

# WOJSKOWA AKADEMIA TECHNICZNA

im. Jarosława Dąbrowskiego



Rozprawa doktorska

## **Wybrane metody uczenia maszynowego w zadaniach wykrywania anomalii procesów**

*mgr inż. Maciej Gołgowski*

Promotor:

prof. dr hab. inż. Stanisław Osowski

Warszawa 2023



## Streszczenie

Rozprawa doktorska poświęcona jest opracowaniu systemów wykrywania anomalii procesów na podstawie zarejestrowanych sygnałów. Zaproponowano zastosowanie wielu rozwiązań klasyfikatorów współpracujących ze sobą w zespole dla wypracowania optymalnej decyzji (poprzez głosowanie większościowe). W wyniku analizy teoretycznej zaproponowano dwa rodzaje rozwiązań zespołu. Jedno z nich bazuje na zastosowaniu klasyfikatorów klasycznych (płytkich), w tym las losowy drzew decyzyjnych, klasyfikator typu gradient boosting, perceptron wielowarstwowy, maszyna wektorów nośnych SVM, klasyfikator K najbliższych sąsiadów, naiwny klasyfikator Bayesa oraz klasyfikator bazujący na procesach gaussowskich. Drugie rozwiązanie wykorzystuje zespół bazujący na strukturach głębokich klasyfikatorów CNN. Dla uzyskania najlepszych wyników rozpoznania anomalii od procesu normalnego ważną rolę odgrywa wstępne przetworzenie oryginalnych szeregów czasowych reprezentujących proces w zestaw atrybutów wejściowych dla zespołu klasyfikatorów. W pracy podstawę takiego preprocesingu danych stanowi transformacja falkowa, zarówno typu ciągłego (CWT) jak i dyskretnego (DWT).

Zaproponowane rozwiązania zostały przetestowane na trzech rodzajach problemów. Dwa z nich bazują na zarejestrowanych szeregach czasowych i dotyczą wykrywania anomalii w sygnałach EKG oraz uszkodzenia łożysk tocznych. Trzeci problem związany jest z wykrywaniem podróbek obrazów typu deep fake wyekstrahowanych z filmów video. Przebadane zostały różne warianty doboru parametrów obu systemów, uzyskując w efekcie bardzo dobre wyniki wykrycia anomalii, lepsze lub porównywalne z najlepszymi rezultatami prezentowanymi w literaturze światowej.

**Słowa kluczowe:** wykrywanie anomalii, transformacja falkowa, klasyfikacja, deep fake, sieci neuronowe

## **Abstract**

The PhD thesis was directed to develop algorithms, based on the measured signals, for recognizing the anomaly processes. As a result of research two types of computer systems, applying the ensemble of classifiers have been elaborated. The first one is based on application of many classical (shallow) classifiers, including random forest, gradient boosting classifier, multilayer perceptron, support vector machine, K nearest neighbor classifier, naïve Bayes classifier and Gaussian process classifier. The second type of ensemble was composed on deep learning classifiers of different architectures realizing convolutional neural networks. Irrespective of the type of ensemble the input attributes for the classifiers are formed based on wavelet transformation: continuous wavelet (CWT) and discrete wavelet (DWT).

The proposed solutions of the systems have been checked on three types of anomalies. The first two (EKG rhythms and bearing failures) are characterized by the measured time series. The third task of discovering anomaly relates to the deep fake image recognition, which are extracted from video films. The numerous numerical experiments performed within these three tasks have confirmed good efficiency of the developed systems, of the accuracy comparable to the best solutions presented in the scientific papers published in journals and international conferences.

