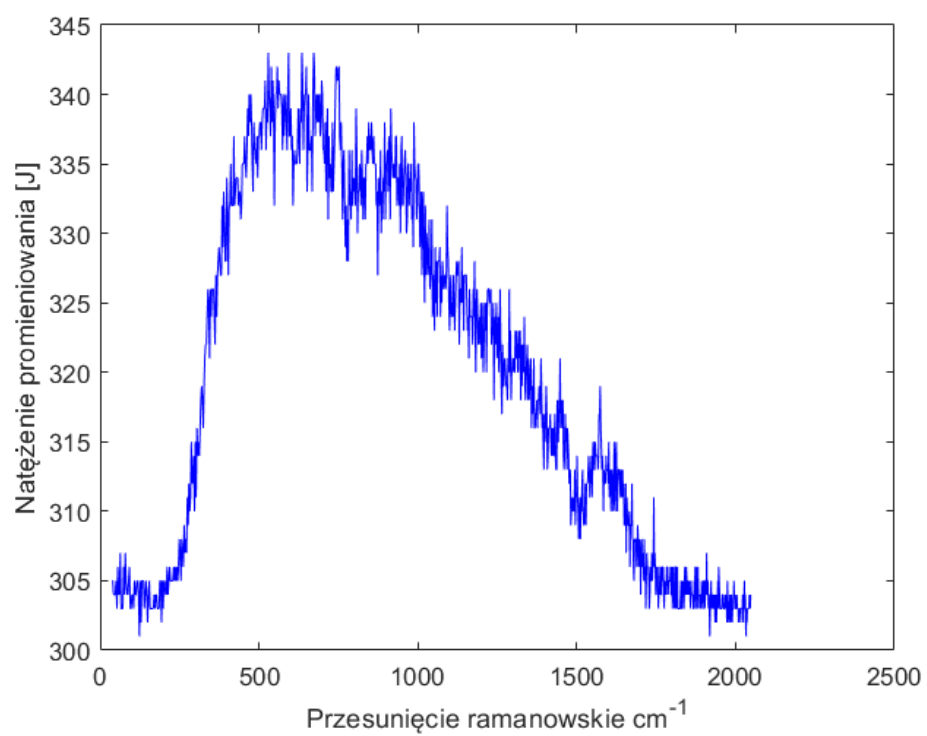
Wykres 5. Widmo ramanowskie alkoholu



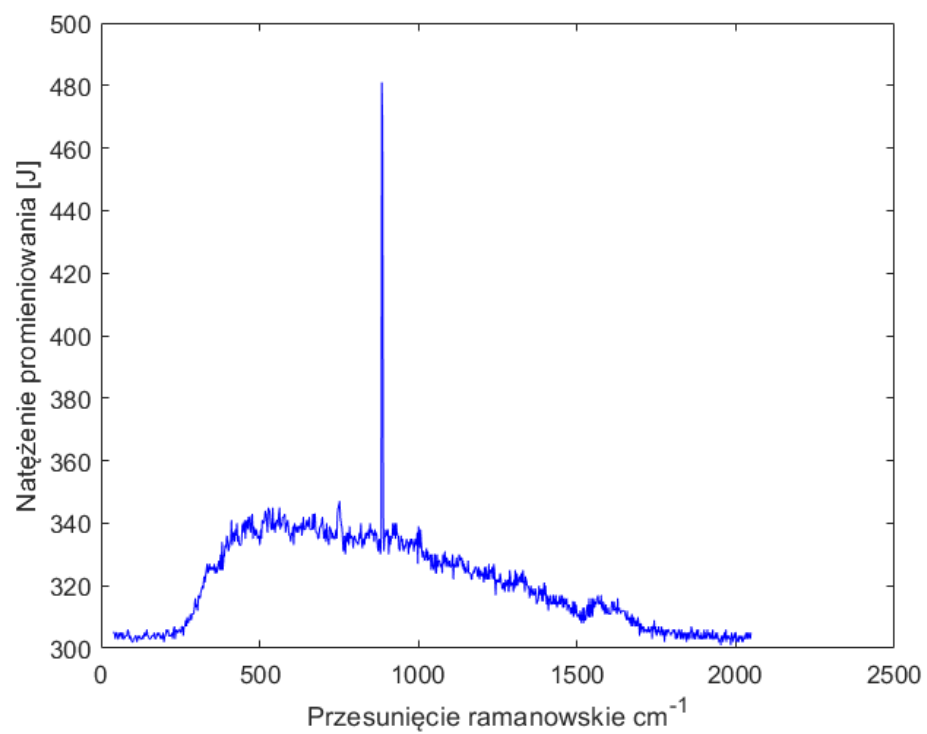
Wykres 6. Widmo ramanowskie glukozy z hydroxapatytem



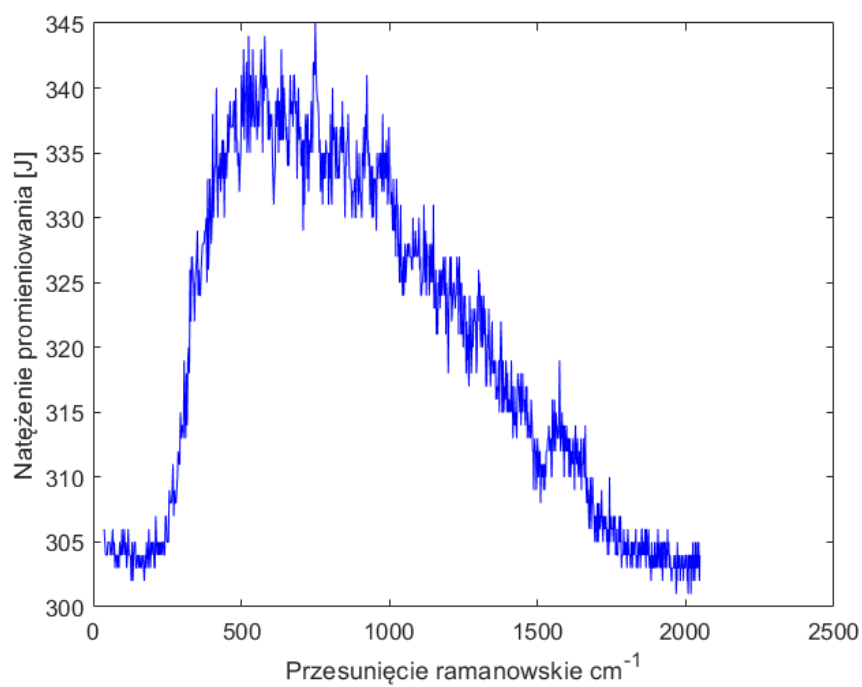
Wykres 7. Widmo ramanowskie psiej krwi 666079



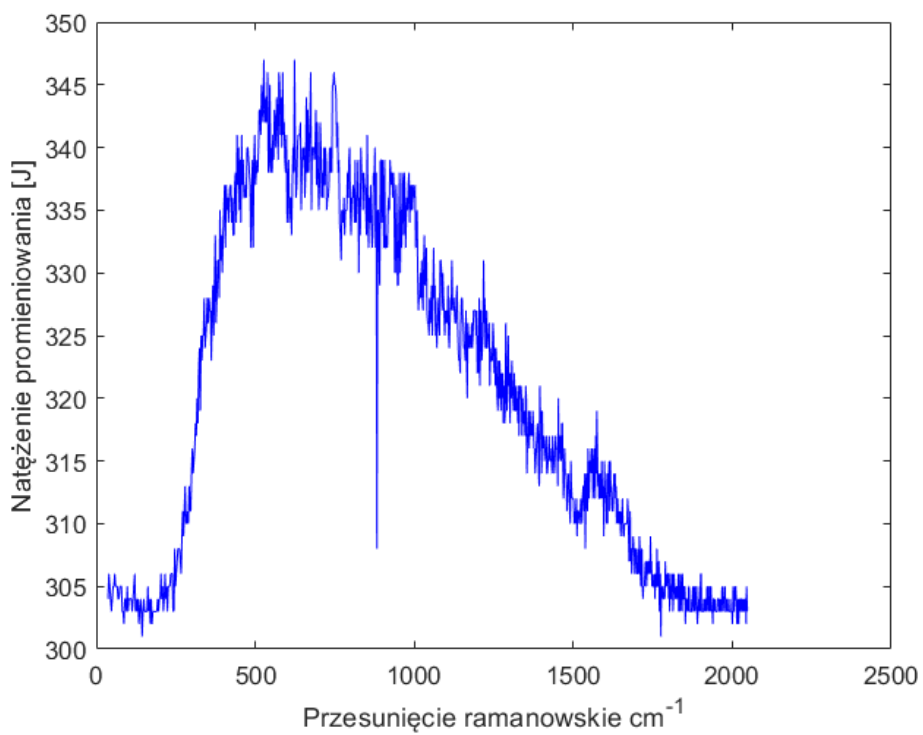
Wykres 8. Widmo ramanowskie psiej krwi 664089



Wykres 9. Widmo ramanowskie propofolu z szumem losowym



Wykres 10. Widmo ramanowskie ludzkiej krwi



Wykres 11. Widmo ramanowskie psiej krwi z szumem losowym.

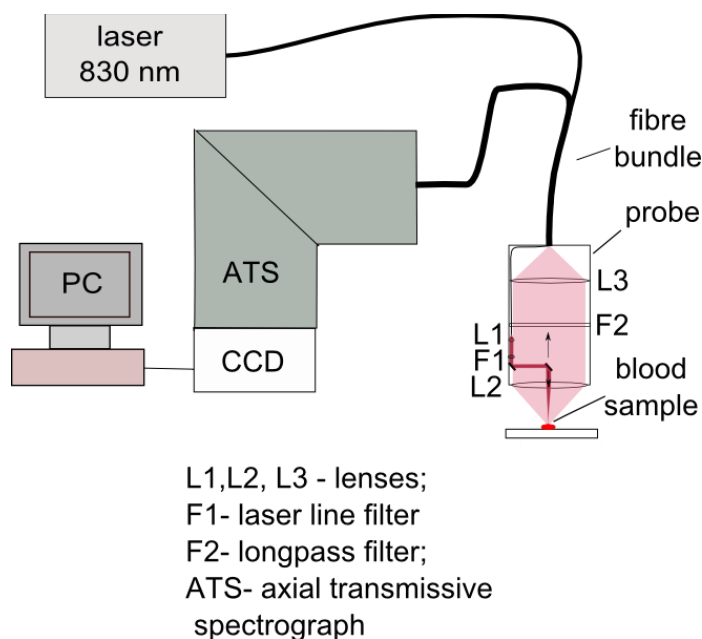
3.7. Identyfikacja problemu

Problem jaki został postawiony przed autorem pracy był następujący. Sprawdzić czy z wykorzystaniem dostępnych narzędzi matematycznych można interpretować skomplikowane widma krwi. Widma są nie tylko niezwykle skomplikowane w całym badanym paśmie, również, ze względu na małą różnicę w składzie krwi pomiędzy poszczególnymi dawcami, są ciężkie do odróżnienia. Różnice często były wielkości jedności. Nieuchwytnie gołym okiem na wykresie, jedynie przy analizie czystych danych pochodzących bezpośrednio z układu pomiarowego. Jednak drobne różnice nie pozwalały stworzyć modelu na podstawie tych obserwacji. Surowe dane nie są przystępne do przetwarzania przez człowieka. Operacje matematyczne mogące poprawić wyświetlanie na wykresie polegają na przetwarzaniu macierzy z danymi. Macierze z nami są bardzo trudne w analizie. Dlatego po przeprowadzeniu niezbędnych przekształceń, poklasyfikowane dane są wyświetlane w odpowiedniej dla ludzi formie wykresu.

4. OPIS UKŁADU POMIAROWEGO ORAZ BADANYCH PRÓBEK

4.1. Układ pomiarowy

Układ pomiarowy składał się z modułów połączonych ze sobą za pomocą przewodów światłowodowych. Poszczególne bloki układu oraz jego niezależne części miały dopasowane parametry optyczne oraz fizyczne w celu poprawnego działania [8]. Poglądowy rysunek układu pomiarowego został przedstawiony na rysunku 4.1. System, wykorzystujący laser diodowy o długości fali 830 nm skonfigurowany był w dwóch wersjach (1 - wcześniejsza) ze spektrometrem, sondą i kamerą CCD zestawu VTT Ramstas (Finlandia), (2- późniejsza) ze spektrometrem i sondą z VTT Ramstas oraz kamerą CCD Andor DV-401-BV.



Rys 4.1. Rysunek poglądowy układu pomiarowego wykorzystanego do przeprowadzenia badań.

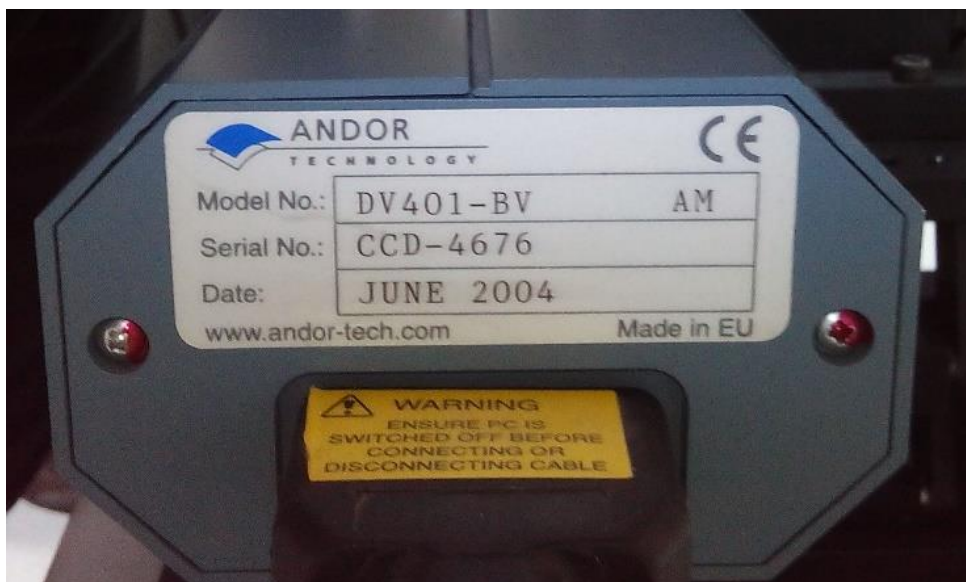
Najważniejszymi elementami układu pomiarowego były:

- Laser
- Specjalna zamknięta przestrzeń z detektorem
- Stanowisko komputerowe.

Zdjęcia 4.1 oraz 4.2 przedstawiają laser i detektor użyty w pomiarach.



Zdjęcie 4.1. Laser użyty w pomiarach.



Zdjęcie 4.2 Detektor promieniowania

Stanowisko komputerowe pozwalało na przetwarzanie oraz zapis pomiaru. Umożliwiało ono również wstępną analizę widma (w celu początkowej oceny poprawności badania). Analiza była możliwa, ponieważ można było wyświetlić widmo na ekranie komputera. Oprogramowanie użyte do wyświetlania wyników eksperymentu jest dostarczane od producenta. Komercyjne rozwiązanie softwarowe nie zostało wykorzystane w niniejszym projekcie. Praca interesuje się rozwiązaniami open softwarowymi. Dostarcza je producent kamery detekcyjnej, nazwa to Andor.mcd®. Za pomocą światłowodu laser został podłączony do filtra oraz rozdzielacza wiązek. Promieniowanie laserowe,

doprowadzone do próbki, która znajduje się w zamkniętym szczelnie korpusie, wzbudza badane molekuly. Wzbudzone zostaje wielokierunkowe rozpraszanie Rayleigha, pasma stokesowskie oraz anty-stokesowskie. Im węższa wiązka promieniowania ramanowskiego tym szersze pasmo otrzymanego widma ramanowskiego.