**Keywords:** anomaly detection, wavelet transform, classification, deep fake, neural networks

# Spis treści

Streszczenie.....	3
Abstract.....	4
1 Wstęp .....	7
1.1 Pojęcie anomalii procesu .....	7
1.2 Stosowane metody wykrywania anomalii procesów .....	8
1.3 Cel i przegląd zawartości pracy .....	13
2 Metody uczenia maszynowego stosowane w pracy .....	17
2.1 Wprowadzenie do metod uczenia maszynowego.....	17
2.2 Wstępne przetworzenie danych z użyciem transformacji falkowej .....	17
2.3 Modele klasyfikatorów zastosowane w pracy .....	23
3 Wykrywanie anomalii w sygnałach EKG.....	36
3.1 Wstęp do przetwarzania sygnałów EKG .....	36
3.2 Baza danych zastosowana w eksperymentach.....	38
3.3 Generacja cech diagnostycznych sygnału EKG z wykorzystaniem parametrów statystycznych.....	39
3.4 Wyniki zastosowania modeli płytkich w wykrywaniu anomalii EKG.....	46
3.5 Sieci głębokie w wykrywaniu anomalii.....	49
4 Wykrywanie uszkodzeń łożysk .....	55
4.1 Wprowadzenie do analizy uszkodzeń.....	55
4.2 Baza danych w eksperymentach.....	56
4.3 Analiza częstotliwościowa sygnałów uszkodzeń łożysk.....	58
4.4 Zastosowanie zespołu klasyfikatorów płytkich w wykrywaniu uszkodzeń .....	60
4.5 Zastosowanie sieci głębokich w wykrywaniu anomalii łożysk.....	65
5 Wykrywanie anomalii typu „ <i>deep fake</i> ” w obrazach.....	70

5.1	Definicja problemu <i>deep fake</i> .....	70
5.2	Baza danych użytych w eksperymentach .....	72
5.3	Detekcja obrazu twarzy z klatki video .....	77
5.4	Proponowana procedura wykrywania obrazów twarzy typu <i>deep fake</i> .....	82
5.5	Wyniki eksperymentów numerycznych .....	88
6	Podsumowanie i wnioski końcowe.....	93

# 1 Wstęp

## 1.1 Pojęcie anomalii procesu

Anomalia procesu jest utożsamiana z wystąpieniem nietypowego zbioru wartości obserwowanych zmiennych w stosunku do zbioru uważanego za normalny. W przypadku szeregów czasowych, jeśli są to pojedyncze wartości odstające od sąsiadów, noszą one zwykle angielskojęzyczną nazwę „*outliers*”. Anomalie mogą dotyczyć również obrazów. Typowym przykładem jest obraz typu „*deep fake*” powstały z obrazu oryginalnego poprzez zastąpienie wybranej grupy pikseli poprzez inne wartości.

Przyjmuje się, że procesy naturalne, obserwowane w postaci danych pomiarowych, są rezultatem pewnych reguł i praw występujących w naturze. Na podstawie obserwowanych wartości formułowane są pewne hipotezy weryfikowane poprzez pomiary. Dopóki pomiary potwierdzają przyjętą hipotezę to proces jest uważany za normalny. Wystąpienie zmian obserwowanych wartości w stosunku do oczekiwanych, według przyjętej hipotezy, prowadzi do stanu procesu, który może być uznany za stan anomalny. Problemem głównym w wykryciu anomalii jest brak jednoznacznej definicji progu, umożliwiającej rozróżnienie procesu normalnego od anormalnego.

W praktyce, detekcja anomalii polega na porównaniu aktualnych wartości procesu z ich normalnym zachowaniem w przeszłości. Podstawowym założeniem jest przyjęcie stacjonarności procesu, zakładającym niewielkie (akceptowalne) zmiany parametrów opisujących proces, co oznacza powtarzalność (z określoną tolerancją) wartości w określonym przedziale czasowym. Przy braku wystąpienia anomalii, zakłada się, że parametry statystyczne opisujące proces z przeszłości mają zastosowanie również do przyszłości.

Naturalnym rozwiązaniem wydaje się zastosowanie metod klasyfikacji z użyciem uczenia maszynowego do wykrycia anomalii. Takie rozwiązanie niesie wiele dodatkowych problemów. Pierwszym wyzwaniem jest zwykle brak zrównoważenia populacji reprezentujących proces normalny od anomalnego (przykłady anomalii są zdecydowanie rzadsze). Innym problemem jest zwiększona różnorodność rodzajów anomalii w stosunku do przypadków normalnych. Częstość przypadkiem jest częściowe pokrywanie się zarejestrowanych przebiegów anomalnych z normalnymi. Dodatkowe pytania związane z tym problemem to:

- jak scharakteryzować przebiegi normalne, zwłaszcza jeśli dotyczą różnych przypadków uważanych za normalne (różne klasy przynależności),

- jak określić dopuszczalne zakresy zmienności przebiegów normalnych w stosunku do przebiegów anormalnych.

W efekcie system automatycznego wykrywania anomalii może wygenerować wyniki obarczone błędami, na które składają się rozpoznania fałszywie pozytywne (rozpoznanie procesu normalnego jako anomalii) lub fałszywie negatywne (rozpoznanie anomalii jako procesu normalnego). Stąd popularne stało się podejście rozmyte, przypisujące każdemu analizowanemu procesowi wartości prawdopodobieństwa przynależności do procesu anormalnego. W ten sposób można segregować stopień anomalności obserwowanych przypadków, bez przyjęcia jednoznacznego progu anormalności [1].