W zastawie, którym było testowane nasze rozwiązanie, należy wspomnieć również o środowisku Matlab, które miało za zadanie przetwarzać zgromadzone dane. Jest to też element niezbędny, aby uruchomić TOOLBOX LSSVM (Least Squares Support Vector Machines). Podsumowując, układ pomiarowy składa się z dwóch części: fizycznej, gdzie był realizowany pomiar oraz części programistycznej, gdzie pomiar był przetwarzany i interpretowany. [9]

4.2. Badane próbki (krew)

Krew jest rodzajem tkanki łącznej o charakterystycznych właściwościach. Po pierwsze jest mieszaniną kilku płynów, często opisywaną w literaturze jako emulsja. [10] Po drugie jest zamknięta w organizmach żywych i praktycznie odizolowana od środowiska zewnętrznego. Składnikami krwi są płynne substancje pozakomórkowe takie jak: osocze, krwinki czerwone, krwinki białe oraz u ssaków zwane płytkami krwi trombocyty [10]. Cechy te powodują, że ten materiał jest nietrwały, ulega biodegradacji, utlenianiu i rozwarstwianiu na poszczególne składniki mieszaniny.

Dlatego ważne jest odpowiednie przygotowanie próbki do wykonywania badań oraz odpowiednie przechowywanie w niskiej temperaturze, nie powodującej jednak zamarznięcia wody. Przed przystąpieniem do pobrania próbki należy krew rozmieszać. Jeśli próbka trwale zmieniła kolor lub stan skupienia oznacza to jej nieodwracalną degradację przez co staje się niezdadna do wykorzystania w badaniach.

Często jednak dochodzi do uszkodzenia w czasie pomiarów, których czas wykonania sięga kilku minut. Uszkodzenie to może zostać nie zauważone, przez osobę wykonującą badania. W takim przypadku należałoby w czasie przetwarzania danych, w środowisku Matlab, odpowiednio wykrywać wyniki uszkodzonych próbek i je eliminować. Dlatego powstała rozszerzona wersja algorytmu redukującego szumy, gdzie usuwane są wektory ze średnią wartością komórki, która przekroczyła dwukrotnie średnią wartość komórki dla widma wzorcowego. W przyszłości można byłoby ten proces przekazać do sieci neuronowej, która ucząc się poprawnej wartości sama dokonywałaby korekty. Zdjęcie 4.3 przedstawia próbkę krwi umieszczoną na zaadoptowanej do badań aluminiowej płytce.



Zdjęcie 4.3 Próbkę krwi umieszczona na płytce aluminiowej specjalnie zaadaptowanej do badań.

5. ALGORYTMY STOSOWANE DO PRZETWARZANIA DANYCH

5.1. Zastosowanie algorytmów przetwarzania danych

Dane rejestrowane przez układ pomiarowy mają postać macierzy z danymi. Kamera CCD rejestrująca widmo ramanowskie zapisuje wartości pomierzonego promieniowania w kolejnych pikselach. Pomierzona wartość jest zapisywana w macierzy. Pomiar dla próbki ma więc postać wartości w wierszach. Rozdzielczość matrycy to 1024x300. Oprogramowanie producenta dedykowane do kamery pozwala na zapis z rozszerzeniem zarówno .asc jak również .sif. Rozszerzenie .sif umożliwia sprawdzanie widma w formie wykresów na monitorze sprzętu komputerowego. Program Matlab jak również projekt dyplomowy bazuje na rozszerzeniu .asc, pozwala to łatwe wczytywanie plików oraz ich przetwarzanie. Surowe dane łatwo można poddać przetwarzaniu przez algorytmy TOOLBOX LSSVM.

Część próbek miała uśredniane wartości dla danego rodzaju materiału, jednak ten sposób ich kompresji okazał się niewłaściwy, ze względu na to, że poziom zaszumienia ulegał zwiększeniu zamiast redukcji. Pliki, które były utworzone w czasie pomiarów, trafiały następnie do bazy danych, która pozwalała archiwizować i przetwarzać dane. Archiwizacja próbek pozwala oczywiście używać ich wielokrotnie. Nazwa próbek zarchiwizowanych postępuję konsekwentnie według wzoru:

NumerIdentyfikującyDawcę_LiczbaSekundNaświetlania_IlośćIteracjiDlaDanejPróbki_LiczbaCykliNaświetlania.

Przy wstępnym przetwarzaniu danych korzystano z algorytmu odcinającego szумы pomiarowych. Niestety, w danych występują szумы informacyjne, których pozbycie się jest skomplikowane i opisane w kolejnych rozdziałach.

5.2. Algorytm PCA

Analiza głównych składowych (Principal Component Analysis), w skrócie PCA, ma na celu zmniejszenie liczby zmiennych co pozwala ograniczyć liczbę wymiaru danych. Dokonuje się to poprzez wybranie przestrzeni, która obejmuje tylko wybrane wektory. Te wektory są nazywane głównymi składowymi. Oczywiście przekształcenie jest stratne, czyli traci się część informacji, zawartej w wektorach nie będących branymi pod uwagę. Podstawową zasadą dla tej analizy jest to, aby wektory główne charakteryzowały się jak największą zmiennością. Pomijane są wektory zbyt do siebie podobne.

PCA składa się na serię kroków wykonywanych w określonej kolejności. Na początku zmienne są normalizowane. Wykonuje się to za pomocą metody ZScore polegającej na ustandaryzowaniu wektorów wewnątrz macierzy. W kolejnym kroku obliczamy składowe, które

charakteryzują się największą zmiennością. W tym celu stosuje się macierz kowariancji zmiennych, w naszym wypadku prostopadłych do siebie wektorów. Ortogonalizacja wektorów powoduje, że nie będą one ze sobą skorelowane, co umożliwi wyznaczenie naszej przestrzeni wektorów głównych, niezależnych od siebie. Wykonujemy następnie operację znalezienia wartości własnych macierzy kowariancji i odpowiadających jej wektorów. Sortujemy nasze wektory w kierunku największej wartości. Otrzymujemy posortowane wektory nazywane składowymi głównymi. Ostatnią operacją wykonywaną w algorytmie PCA jest wybór najbardziej znaczących wektorów. Pierwsza składowa będzie zawierała najwięcej informacji ze zbioru, druga składowa z pozostałej części.

Analiza PCA została wykorzystana w niniejszej pracy do zmniejszenia problemu wymiarowości. Widma zostały przeanalizowane pod kątem składowych głównych, a następnie zredukowane.

5.3. Algorytm SVM

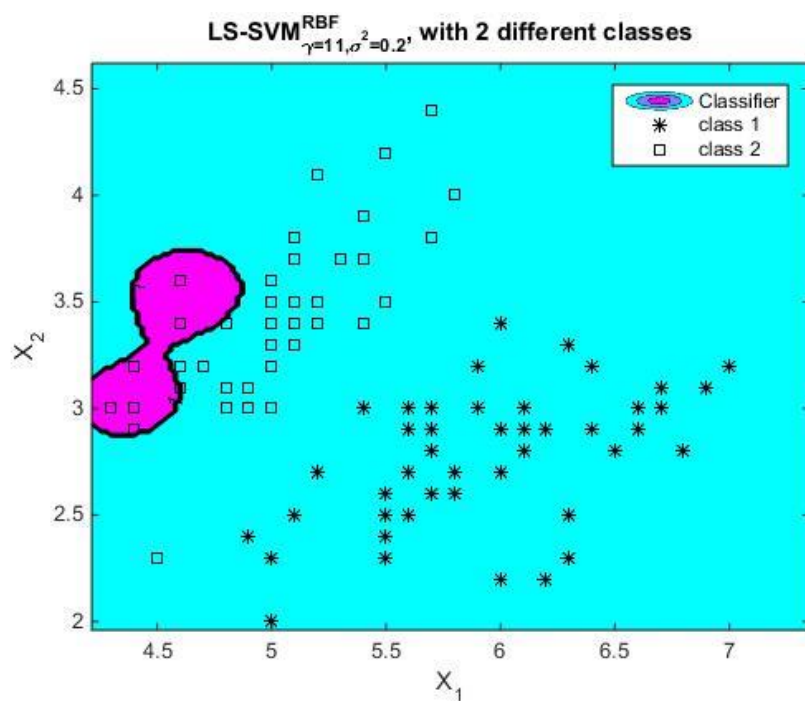
Analiza głównych składowych należy do analizy czynnikowej, która pozwala rozróżniać między sobą poszczególne próbki. Główne składowe macierzy wyznaczone w algorytmie pozwalają grupować dane tam, gdzie różnice są słabo dostrzegalne.

Support Vector Machine, w skrócie SVM potrafi klasyfikować dane dzieląc je na 2 grupy. Ta cecha nosi nazwę binarności algorytmu klasyfikowania. Trenowanie i strojenie maszyny wektorów nośnych polega na znalezieniu optymalnych parametrów prostej, która podzieli zbiór na dwie części. Granica pomiędzy zbiorami nazywana jest marginesem. Znaleziony margines powinien być jak najszerszy. Szeroki margines wyznaczony przy trenowaniu sieci pozwala na zachowanie jak najbardziej uniwersalnego modelu, czulszego, z większą liczbą rodzajów danych. Tak utworzone podzbiory po obu stronach marginesu powinny zawierać tylko przypadki należące do danej klasy. Jeżeli jest to niemożliwe algorytm będzie starał się utworzyć granicę miękką pomiędzy nimi. Każdy przypadek klasy, która znajdzie się po nieodpowiedniej stronie granicy będzie traktowany jako pewnego rodzaju błąd i otrzyma mniejszą wagę w obliczonym modelu. Pozwala to wykorzystać SVM nawet w przypadku, kiedy wyznaczona hiperpłaszczyzna nie pozwoliła rozwiązać problemu, przez zbyt dużą ilość wymiarów. SVM dla przypadków dwuwymiarowych lub mniejszych szuka prostej, która pozwala podzielić grupy na dwa zbiory. Istnieje oczywiście nieskończona ilość prostych, które przechodzą przez punkty. Wyznacza się więc 2 proste dla skrajnych punktów danej klasy, następnie przesuwają się one w kierunku skrajnych punktów klasy drugiej. Otrzymujemy w ten sposób marginesy każdego klasyfikatora. Znaleziona przestrzeń podczas operacji przesuwania marginesów nazywa się największym marginesem. Dla przypadku dwuwymiarowego wyznaczamy prostą, jednak jeżeli będziemy rozważać przypadek n -wymiarowy musimy wyznaczyć hiperpłaszczyznę. Wymiar hiperpłaszczyzny jest o jeden mniejszy od wymiaru przetwarzanych danych. Znaleziona granica w hiperpłaszczyźnie jest przenoszona do dwóch wymiarów, aby zamieścić ją na wykresie. Składa się

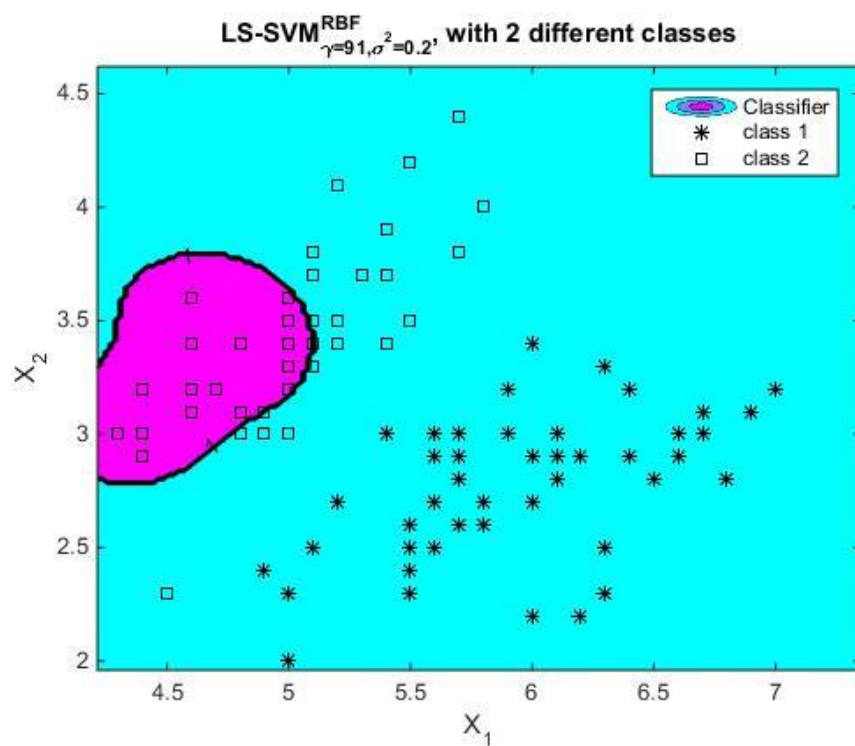
wtedy z prostych, które układają się w łamaną, która stanowi nasz margines maksymalny. Drugim czynnikiem, który pojawia się w bardzo skomplikowanych wielowymiarowych danych, jest obecność punktu po drugiej stronie marginesu niż punkty klasy, do której należy. Stosuje się tu tak zwaną zmienną rozluźniającą [11]. Zmienna rozluźniająca jest czynnikiem, który wpływa na dodanie w modelu obniżonej wagi parametrom punktów, które leżą poza granicą marginesu miękkiego wyznaczonego przez klasyfikator SVM. Margines, który jest wyznaczany do najbliższych położonych punktów, klasyfikuje te punkty jako punkty podparcia. Wektory, które tworzą punkty podparcia noszą nazwę wektorów nośnych, dodatkowo do wektorów w modelu dodaje się wagi ujemne, które stanowią rodzaj kary, dla punktów nie będących po właściwej stronie marginesu. Wspomnianą, już hiperpłaszczyznę wyznacza się z powodu twierdzenia Covera. Twierdzenie to mówi o tym, że przeniesienie problemu do większej ilości wymiarów pozwoli znaleźć jego rozwiązanie liniowe. Będzie można ustawić granicę pomiędzy klasami, jednak w większej ilości współrzędnych. Następnie rozwiązany problem trzeba przenieść do dwóch wymiarów. Transformowanie jest realizowane przez funkcję jądra. Zastosowanie kernel (funkcji jądra) prowadzi do większego skompilowania struktury naszych danych. Jest to jednak tylko pozorne, ponieważ przekleństwo wymiarów jest tutaj pomijalne [12]. Dzieje się tak dlatego, że do wyznaczenia marginesu używamy tylko wektorów nośnych, wszystkie inne są pomijane, znajdują się wewnątrz odgradzanego zbioru. Algorytm, który stroi model SVM do klasyfikacji na początku nie wie, ile wymiarów będzie potrzebne, żeby znaleźć rozwiązanie. Często pojawia się tutaj ograniczenie liczby wymiarów, tak jak przy LSSVM. Maksymalna liczba wymiarów wynikająca z dokumentacji to 256. Dane pochodzące z analizy ramanowskiej przebadanych próbek posiadają 1024 wymiary. SVM została wybrana do przetwarzania skomplikowanych widm ramanowskich ze względu na właściwości odporności na szумы informacyjne, jest powszechnie stosowana w bio-informatyce.

5.4. LSSVM (*Least Squares Support Vector Machines*)

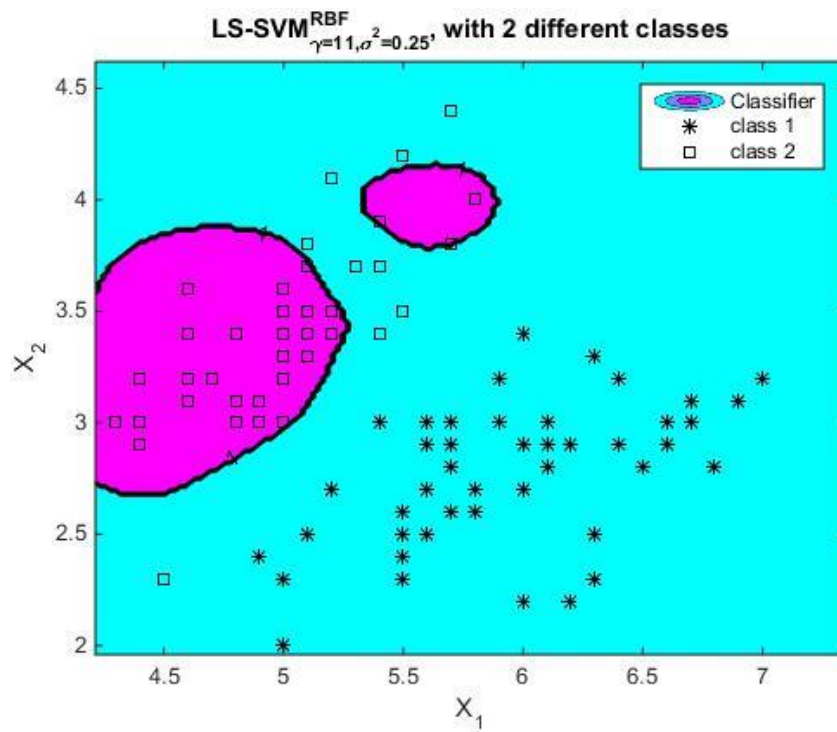
Metoda maszyny wektorów nośnych (SVM) w przypadku Toolbox opiera się na znalezieniu współczynników prostej w modelu - α i b . Współczynniki pozwalają na rozmieszczenie danych dla modelu na wykresie w taki sposób, żeby poprowadzić optymalną granicę pomiędzy klasami. Przekształcenia w $n-1$ wymiarowej przestrzeni, gdzie n jest liczbą wymiarów danych, z których składają się klasy, jest wysoce nieprzystępne dla człowieka. Toolbox jest udostępniony za darmo na licencji MIT [12]. Na wykresach 12,13,14 oraz 15 zaprezentowano strojenie parametrów γ i σ (szerokości marginesu) metodą nieliniową Kernel_RGB.



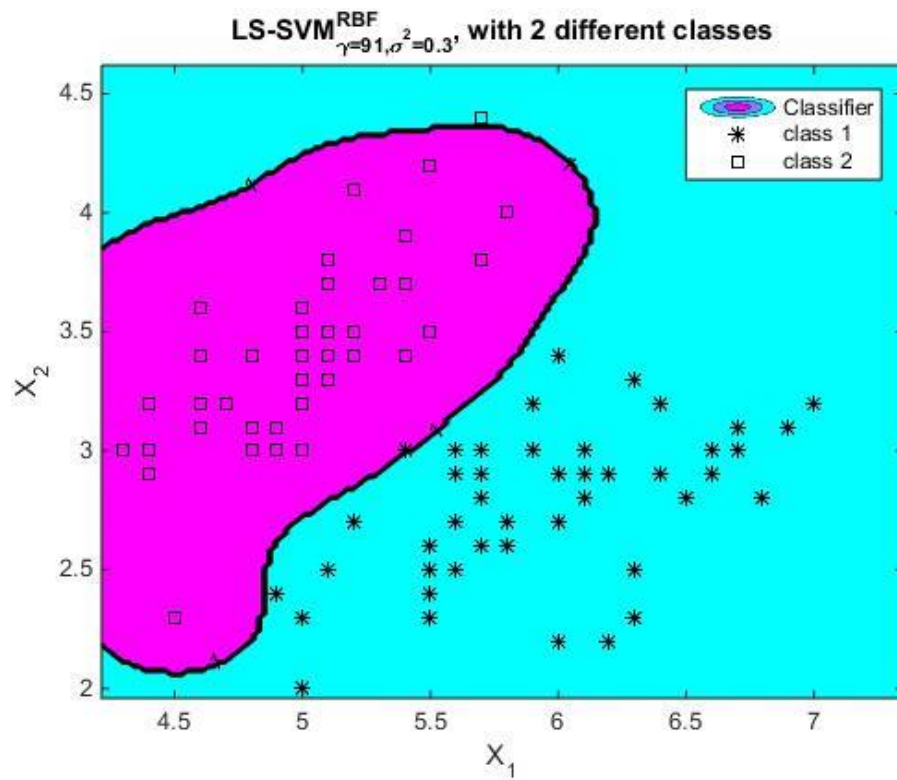
Wykres 12. Metoda nieliniowa Kernel przy współczynnikach $\gamma=11$ $\sigma^2=0.2$



Wykres 13. Metoda nieliniowa Kernel przy współczynnikach $\gamma=91$ $\sigma^2=0.2$



Wykres 14. Metoda nieliniowa Kernel przy współczynnikach $\gamma=11$ $\sigma^2=0.25$



Wykres 15. Metoda nieliniowa Kernel przy współczynnikach $\gamma=11$ $\sigma^2=0.2$

6. DANE DO KLASYFIKACJI

6.1. Dane do strojenia algorytmu

Dane, których użyto do strojenia algorytmu zaczerpnięto z bazy danych irsi.data.set [13]. Jest to znakomicie pomierzona i poklasyfikowana baza danych wielkości lrysów różnych gatunków tego samego rodzaju rośliny. Poprawny rozkład bazy danych dla klasyfikatora jest znany, dlatego można na podstawie wyników dobrać najważniejsze parametry (gamma, sigma) algorytmów przetwarzania oraz sprawdzić czy na pewno klasyfikują poprawnie (alfa, b). [14]

6.2. Jakość danych

Najważniejszym fundamentem poprawnego działania aplikacji mającej przetwarzać dane jest jakość danych, które wykorzystuje się w tworzeniu modelu oraz ilość danych na których zostanie sprawdzone działanie modelu [15]. Jakościowa ocena danych podlega na ocenie kluczowych cech tych danych. Sposób ich wykorzystania będzie podyktowany ich najważniejszymi cechami. Dla przykładu, jeżeli dane będą posiadały zbyt dużą ilość wektorów, które opisują ich właściwości, głównym zadaniem algorytmów przygotowujących do klasyfikowania będzie ograniczenie wymiarowości danych tak, aby przetwarzanie było bardziej optymalne lub by zredukować cechy które są skorelowane z cechami sąsiadującymi. Dlatego tak kluczowe jest zbadanie pochodzenia danych, formatu zapisu lub ilości informacji, które przechowują. Kluczowe cechy danych jakie należy barć pod uwagę są opisane w tym rozdziale.

6.3. Cechy danych

Dane, które otrzymujemy i będziemy przetwarzać musimy przeanalizować pod względem błędów jakie mogą się w nich pojawić. Błędy będą powodować niestabilne działanie naszego modelu. Błędy na kolejnych etapach przetwarzania skomplikowanych danych promieniowania Ramana będą dodatkowo się nakładać na szumy samego sygnału. Ewentualna eliminacja niepoprawnego działania klasyfikatora lub niemożność analizy wyników będzie nieprawdopodobnie trudna. Liczba zmiennych, które mogą wpłynąć na wynik pomiaru sprawia, że zagadnienie jest bardzo trudne. Dlatego stopniowa analiza i zapisywanie głównych cech danych, da większą świadomość przy interpretacji wyników. Pierwszą ważną cechą danych jest występowanie w nich błędów systematycznych. Błędy te będą zawsze występowały w próbkach i wynikają najczęściej z defektów pomiarowych. W przypadku jednego środowiska pomiarowego nie trzeba brać ich pod uwagę, jednak jeżeli pomiary były wykonywane w różnym czasie za pomocą różnego sprzętu, należy się spodziewać, że w danych wystąpi błąd statystyczny. Kolejną cechą próbek pomiarowych jest szum losowy, który wystąpił w danych. Szum losowy to losowe odchylenie wartości od rzeczywistych wartości jakie powinny zostać pomierzone. Tego typu szumy zawsze występują w danych pomiarowych. Możemy go niwelować

jeżeli z wiedzy teoretycznej jesteśmy w stanie naprawiać błędy pomiarowe - wygładzić wykres lub wykasować część danych. Nie jest to łatwe biorąc pod uwagę złożoność danych. Często pomierzony szum może wydawać się istotną częścią pomierzonych danych, często niewielkie odchylenia, które wydają się szumem są istotną częścią danych. W wyniku tego problemu zdecydowano się nie usuwać szumów przy próbie klasyfikacyjnej. Uznano, że algorytm SVM sam doskonale eliminuje szумы, nadając im niskie wagi w przypadku wystąpienia zbyt różnych wartości. Kolejną cechą danych jaką należy brać pod uwagę jest liczba cech jaka występuje dla każdego wektora danych. Ilość cech należy starannie dobierać. Przetwarzanie zbyt dużej ilości cech dla danych powoduje niepożądane zjawisko przekleństwa wymiarów. Polega ono na niemożności generalizacji problemu, gdyż różnice pomiędzy każdym elementem wektora stają się zbyt rozproszone, odległości pomiędzy punktami na wykresie zbyt koherentne, cechy zaś wzajemnie ze sobą skorelowane. Z drugiej strony problem zbyt małej ilości wymiarów uniemożliwia właściwą interpretację danych, nie pozwala właściwie uchwycić badanego zjawiska. Rozwiązanie tego problemu staje się łatwiejsze wraz z praktyką inżynierską. Każdy autor algorytmu, który będzie przetwarzał dane, musi sam sobie odpowiedzieć na pytanie jaka ilość cech jest niezbędna do opisu zjawiska, a jednocześnie gdzie leży granica przekleństwa wymiarów. W niniejszej pracy rozwiązanie praktyczne było wymuszone przez cechy środowiska programistycznego. Ograniczenie liczby wymiarów przy przetwarzaniu do 256 wymusiło znalezienie sposobu na ograniczenie liczby cech czterokrotnie z 1024 lub 1084. Okazało się, że ta liczba wymiarów przy zastosowaniu wstępnego przetwarzania PCA oraz kompresji danych do najbardziej kluczowych 256 wektorów daje świetne wyniki, a problem przekleństwa wymiarów nie występuje.

Algorytmy klasyfikacyjne potrafią poradzić sobie z mocno zaszumionymi danymi i dać wynik swojego przetwarzania. Interpretowanie danych wyjściowych zależy jednak od osoby planującej eksperyment. Tak samo jest z jakością danych wejściowych. Autor eksperymentu, badacz czy programista są odpowiedzialni za jakość dostarczonych danych. Środowisko komputerowe, które często wprowadza dodatkowe błędy czy ograniczenia jest tylko narzędziem. Należy zdawać sobie sprawę z tego, że mierne efekty klasyfikowania przez algorytmy są kwestią błędów ludzkich, tam należy szukać ewentualnych przyczyn porażki.

6.4. Pochodzenie danych

Dane wykorzystane w przetwarzaniu to wrażliwe dane biologiczne, które są trudno dostępne. Próbkę krwi ludzkiej i zwierzęcej zostały uzyskane od instytucji, które mają kompetencje oraz odpowiednie zgody do ich pozyskiwania. Do pracy badawczej były przekazywane odpady, które wcześniej zostały pomierzone w celach diagnostycznych, dzięki czemu zyskiwały dodatkowy atut w postaci listy cech mogących zostać poddanych analizie. Pobieranie próbek nie było przedmiotem niniejszej pracy. Źródła pochodzenia to Regionalne Centrum Krwiodawstwa w Gdańsku oraz lecznica weterynaryjna.

7. DODATKOWE ROZWIĄZANIA PROGRAMISTYCZNE W PRACY MAGISTERSKIEJ

7.1. Baza danych

System klasyfikujący, który przetwarza big data, czyli olbrzymiej ilości dane dzieląc je i analizując, opierają jakość swojej pracy głównie na jakości tych danych. Fundamentem każdego systemu przetwarzania i klasyfikowania są dobrej jakości dane, które służą do tworzenia jak najbardziej uniwersalnego i pełnego modelu na podstawie którego przetwarzamy informacje. Analiza jakościowa danych jest tylko wstępnym etapem. Dane muszą być łatwo dostępne z bazy danych, która zapewni odpowiednią przepustowość i dostarczy je do algorytmu. W projekcie dyplomowym charakter wykonywania pomiarów a następnie przetwarzania ich w środowisku Matlab wymusił cechy wykorzystanego rozwiązania dla bazy danych. Po pierwsze zbudowany system z danymi musiał mieć cechy hybrydowe modelu hierarchicznego oraz modelu relacyjnego.

Baza danych pełni jeszcze jedną nie mniej ważną funkcję. System wbudowanego przetwarzania PCA, który został wykorzystany przy kompresji danych, inaczej pobiera dane niż system Toolbox LSSVM. Wiąże się to z tym, że algorytmy przetwarzania danych muszą otrzymać odpowiednio sformatowane, właściwe dla siebie dane. Dane są więc przechowywane w taki sposób, aby ich transpozycja była możliwie jak najmniej kosztowna dla pamięci RAM komputera oraz jak najszybsza.

7.2. Hierarchiczny element systemu hybrydowego składowania danych

Cechy systemu hierarchicznej baza danych dziedziczyła w sposobie w jakim pliki z widmami ramanowskim z eksperymentu były tworzone oraz przechowywane. Był to system wybitnie hierarchiczny, gdzie w folderach z nazwą eksperymentu, informacją o czasie wykonania oraz ilością pomiarów, były przechowywane poszczególne pliki macierzy z rozszerzeniem asc. Stanowisko komputerowe na stanowisku pomiarowym generowało także pliki o rozszerzeniu .sif oraz pliki sformatowane do plików tekstowych. Pliki jakie musiały znaleźć się w bazie danych pochodziły również ze starego systemu. Rozdzielczość laserów oraz kamery pomiarowej wynosiła 1084 lub 1086 pikseli. Utrudniło to dodatkowo pracę nad relacyjną częścią systemu bazodanowego. Niektóre pliki należało również usunąć, aby nie zakłócały pracy automatycznego wczytywania do bazy danych. Każdy plik był opisany numerami lub nazwą związku co umożliwiało jego identyfikację. Na razie nie dodano do bazy danych informacji umożliwiającej dokładną identyfikację próbek, jednak jak tylko się pojawi w formacie cyfrowym, będzie można to wykonać automatycznie. Hierarchiczna część bazy danych tworzy się standardowo tak jak w każdym systemie operacyjnym, gdzie zarządza się plikami. Niewątpliwą zaletą tej części hybrydowego rozwiązania jest możliwość łatwego udostępniania miejsca z bankiem danych, które mogą pochodzić od dowolnego stanowiska pomiarowego.

Wystarczy wymiar danych danego formatu dodać do automatycznego algorytmu wczytywania plików, który jest odpowiedzialny za część relacyjną bazy danych. Współczesne rozwiązania pozwalają udostępnić hierarchiczną bazę danych do dowolnego miejsca na świecie, dzięki czemu można wykonywać pomiary i dodawać je do analizy ze stanowiska pomiarowego mając tylko połączenie internetowe. Udostępnianie danych do badań jest równie proste. Hierarchiczny system jest czytelniejszy dla użytkownika przyzwyczajonego do korzystania z najpopularniejszych systemów operacyjnych. Pliki dodatkowo w sposób automatyczny przechowują dodatkowe informacje takie jak: data utworzenia oraz data modyfikacji. Pozwala to na jeszcze lepszą identyfikację i planowanie eksperymentów.