W statystyce do określenia normalności procesu typu gaussowskiego są używane takie parametry jak wartość średnia, mediana czy moda, traktowane jako punkty odniesienia. Każdy pomiar może być reprezentowany przez pojedynczy parametr bądź przez ich zbiór w postaci wektorowej. Odległości pomiędzy aktualnie zmierzonymi wartościami procesu reprezentowanymi przez wybrane parametry statystyczne mogą być miarą anomalności. W przypadku rozkładów wielomodalnych należy wyróżnić wiele punktów odniesienia. Przykładem mogą tu być klastry danych reprezentowane poprzez ich centra. Położenie aktualnie zmierzonych danych (traktowanych jako punkt w przestrzeni wielowymiarowej) może być uznane za anomalne, gdy tak zdefiniowany punkt jest odległy od innych pozostałych punktów we wszystkich klastrach. Rozpatrując problem anomalii można uwzględnić wiele jego aspektów:

- jaka jest relacja aktualnego pomiaru względem tych z przeszłości (problem wykrycia wartości odstających od tych z przeszłości),
- jaka jest relacja aktualnego pomiaru względem pomiarów tworzących klastry, w szczególności względem brzegów danego klastra.

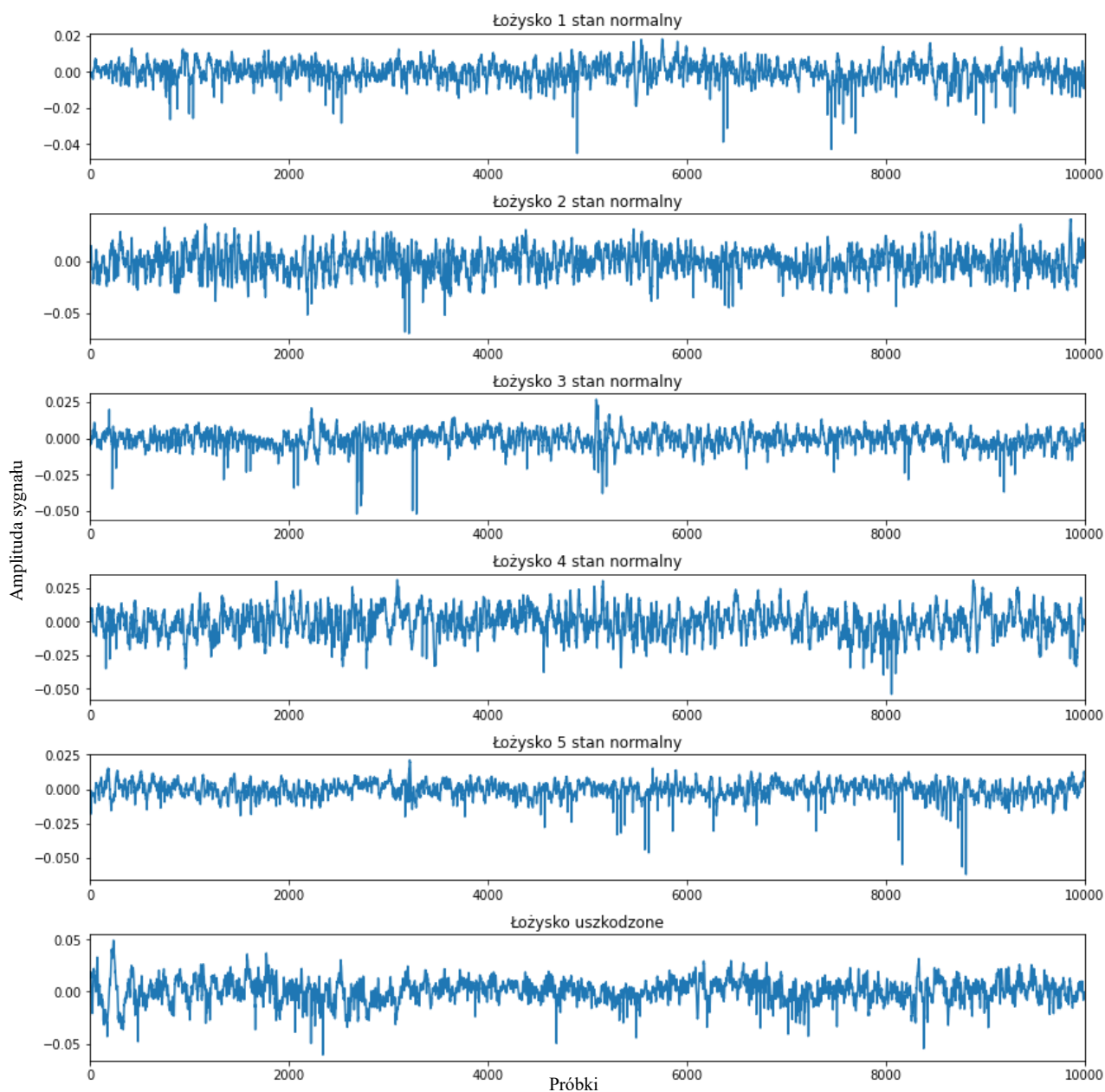
## **1.2 Stosowane metody wykrywania anomalii procesów**

Opracowano wiele metod wykrywania anomalii procesów. Metody te bazują bądź na zasadzie odległościowej (punkty dalekie od pozostałych można uznać za bardziej anomalne), bądź na podstawie gęstości rozkładu (mniejsza gęstość rozkładu danych w obszarze utożsamiana jest z większym stopniem anomalności).

Jako przykład rozpatrzmy szeregi czasowe reprezentujące sygnały drgań łożysk tocznych przedstawione na rys. 1.1. Pierwsze 5 sygnałów reprezentują stan normalny łożyska, a ostatni



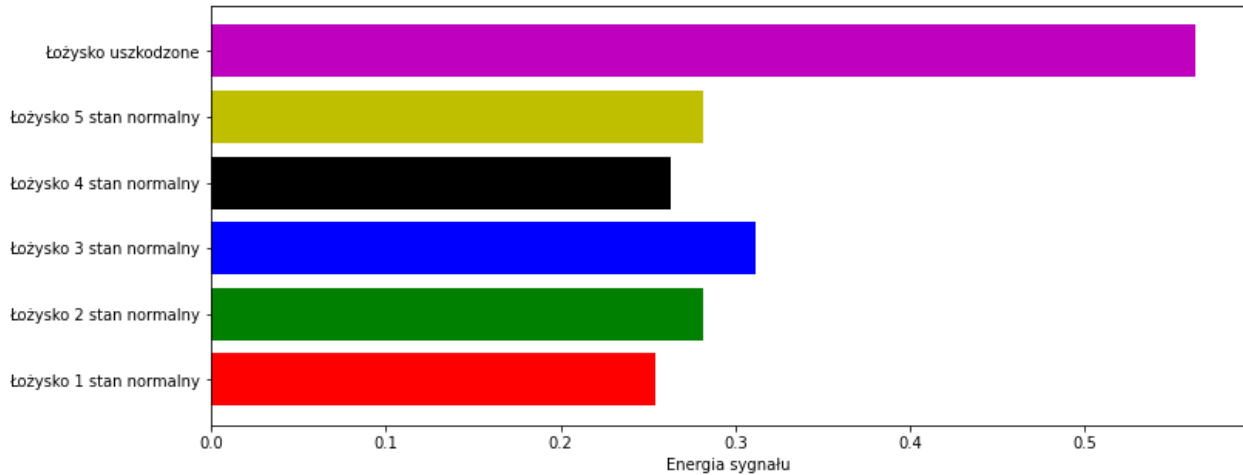
wykres dla łożyska uszkodzonego. Jako parametr charakteryzujący ciąg pomiarowy rozpatrzmy w pierwszej kolejności pojedynczy parametr statystyczny reprezentujący energię sygnału.



Rys. 1.1 Sygnały pomiarowe łożyska tocznego: pierwsze 5 sygnałów od góry reprezentują łożysko w stanie normalnym, a sygnał szósty łożysko z uszkodzonym elementem tocznym oraz jednocześnie uszkodzonymi bieżniami wewnętrzną i zewnętrzną.

Na rys. 1.2 przedstawiono stan poszczególnych obserwacji biorąc pod uwagę jedynie energię sygnału (suma kwadratów wartości elementów ciągu czasowego). Widoczne jest bliskie

sąsiedztwo obserwacji należących do stanu normalnego i znaczna odległość w przypadku wystąpienia uszkodzenia.