7.3. Relacyjny element systemu hybrydowego składowania danych

Środowisko przetwarzania danych Matlab nie działa w modelu hierarchicznym. Operuje na wektorach jedno bądź wielowymiarowych oraz na macierzach danych w postaci tablic. Model hierarchiczny nie sprawdza się w tym środowisku, ponieważ ukryte tablice w folderach trudno z nich wydobyć, zwłaszcza przy rozbudowanej strukturze drzewa. Idealnym rozwiązaniem do przetwarzania w aplikacjach z dziedziny big data są modele relacyjne baz danych. Na taką architekturę zdecydowano się przy rozwiązaniu przygotowanym do realizacji niniejszej pracy. Zaprojektowano algorytm, który w sposób automatyczny przeszukuje drzewa folderów modelu hierarchicznego i umieszcza dane w odpowiadających im macierzach. Macierze dzielą się ze względu na rozdzielczość danego systemu pomiarowego z którego pochodzą. W chwili obecnej są to macierze o wymiarze 1024 lub 1084 wierszy. Tablice danych zostały przygotowane w taki sposób, żeby możliwe było wykorzystanie danych od razu do trenowania modelu. Na końcu każdego zestawu w tablicy została dodana kolumna, w której przechowywana jest nazwa z jakiego pliku, czyli pomiaru, pochodzi dana.

Funkcja automatycznego przeczesywania folderów i tworzenia bazy relacyjnej zwraca również tablicę, która przechowuje wszystkie nazwy plików oraz wymiary macierzy, z których one pochodzą. Służy to późniejszym algorytmom automatycznego przetwarzania tych danych. Do algorytmu dodano okno dialogowe, gdzie wyświetlają się postępy w importowaniu danych. Proces przy obecnej ilości plików pomiarowych trwa około 15 minut. W przyszłości należy go zoptymalizować. Nie jest to jednak niezbędne do płynnego działania programu, ponieważ operacja wczytania bazy danych jest jednorazowa. Widać w przypadku przetwarzania tak dużych tablic problem wydajności środowiska Matlab. Nie można wykonać podglądu tablicy, która posiada więcej niż 524288 elementów. Ograniczenia Matlab zostały pokonane poprzez podział na tablice danych po 100 plików. Dzięki temu uzyskano większą płynność działania. Jest to istotne ze względu na możliwości obliczeniowe komputerów osobistych. Na takim właśnie były przygotowywane klasyfikatory do przetwarzania, co naturalnie stworzyło dodatkowe ograniczenia obliczeniowe. Jeżeli będzie konieczność bazy danych można rozszerzyć o dodatkowe tablice.

8. WYNIKI

Uzyskane wyniki były inne niż oczekiwano. Po próbach na czystych danych, które się nie udały, prace utknęły w miejscu ze względu na problemy z wydajnością środowiska oraz ogromne problemy z sklasyfikowaniem próbek, nieczytelne wykresy. Przełom nastąpił wraz z odkryciem ograniczeń środowiska programistycznego. Pojawił się pomysł kompresji danych. Wykorzystano do tego algorytm PCA. Praca zakończyła się sukcesem. Klasyfikator zadziałał na danych skompresowanych.

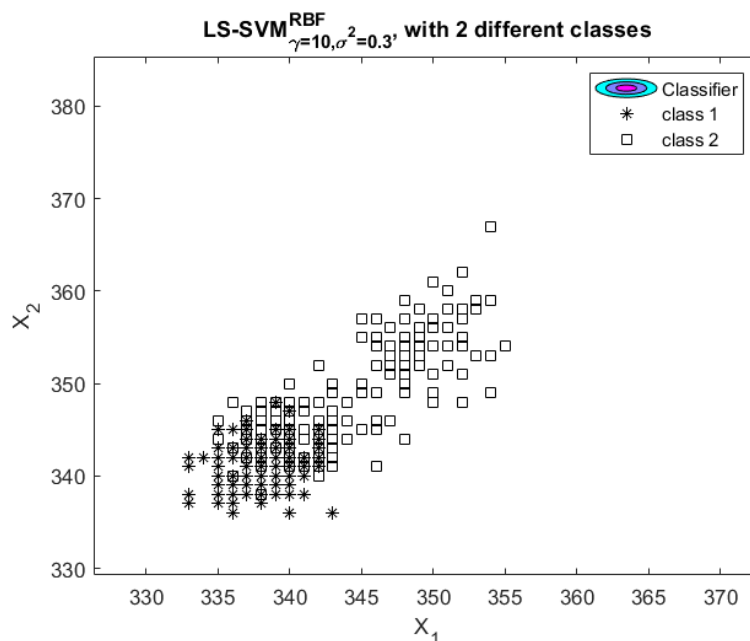
W pierwszej kolejności, przed utworzeniem bazy danych, dokonano sprawdzenia możliwości algorytmu i jego poprawnego działania. Wykonano to na zbiorze danych iris.data.set. Jest to zbiór danych zawierający przykładowe wymiary płatków kwiatów z rodziny irysów. [16]. Pozwalał on dostroić algorytm ponieważ wynik klasyfikacji tego zbioru jest znany. Dobrano odpowiednie wartości gamma i sigma, które były najlepsze dla zbioru strojącego. Wartości gamma i sigma określają szerokość marginesu jaki stosujemy pomiędzy rozdzielonymi płatkami kwiatów. Następnie dla tych ustawień wykonano analizę na bazie danych. Pozwoliło to poczynić obserwacje na temat skuteczności tego algorytmu.

Zebrane próbki widma ramanowskiego zawierają liczne defekty, które należy wyeliminować przy kolejnych próbach klasyfikacji za pomocą TOOLBOX LSSVM. Podstawowym defektem fizycznym próbek była degradacja próbek organicznych krwi. W części teoretycznej opisano zależności pomiędzy promieniowaniem ramanowskim, a wiązką wzbudzającą. Konkluduje to do wniosku: im większe natężenie wiązki padającej, tym silniejsze efekty rozpraszania będą obserwowane. Niestety próbki organiczne można zniszczyć poprzez wystawienie ich na działanie dużej mocy przez zbyt długi czas. Czas jest też ograniczony zmianą stanu skupienia przez krew (ze względu na krzepnięcie). W czasie wykonywania pomiarów trzeba było zastosować kompromis pomiędzy długością czasu naświetlania, a mocą wiązki laserowej. Nie było tutaj ustalonej procedury, czasy naświetlania były wyznaczane eksperymentalnie lub wynikały z doświadczenia osoby wykonującej pomiar. Niestety w wyniku zniszczenia próbek ponowne pomiary dla niektórych z nich nie nadawały się do dalszego przetwarzania. W środowisku programistycznym problem został ominięty, udało się łączyć macierze w poszczególne bloki. Analiza porównawcza była realizowana na tych blokach dla poszczególnych próbek. W niniejszej pracy zrezygnowano z wyznaczania średniej, ponieważ próbki miały za mało cech charakterystycznych.

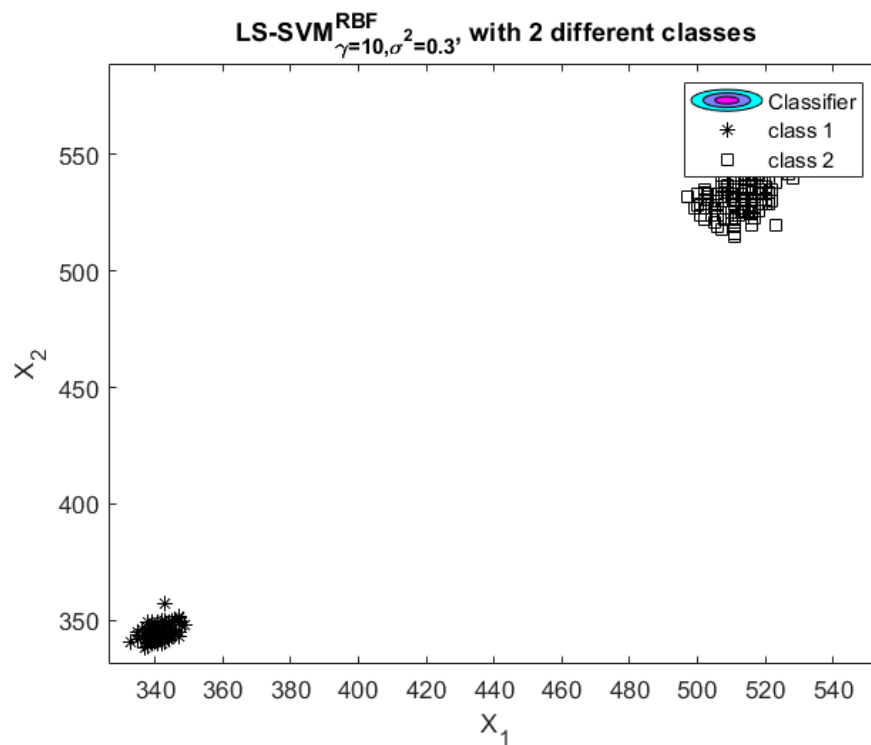
Wyniki eksperymentów niestety nie są satysfakcjonujące. Zebrane dane nie pozwoliły się satysfakcjonująco porównywać za pomocą metod nieliniowych. Pojawiały się błędy oraz problemy z wydajnością. Środowisko programistyczne często się zawieszało. Błędy polegały na całkowitym rozproszeniu danych w modelu. SVM nie mógł wyznaczyć miękkiego marginesu. Często zdążało się wręcz przeciwnie, że margines obejmował wszystkie próbki zarówno jednej jak i drugiej klasy

badanych danych. Błędy wskazywały ewidentnie na zbyt dużą ilość zastosowanych wymiarów. Wyglądały jakby rozdzielczość klasyfikatora była za mała, a było wręcz przeciwnie.

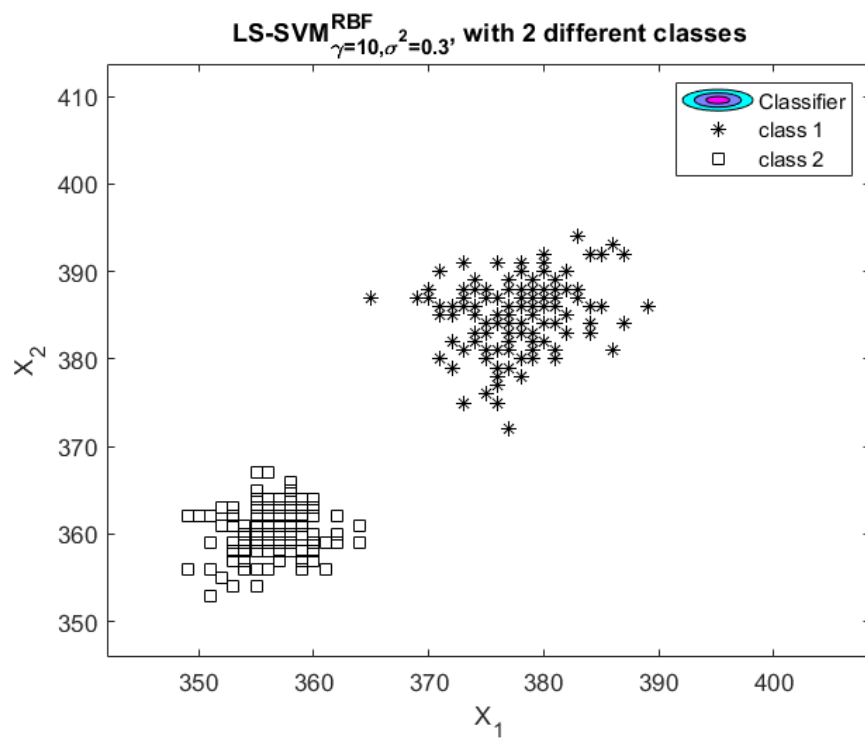
Powrócono do analizy dokumentacji. Programistyczne ograniczenie dla środowiska Matlab oraz Toolbox-a LSSVM to maksymalna ilość 256 wymiarów dla próbki. Należało zmniejszyć rozdzielczość plików pochodzących z kamery tak, aby były możliwe w przetwarzaniu przez maszynę wektorów nośnych. Zdecydowano się zastosować kompresję, jednak żeby zachować jak najwięcej informacji dane musiały być przeanalizowane przez inny algorytm. Spełniającym kryteria wyznaczania najbardziej znaczących wartości był algorytm PCA (analiza głównych składowych). Algorytm działał następująco. Utworzoną bazę danych kompresowano do wektorów mających maksymalnie 256 cech charakterystycznych, następnie tak skompresowaną bazę danych przetwarzano za pomocą funkcji, która realizowała trenowanie modelu dla danej klasy. Funkcja na końcu sprawdzała skuteczność wykrywania modelu. Udało się osiągnąć satysfakcjonujący wynik powyżej 90% skuteczności dla rozpoznawania krwi pomiędzy dawcami, krwi dawcy z lekiem od krwi dawcy bez leku. Algorytm słabiej działał, bo tylko ze skutecznością 60% dla krwi pochodzącej od psów. Może to być spowodowane słabą jakością pomiarów. Poniższe wykresy przedstawiają przykładowe rozdzielenie. Wyniki na wykresach od 16 do 22 prezentują poklasyfikowane widma Ramana. Dla opracowanego algorytmu nie ma znaczenia pochodzenia próbki. Potrafi klasyfikować również pliki z różnych systemów. Baza danych składała się z około 20 000 wektorów pomierzonych widm. Poniżej została zaprezentowana najciekawsza seria porównywanych próbek, która wskazuje kierunki dalszych możliwości aplikacji systemu.



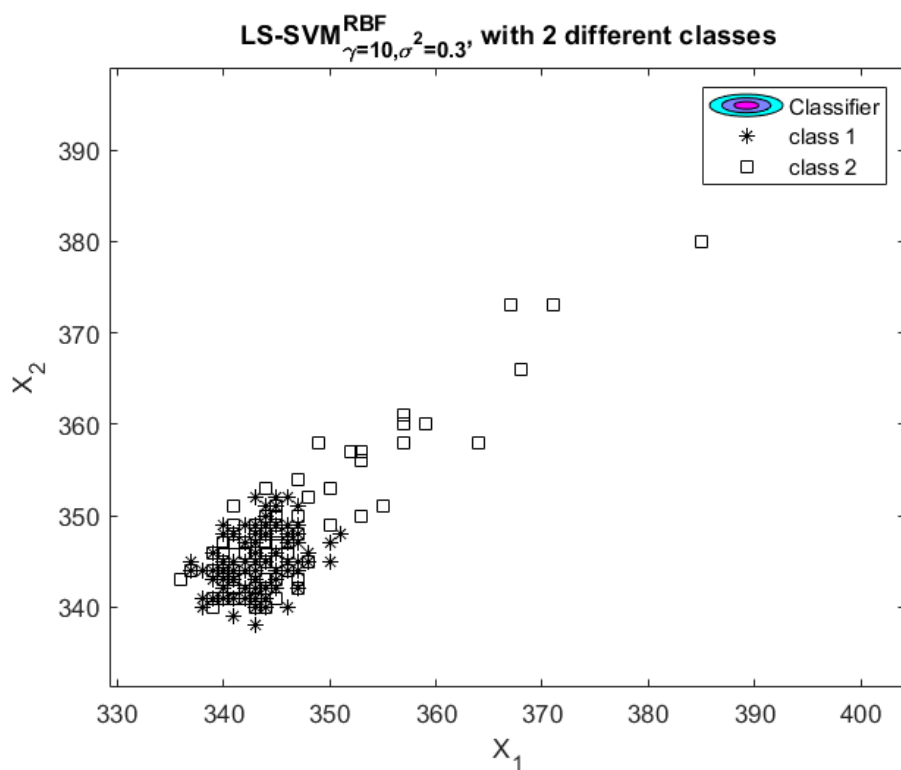
Wykres 16 Pogrupowanie krwi różnych dawców na starym systemie pomiarowym (1- VTT Ramstas).



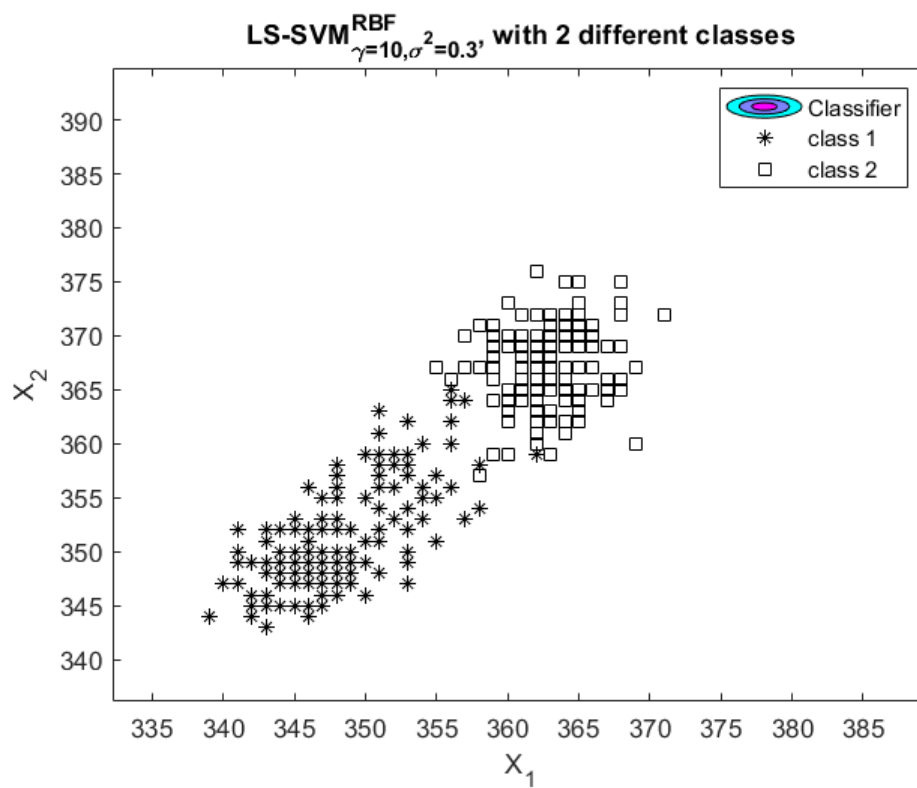
Wykres 17 Rozdzielenie danych dla alkoholu oraz pustej płytki pomiarowej (1- VTT Ramstas).



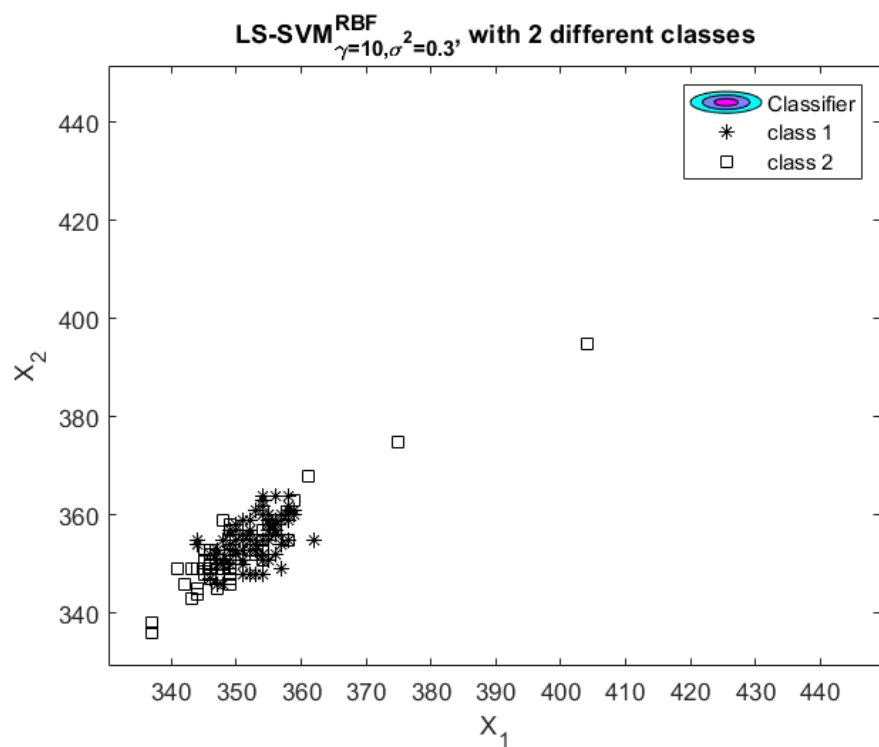
Wykres 18. Sklasyfikowane dane dla różnych alkoholi (1- VTT Ramstas).



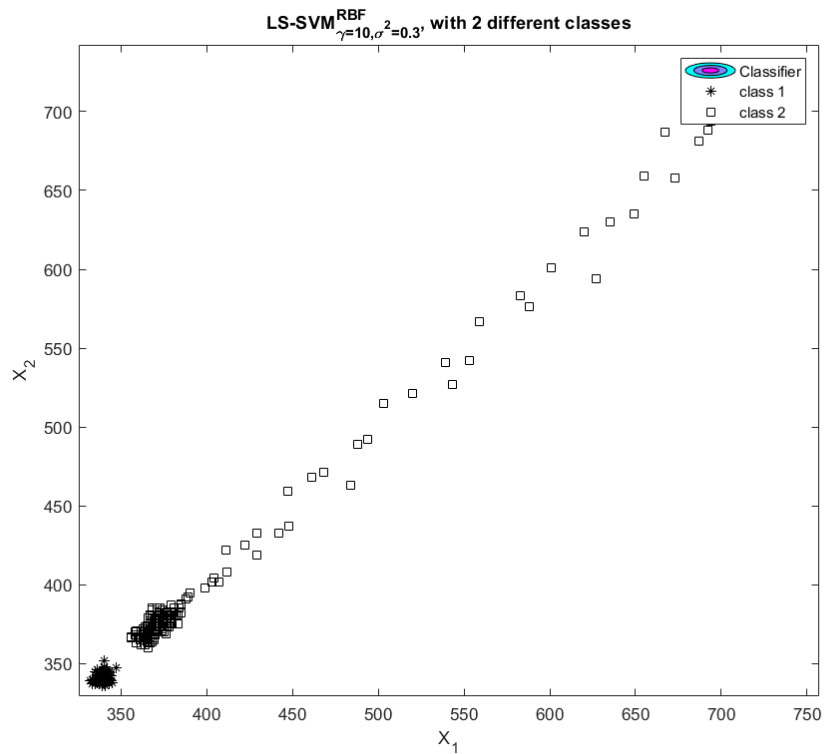
Wykres 19. Sklasyfikowane dane krwi 2 różnych psów (2-system z kamerą Andor)



Wykres 20. Sklasyfikowane dane dla krwi z lekiem oraz bez leku(2-system z kamerą Andor)



Wykres 21. Sklasyfikowane dane dla krwi 2 różnych psów (2-system z kamerą Andor)



Wykres 22. Sklasyfikowane dane dla krwi psiej i ludzkiej(2-system z kamerą Andor)

9. DYSKUSJA I WNIOSKI

Głównym problemem jaki został postawiany przed wykonaniem pracy było sprawdzenie, czy skomplikowane widma ramanowskie, które różnią się między sobą w bardzo małym stopniu można rozróżnić i odpowiednio sklasyfikować. Materiał posiada bardzo skomplikowaną strukturę, której widmo przeniesione do łatwej do oceny dla człowieka przestrzeni, jaką jest wykres, jest nieodróżnialne od innego widma podobnego materiału. Badana materia jest pełna licznych błędów statystycznych, bądź szumu losowego, co często uniemożliwia wstępne porównanie. Problematyka z jaką się zmierzono została podzielona na etapy, które bezpośrednio przełożyły się na kształt programu badawczego.

Program badawczy składał się następujących etapów:

1. Zaprojektowanie bazy danych, która będzie składała się z systemu hybrydowego (relacyjno- hierarchicznego)
2. Dostrojenia przeniesionego do środowiska Matlab algorytmu SVM (maszyny wektorów nośnych)
3. Analizy przetworzenia surowych danych w algorytmie SVM
4. Zastosowania kompresji danych przez algorytm PCA
5. Klasyfikacji próbek badanych, wstępnie skompresowanych PCA, za pomocą algorytm SVM

Dane skompresowane za pomocą algorytmów SVM zostały poddane procesowi klasyfikacji. Udało się stworzyć modele klasyfikujące o bardzo wysokim stopniu wykrywania powyżej 90%. Jedynie próbki krwi psiej wykazywały zbyt małą zmienność osobniczą lub pomiary miały gorszą jakość. Udało się też sprawdzić, że widma które były zarejestrowane w postaci długiego naświetlania były skuteczniej rozróżnialne przez algorytm niż widma zarejestrowane dużą ilością, ale bardzo krótkich naświetlań. W pracy badawczej wykorzystane zostało środowisko programistyczne Matlab 2018b. Toolbox LSSVM w wersji 1.8.

Praca badawcza pokazała, że istnieje możliwość zbudowania systemu bezinwazyjnego pomiaru składników krwi. Należy w tym celu stworzyć model, który będzie opierał się na znacznie większej ilości danych szczegółowych pochodzących od dawców. Przy odpowiedniej rozdzielczości dostarczonych próbek pomiarowych będzie można przeprowadzać dokładną analizę makro i mikro elementową krwi. Warto też rozważyć zrezygnowanie ze środowiska Matlab, ponieważ wymusza ono na programiście ograniczenia, które w przypadku systemu komputera osobistego są zasadne, ale jeżeli będziemy dysponować większą mocą obliczeniową, możemy nie wykorzystać w pełni potencjału zebranych danych.

Program badawczy sugeruje też, że można przy odpowiednio wykonanych pomiarach zbudować system rozpoznawania próbek krwi, bezpośrednio przez tkanki miękkie pacjenta, bez inwazji w jego układ krwionośny. [17]. Należy dysponować jeszcze większą rozdzielczością oraz większą wydajnością komputera, a także odizolować sygnał pochodzący od tkanek. Stanowisko powinno zostać pozbawione wszelkich ograniczeń, sztucznie wymuszanych na programiście.

W czasie badań przeprowadzono eksperymenty, które wykazały, że można odróżnić krew psią od ludzkiej. Na takie rozwiązania czekają analitycy kryminalni, którzy na potencjalnym miejscu zbrodni, kiedy zabezpieczają ślady, potrzebują szybkiej analizy liczby ofiar. [18]

Czasopisma naukowe zawierają liczne przykłady możliwego zastosowania [19] [20] [21] [22].

Eksperyment pokazał, że możemy rozróżnić widma krwi z różnym stężeniem leku. Umożliwi to szybkie wykrywanie jaka dawkę powinna zostać jeszcze podana pacjentowi przez lekarza lub pielęgniarkę celem osiągnięcia żądanej koncentracji, co jest ważne np. przy podawaniu leków anestezjologicznych. Będzie można też badać krew w celu wykrycia niebezpiecznych substancji lub substancji psychoaktywnych. Badania nad uzyskaniem klasyfikacji z widma krwi w fantomie wciąż trwają.

10. WYKAZ LITERATURY

- [1] J. Sadlej, Spektroskopia Molekularna, Warszawa: Wydawnictwo Naukowo-Techniczne, 2002.
- [2] W. Demtröder, Spektroskopia Laserowa, Warszawa: Wydawnictwo Naukowe PWN, 1993.
- [3] M. Drozdowski, Spektroskopia ciała stałego, Poznań: Wydawnictwo Politechniki Poznańskiej, 1996.
- [4] Gardiner and Graves, Practical Raman Spectroscopy, Springer-verlag, 1991.
- [5] Z. Kęcki, Podstawy spektroskopii molekularnej, Warszawa: Wydawnictwo PWN, 1998.
- [6] Kuptsov i A. H. Zhizhin, Handbook of Fourier Transform Raman and Infrared Spectra of Polymers, Elsevier, 1998.
- [7] H. Barańska, A. Łabudzińska i J. Terepiński, Laserowa Spektrometria Ramanowska, Warszawa: Państwowe Wydawnictwo Naukowe, 1981.
- [8] M. KERANEN, M. GNYBA i P. RAERINNE, „Synthesis and Characterization of Optical Sol–Gel Adhesive for Military,” *Journal of Sol-Gel Science and Technology*, tom 0.1023, nr JSST.0000048019.00322., 2004.
- [9] M. Gnyba, M. Wróbel, R. Urniaż, T. Myllylä i M. Jędrzejwska-Szczerska, „Detection of propofol concentrations in blood by Raman spectroscopy,” Munich, 2015.
- [10] K. Cieśliski, Hydrodynamiczne uwarynkowania krążenia mózgowego, Warszawa: Akademicka Oficyna Wydawnicza, 2001.
- [11] M. Szeliga, Data Science i uczenie maszynowe, Warszawa: Wydawnictwo Naukowe PWN SA, 2017.
- [12] K. D. Brabanter, „lssvmlab,” 30 9 2019. [Online]. Available: <https://www.esat.kuleuven.be/sista/lssvmlab/>. [Data uzyskania dostępu: 30 9 2019].
- [13] R. Fisher, „The use of multiple measurements in taxonomic problems,” *Annual Eugenics*, pp. 179-188, 1936.
- [14] R. Fisher, „<https://archive.ics.uci.edu/>,” 30 wrzesień 2019. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/iris>. [Data uzyskania dostępu: 30 wrzesień 2019].
- [15] D. McIlwraith, H. Marmanis i D. Babenko, Inteligentna sieć. Algorytmy przyszłości. Wydanie II, Gliwice: Helion, 2017.
- [16] R. L. McCreery, Raman Spectroscopy for Chemical Analysis, John Wiley & Sons, 2005.

- [17] YONGZENG L, WEI H i RONG C, „Rapid detection of nasopharyngeal cancer using Raman spectroscopy and multivariate statistical analysis,” *Molecular & Clinical Oncology*, pp. 375-380, 2015.
- [18] McLaughlin, Doty i Lednev, „Discrimination of human and animal blood traces via Raman spectroscopy,” *Forensic Science International*, pp. 23891-95, 2014.
- [19] E. Gebel, „Species in a snap: Raman analysis of blood,” *Anal. Chem.*, p. 786, 2009.
- [20] K. Virkler i I.K. Lednev, „Blood species identification for forensic purposes using Raman spectroscopy combined with advanced statistical analysis,” *Anal. Chem.*, p. 7773–7777, 2009.
- [21] K. Virkler i I.K. Lednev, „Raman spectroscopic signature of blood and its potential application to forensic body fluid identification,” *Anal. Bioanal. Chem.*, p. 525–534, 2010.
- [22] McLaughlin G, Sikirzhyski V i Lednev I., „Circumventing substrate interference in the Raman spectroscopic identification of blood stains,” *Forensic Science International*, pp. 157-166, 2013.
- [23] W.R. Premasiri, J.C. Lee i L.D. Ziegler, „Surface-enhanced Raman scattering of whole human blood, blood plasma, and red blood cells: cellular processes and bioanalytical sensing,” *J. Phys. Chem. B*, p. 9376–9386, 2012.
- [24] M. Szeliga, *Data Science i uczenie maszynowe*, Warszawa: Wydawnictwo Naukowe PWN SA, 2017.
- [25] J. Suykens, „Kernel methods for complex networks and big data,” w *invited lecture at Statlearn*, Grenoble, 2015.
- [26] Kourkoumelis N, Balatsoukas I, Moulia V, Elka A, Gaitanis G i Bassukas I, „Advances in the in Vivo Raman Spectroscopy of Malignant Skin Tumors Using Portable Instrumentation,” *International Journal Of Molecular Sciences*, pp. 14554-14570, 2015.
- [27] Czamara K, Majzner K, Pacia M, Kochan K, Baranska M i Kaczor A, „Raman spectroscopy of lipids,” *Journal of Raman Spectroscopy*, pp. 14-17, 2015.
- [28] W. Batur, *najważniejsze teorie biologii*, Warszawa: Wydawnictwo naukowe PWN, 2011.
- [29] "The MathWorks," 06 12 2018. [Online]. Available: <http://www.mathworks.com/help/stats/support-vector-machines-svm.html?searchHighlight=SVM>.
- [30] „The MathWorks,” 06 12 2015. [Online]. Available: <http://www.mathworks.com/help/stats/pca.html?searchHighlight=PCA>.

[31] Praca zbiorowa pod redakcją Kamili Małek, Spektroskopia Oscylacyjna, Warszawa:
Wydawnictwo Naukowe PWN SA, 2016.

11. ZAŁĄCZNIKI - KOD ŹRÓDŁOWY

11.1. *Kod algorytmu tworzącego element relacyjny bazy danych*

```
function
[name,dane1024_cz1,dane1086_cz1,dane1084_cz1,danedInne_cz1,dane1024_cz2,dane1086_cz2
,dane1084_cz2,danedInne_cz2,dane1024_cz3,dane1086_cz3,dane1084_cz3,danedInne_cz3,dane
1024_cz4,dane1086_cz4,dane1084_cz4,danedInne_cz4,dane1024_cz5,dane1086_cz5,dane1084_
cz5,danedInne_cz5,dane1024_cz6,dane1086_cz6,dane1084_cz6,danedInne_cz6,dane1024_cz7,d
ane1086_cz7,dane1084_cz7,danedInne_cz7,dane1024_cz8,dane1086_cz8,dane1084_cz8,danedl
nne_cz8] = skryptdowyciaganiadanych()

%Tworzenie okienka z informacją o postępie załadowania danych do bazy

%danych

f = waitbar(0,'Start','Name','Ładowanie danych do macierzy',...
'CreateCancelBtn','setappdata(gcf,'canceling',1)');

%Tworzeni przycisku przerywania operacji

setappdata(f,'canceling',0);

%Tworzenie listy plików i lokacji, w których występują

Base = 'C:\Users\AdamJulia\Documents\awladzinskiraman\DaneRamanaDoPrzetwarzania';

List = dir(fullfile(Base, '**', '*.*'));

name = {List.name}.';

folder = {List.folder}.';

Files = [folder, name];

Files = Files .';

%Tworzenie adresów bezpośrednio potrzebnych do importowania

Adresy = strcat(Files(1,:),{'\'},Files(2,:));
```

```
Adresy = Adresy .';
```

```
%Czyszczenie adresów z obiektów nieistniejących
```

```
for i = length(Adresy): -1 : 1
```

```
    A = exist(Adresy{i}, 'file');
```

```
    if (A == 7 || 0)
```

```
        Adresy(i) = [];
```

```
        name(i) = [];
```

```
    end
```

```
end
```

```
%Inicjalizowanie macierzy z danymi, aby poprawić wydajność operacji
```

```
danedotreningu1024_cz1 = zeros(1025,1);
```

```
danedotreningu1084_cz1 = zeros(1085,1);
```

```
danedotreningu1086_cz1 = zeros(1087,1);
```

```
danedotreninguInne_cz1 = zeros(1025,1);
```

```
danedotreningu1024_cz2 = zeros(1025,1);
```

```
danedotreningu1084_cz2 = zeros(1085,1);
```

```
danedotreningu1086_cz2 = zeros(1087,1);
```

```
danedotreninguInne_cz2 = zeros(1025,1);
```

```
danedotreningu1024_cz3 = zeros(1025,1);
```

```
danedotreningu1084_cz3 = zeros(1085,1);
```

```
danedotreningu1086_cz3 = zeros(1087,1);
```

```
danedotreninguInne_cz3 = zeros(1025,1);
```

```
danedotreningu1024_cz4 = zeros(1025,1);  
danedotreningu1084_cz4 = zeros(1085,1);  
danedotreningu1086_cz4 = zeros(1087,1);  
danedotreningulnne_cz4 = zeros(1025,1);
```

```
danedotreningu1024_cz5 = zeros(1025,1);  
danedotreningu1084_cz5 = zeros(1085,1);  
danedotreningu1086_cz5 = zeros(1087,1);  
danedotreningulnne_cz5 = zeros(1025,1);
```

```
danedotreningu1024_cz6 = zeros(1025,1);  
danedotreningu1084_cz6 = zeros(1085,1);  
danedotreningu1086_cz6 = zeros(1087,1);  
danedotreningulnne_cz6 = zeros(1025,1);
```

```
danedotreningu1024_cz7 = zeros(1025,1);  
danedotreningu1084_cz7 = zeros(1085,1);  
danedotreningu1086_cz7 = zeros(1087,1);  
danedotreningulnne_cz7 = zeros(1025,1);
```

```
danedotreningu1024_cz8 = zeros(1025,1);  
danedotreningu1084_cz8 = zeros(1085,1);  
danedotreningu1086_cz8 = zeros(1087,1);  
danedotreningulnne_cz8 = zeros(1025,1);
```

```

%Maksymalna liczba kroków do wykonania potrzebna w pasku postępu

steps = size(name,1);

%Główna pętla wczytująca dane

for i = 1 : size(Adresy,1)

    danezaimportowane = importdata(Adresy{i});

    danezaimportowane(:,1) = [];

    wielkoscdanych = size(danezaimportowane);

    %funkcja przerywająca pętlę w razie wciśnięcia przycisku cancel

    if getappdata(f,'canceling')

        break

    end

    %Dodawanie numeru id nazwy z tabeli z nazwami.

    nazwadoklasyfikacji = ones(wielkoscdanych(:,2),1);

    nazwadoklasyfikacji(:) = i;

    nazwadoklasyfikacji = nazwadoklasyfikacji .';

    name(i,2:3) = num2cell(wielkoscdanych);

    %Tworzenie danych w jednej operacji

    if( wielkoscdanych(:,1) == 1024)

        DaneZAdresy1024 = [danezaimportowane; nazwadoklasyfikacji ];

    elseif (wielkoscdanych(:,1) == 1086)

        DaneZAdresy1086 = [danezaimportowane; nazwadoklasyfikacji ];

```

```

elseif (wielkoscdanych(:,1) == 1084)

    DaneZAdresy1084 = [danezaimportowane; nazwadoklasyfikacji ];

else

    DaneZAdresyInne = [danezaimportowane; nazwadoklasyfikacji ];

end

%Wczytywanie macierzy, podział został wymuszony ograniczeniami
%środowiska

if (100<i)&&(i<=200)

    if( wielkoscdanych(:,1) == 1024)

        danedotreningu1024_cz2 = [danedotreningu1024_cz2, DaneZAdresy1024];

    elseif (wielkoscdanych(:,1) == 1086)

        danedotreningu1086_cz2 = [danedotreningu1086_cz2, DaneZAdresy1086];

    elseif (wielkoscdanych(:,1) == 1084)

        danedotreningu1084_cz2 = [danedotreningu1084_cz2, DaneZAdresy1084];

    else

        danedotreninguInne_cz2 = [danedotreninguInne_cz2, DaneZAdresyInne];

    end

elseif(200<i)&&(i<=300)

    if( wielkoscdanych(:,1) == 1024)

        danedotreningu1024_cz3 = [danedotreningu1024_cz3, DaneZAdresy1024];

    elseif (wielkoscdanych(:,1) == 1086)

        danedotreningu1086_cz3 = [danedotreningu1086_cz3, DaneZAdresy1086];

    elseif (wielkoscdanych(:,1) == 1084)

        danedotreningu1084_cz3 = [danedotreningu1084_cz3, DaneZAdresy1084];

    else

```

```

    danedotreningulInne_cz3 = [danedotreningulInne_cz3, DaneZAdresyInne];

    end

elseif(300<i)&&(i<=400)

    if( wielkoscdanych(:,1) == 1024)

        danedotreningu1024_cz4 = [danedotreningu1024_cz4, DaneZAdresy1024];

        elseif (wielkoscdanych(:,1) == 1086)

            danedotreningu1086_cz4 = [danedotreningu1086_cz4, DaneZAdresy1086];

            elseif (wielkoscdanych(:,1) == 1084)

                danedotreningu1084_cz4 = [danedotreningu1084_cz4, DaneZAdresy1084];

                else

                    danedotreningulInne_cz4 = [danedotreningulInne_cz4, DaneZAdresyInne];

                    end

elseif(400<i)&&(i<=500)

    if( wielkoscdanych(:,1) == 1024)

        danedotreningu1024_cz5 = [danedotreningu1024_cz5, DaneZAdresy1024];

        elseif (wielkoscdanych(:,1) == 1086)

            danedotreningu1086_cz5 = [danedotreningu1086_cz5, DaneZAdresy1086];

            elseif (wielkoscdanych(:,1) == 1084)

                danedotreningu1084_cz5 = [danedotreningu1084_cz5, DaneZAdresy1084];

                else

                    danedotreningulInne_cz5 = [danedotreningulInne_cz5, DaneZAdresyInne];

                    end

elseif(500<i)&&(i<=600)

    if( wielkoscdanych(:,1) == 1024)

        danedotreningu1024_cz6 = [danedotreningu1024_cz6, DaneZAdresy1024];

        elseif (wielkoscdanych(:,1) == 1086)

```

```

danedotreningu1086_cz6 = [danedotreningu1086_cz6, DaneZAdresy1086];

elseif (wielkoscdanych(:,1) == 1084)

danedotreningu1084_cz6 = [danedotreningu1084_cz6, DaneZAdresy1084];

else

danedotreninguInne_cz6 = [danedotreninguInne_cz6, DaneZAdresyInne];

end

elseif(400<i)&&(i<=500)

if( wielkoscdanych(:,1) == 1024)

danedotreningu1024_cz7 = [danedotreningu1024_cz7, DaneZAdresy1024];

elseif (wielkoscdanych(:,1) == 1086)

danedotreningu1086_cz7 = [danedotreningu1086_cz7, DaneZAdresy1086];

elseif (wielkoscdanych(:,1) == 1084)

danedotreningu1084_cz7 = [danedotreningu1084_cz7, DaneZAdresy1084];

else

danedotreninguInne_cz7 = [danedotreninguInne_cz7, DaneZAdresyInne];

end

elseif(600<i)

if( wielkoscdanych(:,1) == 1024)

danedotreningu1024_cz8 = [danedotreningu1024_cz8 , DaneZAdresy1024];

elseif (wielkoscdanych(:,1) == 1086)

danedotreningu1086_cz8 = [danedotreningu1086_cz8 , DaneZAdresy1086];

elseif (wielkoscdanych(:,1) == 1084)

danedotreningu1084_cz8 = [danedotreningu1084_cz8 , DaneZAdresy1084];

else

danedotreninguInne_cz8 = [danedotreninguInne_cz8 , DaneZAdresyInne];

end

```

```

else

    if( wielkoscdanych(:,1) == 1024)

        danedotreningu1024_cz1 = [danedotreningu1024_cz1 , DaneZAdresy1024];

        elseif (wielkoscdanych(:,1) == 1086)

            danedotreningu1086_cz1 = [danedotreningu1086_cz1 , DaneZAdresy1086];

            elseif (wielkoscdanych(:,1) == 1084)

                danedotreningu1084_cz1 = [danedotreningu1084_cz1 , DaneZAdresy1084];

            else

                danedotreninguInne_cz1 = [danedotreninguInne_cz1 , DaneZAdresyInne];

            end

        end

    end

    if not(i/steps == 1)

        waitbar(i/steps,f, strcat(num2str(i), ' z ', num2str(steps)))

    else

        waitbar(i/steps,f,sprintf('Koniec'))

    end

end

end

```

%Czyszczenie kolumny inicjalizującej

```

danedotreningu1024_cz1(:,1) = [];
danedotreningu1086_cz1(:,1) = [];
danedotreningu1084_cz1(:,1) = [];
danedotreninguInne_cz1(:,1) = [];
danedotreningu1024_cz2(:,1) = [];
danedotreningu1086_cz2(:,1) = [];
danedotreningu1084_cz2(:,1) = [];

```



```
danedotreninguInne_cz2(:,1) = [];  
danedotreningu1024_cz3(:,1) = [];  
danedotreningu1086_cz3(:,1) = [];  
danedotreningu1084_cz3(:,1) = [];  
danedotreninguInne_cz3(:,1) = [];  
danedotreningu1024_cz4(:,1) = [];  
danedotreningu1086_cz4(:,1) = [];  
danedotreningu1084_cz4(:,1) = [];  
danedotreninguInne_cz4(:,1) = [];  
danedotreningu1024_cz5(:,1) = [];  
danedotreningu1086_cz5(:,1) = [];  
danedotreningu1084_cz5(:,1) = [];  
danedotreninguInne_cz5(:,1) = [];  
danedotreningu1024_cz6(:,1) = [];  
danedotreningu1086_cz6(:,1) = [];  
danedotreningu1084_cz6(:,1) = [];  
danedotreninguInne_cz6(:,1) = [];  
danedotreningu1024_cz7(:,1) = [];  
danedotreningu1086_cz7(:,1) = [];  
danedotreningu1084_cz7(:,1) = [];  
danedotreninguInne_cz7(:,1) = [];  
danedotreningu1024_cz8(:,1) = [];  
danedotreningu1086_cz8(:,1) = [];  
danedotreningu1084_cz8(:,1) = [];  
danedotreninguInne_cz8(:,1) = [];
```

```
%Przypisanie do danych wyjściowych  
  
dane1024_cz1 = danedotreningu1024_cz1;  
dane1086_cz1 = danedotreningu1086_cz1;  
dane1084_cz1 = danedotreningu1084_cz1;  
daneInne_cz1 = danedotreninguInne_cz1;  
  
dane1024_cz2 = danedotreningu1024_cz2;  
dane1086_cz2 = danedotreningu1086_cz2;  
dane1084_cz2 = danedotreningu1084_cz2;  
daneInne_cz2 = danedotreninguInne_cz2;  
  
dane1024_cz3 = danedotreningu1024_cz3;  
dane1086_cz3 = danedotreningu1086_cz3;  
dane1084_cz3 = danedotreningu1084_cz3;  
daneInne_cz3 = danedotreninguInne_cz3;  
  
dane1024_cz4 = danedotreningu1024_cz4;  
dane1086_cz4 = danedotreningu1086_cz4;  
dane1084_cz4 = danedotreningu1084_cz4;  
daneInne_cz4 = danedotreninguInne_cz4;  
  
dane1024_cz5 = danedotreningu1024_cz5;  
dane1086_cz5 = danedotreningu1086_cz5;  
dane1084_cz5 = danedotreningu1084_cz5;  
daneInne_cz5 = danedotreninguInne_cz5;  
  
dane1024_cz6 = danedotreningu1024_cz6;  
dane1086_cz6 = danedotreningu1086_cz6;  
dane1084_cz6 = danedotreningu1084_cz6;  
daneInne_cz6 = danedotreninguInne_cz6;  
  
dane1024_cz7 = danedotreningu1024_cz7;
```

```

dane1086_cz7 = danedotreningu1086_cz7;

dane1084_cz7 = danedotreningu1084_cz7;

dane1Inne_cz7 = danedotreningu1Inne_cz7;

dane1024_cz8 = danedotreningu1024_cz8;

dane1086_cz8 = danedotreningu1086_cz8;

dane1084_cz8 = danedotreningu1084_cz8;

dane1Inne_cz8 = danedotreningu1Inne_cz8;

delete(f)

end

```

11.2. Kod algorytmu przetwarzającego wstępne skompresowane dane, oraz tworzenie bazy danych do klasyfikacji

```

[name,dane1024_cz1,dane1086_cz1,dane1084_cz1,dane1Inne_cz1,dane1024_cz2,dane1086_cz2,dane1084_cz2,dane1Inne_cz2,dane1024_cz3,dane1086_cz3,dane1084_cz3,dane1Inne_cz3,dane1024_cz4,dane1086_cz4,dane1084_cz4,dane1Inne_cz4,dane1024_cz5,dane1086_cz5,dane1084_cz5,dane1Inne_cz5,dane1024_cz6,dane1086_cz6,dane1084_cz6,dane1Inne_cz6,dane1024_cz7,dane1086_cz7,dane1084_cz7,dane1Inne_cz7,dane1024_cz8,dane1086_cz8,dane1084_cz8,dane1Inne_cz8] = skryptdowyciaganiadanych();

```

```

[macierzuproszczona_dane1024_cz1,macierzuproszczona_dane1084_cz8,macierzuproszczona_dane1084_cz6,macierzuproszczona_dane1084_cz5,macierzuproszczona_dane1084_cz4,macierzuproszczona_dane1084_cz3,macierzuproszczona_dane1084_cz2,macierzuproszczona_dane1024_cz5,macierzuproszczona_dane1024_cz3,macierzuproszczona_dane1024_cz2] = skryptdopca(dane1024_cz1,dane1084_cz8,dane1084_cz6,dane1084_cz5,dane1084_cz4,dane1084_cz3,dane1084_cz2,dane1024_cz5,dane1024_cz3,dane1024_cz2);

```

```

f = waitbar(0,'Start','Name','Wyznacznik SVM',...

'CreateCancelBtn','setappdata(gcf,'canceling',1)');

setappdata(f,'canceling',0);

```

```

[a_dane1024_cz1,bdane1024_cz1] =
hist(dane1024_cz1(1025,:),unique(dane1024_cz1(1025,:)));

[a_dane1084_cz8,bdane1084_cz8] =
hist(dane1084_cz8(1085,:),unique(dane1084_cz8(1085,:)));

[a_dane1084_cz6,bdane1084_cz6] =
hist(dane1084_cz6(1085,:),unique(dane1084_cz6(1085,:)));

[a_dane1084_cz5,bdane1084_cz5] =
hist(dane1084_cz5(1085,:),unique(dane1084_cz5(1085,:)));

[a_dane1084_cz4,bdane1084_cz4] =
hist(dane1084_cz4(1085,:),unique(dane1084_cz4(1085,:)));

[a_dane1084_cz3,bdane1084_cz3] =
hist(dane1084_cz3(1085,:),unique(dane1084_cz3(1085,:)));

[a_dane1084_cz2,bdane1084_cz2] =
hist(dane1084_cz2(1085,:),unique(dane1084_cz2(1085,:)));

[a_dane1024_cz5,bdane1024_cz5] =
hist(dane1024_cz5(1025,:),unique(dane1024_cz5(1025,:)));

[a_dane1024_cz3,bdane1024_cz3] =
hist(dane1024_cz3(1025,:),unique(dane1024_cz3(1025,:)));

[a_dane1024_cz2,bdane1024_cz2] =
hist(dane1024_cz2(1025,:),unique(dane1024_cz2(1025,:)));

for j =1:5:size(bdane1024_cz1,2) - 5

    steps = size(bdane1024_cz1,2);

    if not(j/steps == 1)

        waitbar(j/steps,f, strcat(num2str(j),' z ',num2str(steps)))

    else

        waitbar(j/steps,f,sprintf('Koniec'))

    end

```

```

wektorpocz = find(dane1024_cz1(1025,:)==bdane1024_cz1(1,j),1,'first');

wektorkonc = find(dane1024_cz1(1025,:)==bdane1024_cz1(1,j+1),1,'last');

wekornastdane = find(dane1024_cz1(1025,:)==bdane1024_cz1(1,j+2),1,'first');


figure('Name',string(name(j,1)),'NumberTitle','off');

plot(cyclo830calibration(:,1),dane1024_cz1(1:1024,j), 'b')

xlabel('Przesunięcie ramanowskie cm-1');

ylabel('Natężenie promieniowania [J]');


[wynikrozpoznania_dane1024_cz1] =
funkcjasm2klasy(macierzuproszczona_dane1024_cz1(:,wektorpocz:wektorkonc),dane1024_cz1(1
025,wektorpocz:wektorkonc),macierzuproszczona_dane1024_cz1(:,wekornastdane:end),name(j,1),
name(j+1,1));

end

for j =1:5:size(bdane1084_cz8,2) - 5

    steps = size(dane1084_cz8(1085,:),2);

    if not(j/steps == 1)

        waitbar(j/steps,f, strcat(num2str(j), ' z ', num2str(steps)))

    else

        waitbar(j/steps,f,sprintf('Koniec'))

    end

    wektorpocz = find(dane1084_cz8(1085,:)==bdane1084_cz8(1,j),1,'first');

    if(size(wektorpocz,2) == 1)

        break;

    end
end

```

```

wektorkonc = find(dane1084_cz8(1085,:) == wektorpocz(1,j+1),1,'last');

wekornastdane = find(dane1084_cz8(1085,:) == wektorpocz(1,j+2),1,'first');

[wynikrozpoznania_dane1084_cz8] =
funkcjasm2klasy(macierzuproszczona_dane1084_cz8(:,wektorpocz:wektorkonc),dane1084_cz8(1
025,wektorpocz:wektorkonc),macierzuproszczona_dane1084_cz8(:,wekornastdane:end),name(j,1),
name(j+1,1));

figure('Name',string(name(j,1)),'NumberTitle','off');

plot(cyclo830calibration(:,1),dane1084_cz8(1:1024,j), 'b')

xlabel('Przesunięcie ramanowskie cm-1');

ylabel('Natężenie promieniowania [J]');

end

for j = 1:2:size(bdane1024_cz2,2)- 5

    steps = size(dane1024_cz2(1025,:),2);

    if not(j/steps == 1)

        waitbar(j/steps,f, strcat(num2str(j), ' z ', num2str(steps)))

    else

        waitbar(j/steps,f,sprintf('Koniec'))

    end

    wektorpocz = find(dane1024_cz2(1025,:) == bdane1024_cz2(1,j),1,'first');

    if(size(wektorpocz,2) == 1)

        break;

    end

    wektorkonc = find(dane1024_cz2(1025,:) == wektorpocz(1,j+1),1,'last');

    wekornastdane = find(dane1024_cz2(1025,:) == wektorpocz(1,j+2),1,'first');

```

```

[wynikrozpoznania_dane1024_cz2] =
funkcjasm2klasy(macierzuproszczona_dane1024_cz2(:,wektorpocz:wektorkonc),dane1024_cz2(1
025,wektorpocz:wektorkonc),macierzuproszczona_dane1024_cz2(:,wekornastdane:end),name(j,1),
name(j+1,1));

figure('Name',string(name(j,1)),'NumberTitle','off');

plot(cyclo830calibration(:,1),dane1084_cz8(1:1024,j), 'b')

xlabel('Przesunięcie ramanowskie cm-1');

ylabel('Natężenie promieniowania [J]');

end

for j =1:1:size(bdane1024_cz3,2) - 5

    steps = size(dane1024_cz3(1025,:),2);

    if not(j/steps == 1)

        waitbar(j/steps,f, strcat(num2str(j), ' z ', num2str(steps)))

    else

        waitbar(j/steps,f,sprintf('Koniec'))

    end

    wektorpocz = find(dane1024_cz3(1025,:)==bdane1024_cz3(1,j),1,'first');

    if(size(wektorpocz,2) == 1 || size(wektorpocz,2) == 0)

        break;

    end

    wektorkonc =find(dane1024_cz3(1025,:)==wektorpocz(1,j+1),1,'last');

    wekornastdane = find(dane1024_cz3(1025,:)==wektorpocz(1,j+2),1,'first');

[wynikrozpoznania_dane1024_cz3] =
funkcjasm2klasy(macierzuproszczona_dane1024_cz3(:,wektorpocz:wektorkonc),dane1024_cz3(1

```

```
025,wektorpocz:wektorkonc),macierzuproszczona_dane1024_cz3(:,wekornastdane:end),name(j,1),
name(j+1,1));
```

```
figure('Name',string(name(j,1)),'NumberTitle','off');

plot(cyclo830calibration(:,1),dane1084_cz8(1:1024,j), 'b')

xlabel('Przesunięcie ramanowskie cm-1');

ylabel('Natężenie promieniowania [J]');

end

for j =1:1:size(bdane1024_cz5) - 5

    steps = size(dane1024_cz5(1025,:),2);

    if not(j/steps == 1)

        waitbar(j/steps,f, strcat(num2str(j), ' z ', num2str(steps)))

    else

        waitbar(j/steps,f,sprintf('Koniec'))

    end

    wektorpocz = find(dane1024_cz5(1025,:)==bdane1024_cz5(1,j),1,'first');

    wektorkonc = find(dane1024_cz5(1025,:)==wektorpocz(1,j+1),1,'last');

    wekornastdane = find(dane1024_cz5(1025,:)==wektorpocz(1,j+2),1,'first');

    [wynikrozpoznania_dane1024_cz5] =
funkcjasm2klasy(macierzuproszczona_dane1024_cz5(:,wektorpocz:wektorkonc),dane1024_cz5(1
025,wektorpocz:wektorkonc),macierzuproszczona_dane1024_cz5(:,wekornastdane:end),name(j,1),
name(j+1,1));
```

```
figure('Name',string(name(j,1)),'NumberTitle','off');

plot(cyclo830calibration(:,1),dane1084_cz8(1:1024,j), 'b')

xlabel('Przesunięcie ramanowskie cm-1');
```



```

ylabel('Natężenie promieniowania [JJ]');

end

for j = 1:5:size(bdane1084_cz2,2) - 2

    steps = size(bdane1084_cz2,2);

    if not(j/steps == 1)

        waitbar(j/steps,f, strcat(num2str(j), ' z ', num2str(steps)))

    else

        waitbar(j/steps,f,sprintf('Koniec'))

    end

    wektorpocz = find(dane1084_cz2(1085,:) == bdane1084_cz2(1,j), 1, 'first');

    if(size(wektorpocz,2) == 1 || size(wektorpocz,2) == 0)

        break;

    end

    wektorkonc = find(dane1084_cz2(1085,:) == wektorpocz(1,j+1), 1, 'last');

    wekornastdane = find(dane1084_cz2(1085,:) == wektorpocz(1,j+2), 1, 'first');

    [wynikrozpoznania_dane1084_cz2] =
funkcjasm2klasy(macierzuproszczona_dane1084_cz2(:,wektorpocz:wektorkonc), dane1084_cz2(1
025,wektorpocz:wektorkonc), macierzuproszczona_dane1084_cz2(:,wekornastdane:end), name(j,1),
name(j+1,1));

    figure('Name', string(name(j,1)), 'NumberTitle', 'off');

    plot(cyclo830calibration(:,1), dane1084_cz8(1:1024,j), 'b')

    xlabel('Przesunięcie ramanowskie cm-1');

    ylabel('Natężenie promieniowania [JJ]');

end

```

```

for j = 1:5:size(bdane1084_cz3,2) - 2

    steps = size(bdane1084_cz3,2);

    if not(j/steps == 1)

        waitbar(j/steps,f, strcat(num2str(j), ' z ', num2str(steps)))

    else

        waitbar(j/steps,f,sprintf('Koniec'))

    end

    wektorpocz = find(dane1084_cz3(1085,:) == bdane1084_cz3(1,j), 1, 'first');

    if(size(wektorpocz,2) == 1 || size(wektorpocz,2) == 0)

        break;

    end

    wektorkonc = find(dane1084_cz3(1085,:) == wektorpocz(1,j+1), 1, 'last');

    wekornastdane = find(dane1084_cz3(1085,:) == wektorpocz(1,j+2), 1, 'first');

    [wynikrozpoznania_dane1084_cz3] =
funkcjasm2klasy(macierzuproszczona_dane1084_cz3(:,wektorpocz:wektorkonc), dane1084_cz3(1
025,wektorpocz:wektorkonc), macierzuproszczona_dane1084_cz3(:,wekornastdane:end), name(j,1),
name(j+1,1));

    figure('Name', string(name(j,1)), 'NumberTitle', 'off');

    plot(cyclo830calibration(:,1), dane1084_cz8(1:1024,j), 'b')

    xlabel('Przesunięcie ramanowskie cm-1');

    ylabel('Natężenie promieniowania [J]');

end

for j = 1:5:size(bdane1084_cz4,2) - 2

    steps = size(bdane1084_cz4,2);

    if not(j/steps == 1)

```

```

        waitbar(j/steps,f, strcat(num2str(j), ' z ', num2str(steps)))

    else

        waitbar(j/steps,f,sprintf('Koniec'))

    end

    wektorpocz = find(dane1084_cz4(1085,:) == bdane1084_cz4(1,j), 1, 'first');

    if(size(wektorpocz,2) == 1 || size(wektorpocz,2) == 0)

        break;

    end

    wektorkonc = find(dane1084_cz4(1085,:) == wektorpocz(1,j+1), 1, 'last');

    wekornastdane = find(dane1084_cz4(1085,:) == wektorpocz(1,j+2), 1, 'first');

    [wynikrozpoznania_dane1084_cz4] =
funkcjasm2klasy(macierzuproszczona_dane1084_cz4(:, wektorpocz:wektorkonc), dane1084_cz4(1
025, wektorpocz:wektorkonc), macierzuproszczona_dane1084_cz4(:, wekornastdane:end), name(j,1),
name(j+1,1));

figure('Name', string(name(j,1)), 'NumberTitle', 'off');

plot(cyclo830calibration(:,1), dane1084_cz8(1:1024,j), 'b')

xlabel('Przesunięcie ramanowskie cm-1');

ylabel('Natężenie promieniowania [J]');

end

for j = 1:5:size(bdane1084_cz5,2) - 2

    steps = size(bdane1084_cz5,2);

    if not(j/steps == 1)

        waitbar(j/steps,f, strcat(num2str(j), ' z ', num2str(steps)))

    else

        waitbar(j/steps,f,sprintf('Koniec'))

    end

```

```

        end

        wektorpocz = find(dane1084_cz5(1085,:) == bdane1084_cz5(1,j), 1, 'first');

        if(size(wektorpocz,2) == 1 || size(wektorpocz,2) == 0)

            break;

        end

        wektorkonc = find(dane1084_cz5(1085,:) == wektorpocz(1,j+1), 1, 'last');

        wekornastdane = find(dane1084_cz5(1085,:) == wektorpocz(1,j+2), 1, 'first');

        [wynikrozpoznania_dane1084_cz5] =
funkcjasm2klasy(macierzuproszczona_dane1084_cz5(:,wektorpocz:wektorkonc), dane1084_cz5(1
025,wektorpocz:wektorkonc), macierzuproszczona_dane1084_cz5(:,wekornastdane:end), name(j,1),
name(j+1,1));

        figure('Name', string(name(j,1)), 'NumberTitle', 'off');

        plot(cyclo830calibration(:,1), dane1084_cz8(1:1024,j), 'b')

        xlabel('Przesunięcie ramanowskie cm-1');

        ylabel('Natężenie promieniowania [J]');

    end

    for j = 1:size(bdane1084_cz6,2) - 5

        steps = size(bdane1084_cz6,2);

        if not(j/steps == 1)

            waitbar(j/steps, f, strcat(num2str(j), ' z ', num2str(steps)))

        else

            waitbar(j/steps, f, sprintf('Koniec'))

        end

        wektorpocz = find(dane1084_cz6(1085,:) == bdane1084_cz6(1,j), 1, 'first');

        if(size(wektorpocz,2) == 1 || size(wektorpocz,2) == 0)

```

```

        break;

    end

    wektorkonc = find(dane1084_cz6(1085,:) == wektorpocz(1,j+1),1,'last');

    wekornastdane = find(dane1084_cz6(1085,:) == wektorpocz(1,j+2),1,'first');

    [wynikrozpoznania_dane1084_cz6] =
    funkcjasvm2klasy(macierzuproszczona_dane1084_cz6(:,wektorpocz:wektorkonc),dane1084_cz6(1
    025,wektorpocz:wektorkonc),macierzuproszczona_dane1084_cz6(:,wekornastdane:end),name(j,1),
    name(j+1,1));

    figure('Name',string(name(j,1)),'NumberTitle','off');

    plot(cyclo830calibration(:,1),dane1084_cz8(1:1024,j), 'b')

    xlabel('Przesunięcie ramanowskie cm-1');
    ylabel('Natężenie promieniowania [J]');

    end

    delete(f)

```

11.3. **Kod źródłowy funkcji kompresującej PCA**

```

function
[macierzuproszczona_dane1024_cz1,macierzuproszczona_dane1084_cz8,macierzuproszczona_d
ane1084_cz6,macierzuproszczona_dane1084_cz5,macierzuproszczona_dane1084_cz4,macierzup
roszczona_dane1084_cz3,macierzuproszczona_dane1084_cz2,macierzuproszczona_dane1024_c
z5,macierzuproszczona_dane1024_cz3,macierzuproszczona_dane1024_cz2] =
skryptdopca(dane1024_cz1,dane1084_cz8,dane1084_cz6,dane1084_cz5,dane1084_cz4,dane108
4_cz3,dane1084_cz2,dane1024_cz5,dane1024_cz3,dane1024_cz2)

%okienko z postępem

f = waitbar(0,'Start','Name','Kompresowanie PCA',...

    'CreateCancelBtn','setappdata(gcf,"canceling",1)');

```

```

setappdata(f,'canceling',0);

%Obliczenie PCA, uzyskujemy macierze z informacjami jakie wektory mają
%największa wartość

[coeff_dane1024_cz1,score_dane1024_cz1,latent_dane1024_cz1] =
PCA(dane1024_cz1(1:1024,1));

[coeff_dane1084_cz8,score_dane1084_cz8,latent_dane1084_cz8] =
PCA(dane1084_cz8(1:1084,1));

[coeff_dane1084_cz6,score_dane1084_cz6,latent_dane1084_cz6] =
PCA(dane1084_cz6(1:1084,1));

[coeff_dane1084_cz5,score_dane1084_cz5,latent_dane1084_cz5] =
PCA(dane1084_cz5(1:1084,1));

[coeff_dane1084_cz4,score_dane1084_cz4,latent_dane1084_cz4] =
PCA(dane1084_cz4(1:1084,1));

[coeff_dane1084_cz3,score_dane1084_cz3,latent_dane1084_cz3] =
PCA(dane1084_cz3(1:1084,1));

[coeff_dane1084_cz2,score_dane1084_cz2,latent_dane1084_cz2] =
PCA(dane1084_cz2(1:1084,1));

[coeff_dane1024_cz5,score_dane1024_cz5,latent_dane1024_cz5] =
PCA(dane1024_cz5(1:1024,1));

[coeff_dane1024_cz3,score_dane1024_cz3,latent_dane1024_cz3] =
PCA(dane1024_cz3(1:1024,1));

[coeff_dane1024_cz2,score_dane1024_cz2,latent_dane1024_cz2] =
PCA(dane1024_cz2(1:1024,1));

%sortujemy od największego do najmniejszego

[scoreposortowne_dane1024_cz1,          indexscore_dane1024_cz1] =
sort(score_dane1024_cz1,'descend');

```

[scoreposortowne_dane1084_cz8, sort(score_dane1084_cz8,'descend');	indexscore_dane1084_cz8]	=
[scoreposortowne_dane1084_cz6, sort(score_dane1084_cz6,'descend');	indexscore_dane1084_cz6]	=
[scoreposortowne_dane1084_cz5, sort(score_dane1084_cz5,'descend');	indexscore_dane1084_cz5]	=
[scoreposortowne_dane1084_cz4, sort(score_dane1084_cz4,'descend');	indexscore_dane1084_cz4]	=
[scoreposortowne_dane1084_cz3, sort(score_dane1084_cz3,'descend');	indexscore_dane1084_cz3]	=
[scoreposortowne_dane1084_cz2, sort(score_dane1084_cz2,'descend');	indexscore_dane1084_cz2]	=
[scoreposortowne_dane1024_cz5, sort(score_dane1024_cz5,'descend');	indexscore_dane1024_cz5]	=
[scoreposortowne_dane1024_cz3, sort(score_dane1024_cz3,'descend');	indexscore_dane1024_cz3]	=
[scoreposortowne_dane1024_cz2, sort(score_dane1024_cz2,'descend');	indexscore_dane1024_cz2]	=
%układamy w macierz		
wiedzapca_dane1024_cz1		=
[indexscore_dane1024_cz1,scoreposortowne_dane1024_cz1];		
wiedzapca_dane1084_cz8		=
[indexscore_dane1084_cz8,scoreposortowne_dane1084_cz8];		
wiedzapca_dane1084_cz6		=
[indexscore_dane1084_cz6,scoreposortowne_dane1084_cz6];		
wiedzapca_dane1084_cz5		=
[indexscore_dane1084_cz5,scoreposortowne_dane1084_cz5];		
wiedzapca_dane1084_cz4		=
[indexscore_dane1084_cz4,scoreposortowne_dane1084_cz4];		

wiedzapca_dane1084_cz3 =
[indexscore_dane1084_cz3,scoreposortowne_dane1084_cz3];

wiedzapca_dane1084_cz2 =
[indexscore_dane1084_cz2,scoreposortowne_dane1084_cz2];

wiedzapca_dane1024_cz5 =
[indexscore_dane1024_cz5,scoreposortowne_dane1024_cz5];

wiedzapca_dane1024_cz3 =
[indexscore_dane1024_cz3,scoreposortowne_dane1024_cz3];

wiedzapca_dane1024_cz2 =
[indexscore_dane1024_cz2,scoreposortowne_dane1024_cz2];

najwarznieszewektory_dane1024_cz1 = wiedzapca_dane1024_cz1(1:256,:);

najwarznieszewektory_dane1084_cz8 = wiedzapca_dane1084_cz8(1:256,:);

najwarznieszewektory_dane1084_cz6 = wiedzapca_dane1084_cz6(1:256,:);

najwarznieszewektory_dane1084_cz5 = wiedzapca_dane1084_cz5(1:256,:);

najwarznieszewektory_dane1084_cz4 = wiedzapca_dane1084_cz4(1:256,:);

najwarznieszewektory_dane1084_cz3 = wiedzapca_dane1084_cz3(1:256,:);

najwarznieszewektory_dane1084_cz2 = wiedzapca_dane1084_cz2(1:256,:);

najwarznieszewektory_dane1024_cz5 = wiedzapca_dane1024_cz5(1:256,:);

najwarznieszewektory_dane1024_cz3 = wiedzapca_dane1024_cz3(1:256,:);

najwarznieszewektory_dane1024_cz2 = wiedzapca_dane1024_cz2(1:256,:);

macierzuproszczona_dane1024_cz1 = zeros(256,1);

macierzuproszczona_dane1084_cz8 = zeros(256,1);

macierzuproszczona_dane1084_cz6 = zeros(256,1);

macierzuproszczona_dane1084_cz5 = zeros(256,1);

macierzuproszczona_dane1084_cz4 = zeros(256,1);


```

macierzuproszczona_dane1084_cz3 = zeros(256,1);
macierzuproszczona_dane1084_cz2 = zeros(256,1);
macierzuproszczona_dane1024_cz5 = zeros(256,1);
macierzuproszczona_dane1024_cz3 = zeros(256,1);
macierzuproszczona_dane1024_cz2 = zeros(256,1);

```

%W pętli tworzymy dane z ograniczono liczbą wymiarów wektorów do 256

```

for j = 1 : size(dane1024_cz2,2)

    steps = size(dane1024_cz2,2);

    if not(j/steps == 1)

        waitbar(j/steps,f, strcat(num2str(j), ' z ', num2str(steps)))

    else

        waitbar(j/steps,f, sprintf('Koniec'))

    end

for i = 1 : 256

    row_dane1024_cz2 = najwarniejszevektory_dane1024_cz2(i,1);

    macierzuproszczona_dane1024_cz2(i,j) = dane1024_cz2(row_dane1024_cz2,j);

    if getappdata(f,'canceling')

        break

    end

end

```

```

end

for j = 1 : size(dane1084_cz3,2)

    steps = size(dane1084_cz3,2);

    if not(j/steps == 1)

        waitbar(j/steps,f, strcat(num2str(j), ' z ', num2str(steps)))

    else

        waitbar(j/steps,f,sprintf('Koniec'))

    end

end

for i = 1 : 256

    row_dane1084_cz3 = najwarzniejszewektory_dane1084_cz3(i,1);

    macierzuproszczona_dane1084_cz3(i,j) = dane1084_cz3(row_dane1084_cz3,j);

    if getappdata(f,'canceling')

        break

    end

end

end

for j = 1 : size(dane1084_cz2,2)

    steps = size(dane1084_cz2,2);

    if not(j/steps == 1)

        waitbar(j/steps,f, strcat(num2str(j), ' z ', num2str(steps)))

    else

        waitbar(j/steps,f,sprintf('Koniec'))

    end

end

```

```

for i = 1 : 256

    row_dane1084_cz2 = najwarniejszevektory_dane1084_cz2(i,1);

    macierzuproszczona_dane1084_cz2(i,j) = dane1084_cz2(row_dane1084_cz2,j);

    if getappdata(f,'canceling')
        break
    end

end

end

for j = 1 : size(dane1024_cz1,2)

    steps = size(dane1024_cz1,2);

    if not(j/steps == 1)

        waitbar(j/steps,f, strcat(num2str(j),' z ',num2str(steps)))

    else

        waitbar(j/steps,f,sprintf('Koniec'))

    end

end

for i = 1 : 256

    row_dane1024_cz1 = najwarniejszevektory_dane1024_cz1(i,1);

    macierzuproszczona_dane1024_cz1(i,j) = dane1024_cz1(row_dane1024_cz1,j);

    if getappdata(f,'canceling')
        break
    end

end

end

```

```

end

for j = 1 : size(dane1084_cz8,2)

    steps = size(dane1084_cz8,2);

    if not(j/steps == 1)

        waitbar(j/steps,f, strcat(num2str(j),' z ',num2str(steps)))

    else

        waitbar(j/steps,f,sprintf('Koniec'))

    end

for i = 1 : 256

    row_dane1084_cz8 = najwarzniejszewektory_dane1084_cz8(i,1);

    macierzuproszczona_dane1084_cz8(i,j) = dane1084_cz8(row_dane1084_cz8,j);

    if getappdata(f,'canceling')

        break

    end

end

end

for j = 1 : size(dane1084_cz6,2)

    steps = size(dane1084_cz6,2);

    if not(j/steps == 1)

        waitbar(j/steps,f, strcat(num2str(j),' z ',num2str(steps)))

    else

        waitbar(j/steps,f,sprintf('Koniec'))

    end

end

for i = 1 : 256

```

```

row_dane1084_cz6 = najwarzniejszewektory_dane1084_cz6(i,1);

macierzuproszczona_dane1084_cz6(i,j) = dane1084_cz6(row_dane1084_cz6,j);


if getappdata(f,'canceling')

break

end

end

end

for j = 1 : size(dane1084_cz5,2)

steps = size(dane1084_cz5,2);

if not(j/steps == 1)

waitbar(j/steps,f, strcat(num2str(j), ' z ', num2str(steps)))

else

waitbar(j/steps,f,sprintf('Koniec'))

end

for i = 1 : 256

row_dane1084_cz5 = najwarzniejszewektory_dane1084_cz5(i,1);

macierzuproszczona_dane1084_cz5(i,j) = dane1084_cz5(row_dane1084_cz5,j);


if getappdata(f,'canceling')

break

end

end

end

end

```

```

for j = 1 : size(dane1084_cz4,2)

    steps = size(dane1084_cz4,2);

    if not(j/steps == 1)

        waitbar(j/steps,f, strcat(num2str(j), ' z ', num2str(steps)))

    else

        waitbar(j/steps,f,sprintf('Koniec'))

    end

for i = 1 : 256

    row_dane1084_cz4 = najwarzniejszewektory_dane1084_cz4(i,1);

    macierzuproszczona_dane1084_cz4(i,j) = dane1084_cz4(row_dane1084_cz4,j);

    if getappdata(f,'canceling')

        break

    end

end

end

for j = 1 : size(dane1024_cz5,2)

    steps = size(dane1024_cz5,2);

    if not(j/steps == 1)

        waitbar(j/steps,f, strcat(num2str(j), ' z ', num2str(steps)))

    else

        waitbar(j/steps,f,sprintf('Koniec'))

    end

end

```

```

for i = 1 : 256

    row_dane1024_cz5 = najwarzniejszewektory_dane1024_cz5(i,1);

    macierzuproszczona_dane1024_cz5(i,j) = dane1024_cz5(row_dane1024_cz5,j);


    if getappdata(f,'canceling')

        break

    end

end

end

for j = 1 : size(dane1024_cz3,2)

    steps = size(dane1024_cz3,2);

    if not(j/steps == 1)

        waitbar(j/steps,f, strcat(num2str(j), ' z ', num2str(steps)))

    else

        waitbar(j/steps,f,sprintf('Koniec'))

    end

end


for i = 1 : 256

    row_dane1024_cz3 = najwarzniejszewektory_dane1024_cz3(i,1);

    macierzuproszczona_dane1024_cz3(i,j) = dane1024_cz3(row_dane1024_cz3,j);


    if getappdata(f,'canceling')

        break

    end

end

end

```

```

end

delete(f)

end

```

11.4. *Kod funkcji wykreślającej i obliczającej SVM*

```

function [wynikrozpoznania] =
funkcjasm2klasy(daneklas,wektorklas,daneklasdosprawdzenia,name1,name2)

%Dane muszą być transponowane ze względu na inne działanie PCA wbudowane,
%LSSVM pobiera wiersze jako dane, kolumnę z nazwami klas

X = daneklas .';
Y = wektorklas .';
Xt = daneklasdosprawdzenia .';

type = 'classification';

gam = 10;
sig2 = 0.3;

%w komentarzu są jeszcze inne możliwości trenowania sieci

[alpha,b] = trainlssvm({X,Y,type,gam,sig2,'RBF_kernel'});
%[alpha,b] = trainlssvm({X,Y,type,gam,sig2,'RBF_kernel','original'});
%[alpha,b] = trainlssvm({X,Y,type,gam,sig2,'RBF_kernel','preprocess'});

%Wykreślenie zmienności na wykresie, tak aby ludzkie oko mogło rozpoznać,
%dodatkowo tablica wynik rozpoznania przechowuje rozpoznane z modelu dane

```



```
%wprowadzone do analizy

wynikrozpoznania = simlssvm({X,Y,type,gam,sig2,'RBF_kernel','preprocess'},{alpha,b},Xt);

figure('Name',strcat(string(name1),'_1class____2class_',string(name2)),'NumberTitle','off');

plotlssvm({X,Y,type,gam,sig2,'RBF_kernel','original'},{alpha,b});

end
```