Rys. 1.2 Pozycje danych dotyczących łożyska w stanie normalnym (ostatnie 5 prążków) i łożyska uszkodzonego (pierwszy prążek). Szeregi czasowe są reprezentowane poprzez wartości ich energii.

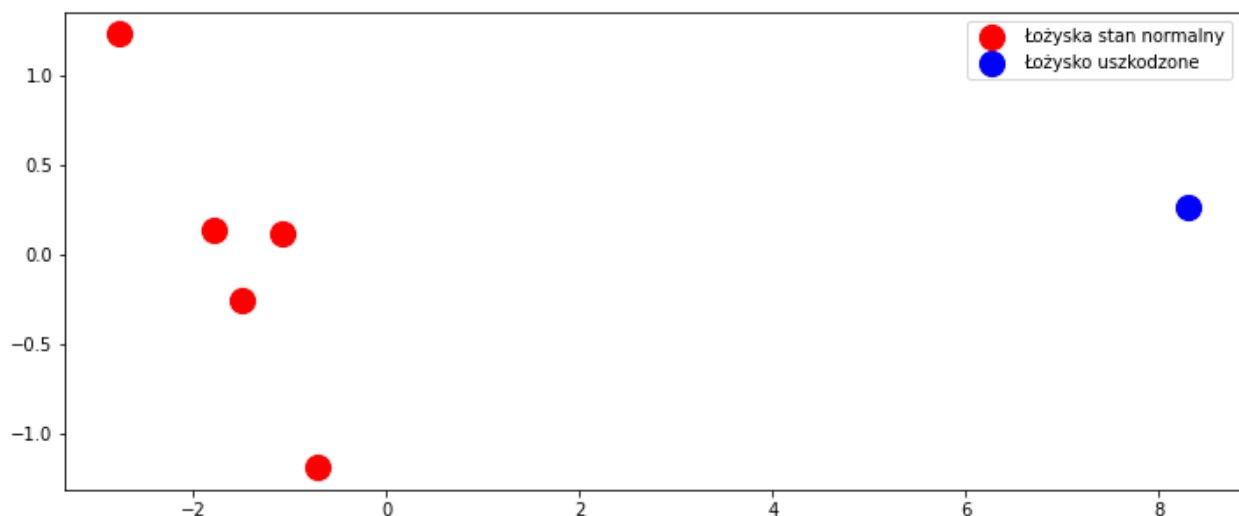
W powyższym przypadku sygnały poddane analizie były reprezentowane przez jeden parametr statystyczny (energię). Więcej informacji o sygnałach niesie zbiór wielu parametrów statystycznych. Scharakteryzujmy powyższe sygnały poprzez wektor zawierający 5 statystyk: wartość średnią, odchylenie standardowe, energię, skośność oraz kurtozę. Stan normalny łożyska jest reprezentowany teraz przez wektor średni z pięciu obserwacji. Tabela 1.1 przedstawia odległości poszczególnych pomiarów  $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5$  (stan normalny łożyska) oraz wektora  $\mathbf{x}_6$  (łożysko uszkodzone) od centrum klastra  $\mathbf{c}$  (wektora średniego) w sensie normy Euklidesa.

Tabela 1.1 Odległości poszczególnych pomiarów  $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5$  (stan normalny łożyska) oraz wektora  $\mathbf{x}_6$  (łożysko uszkodzone) od centrum klastra.

$\ \mathbf{x}_1 - \mathbf{c}\ $	$\ \mathbf{x}_2 - \mathbf{c}\ $	$\ \mathbf{x}_3 - \mathbf{c}\ $	$\ \mathbf{x}_4 - \mathbf{c}\ $	$\ \mathbf{x}_5 - \mathbf{c}\ $	$\ \mathbf{x}_6 - \mathbf{c}\ $
1.8917	3.9266	2.2376	2.7171	2.6155	11.2607

Również przy tej reprezentacji szeregów czasowych widoczne jest znaczne oddalenie sygnału reprezentującego anomalie od sygnałów zmierzonych dla stanu normalnego. Różnica ta jest dobrze odzwierciedlona w sposób graficzny poprzez rzutowanie danych oryginalnych (5- wymiarowych) na dwa wymiary przy zastosowaniu rzutowania PCA [2]. Wynik takiego rzutowania przedstawiony

jest na rys. 1.3. Widoczne jest znaczne oddalenie punktu reprezentującego stan uszkodzony łożyska od punktów reprezentujących stan normalny.



Rys. 1.3 Położenia punktów pomiarowych łożyska na płaszczyźnie po zrzutowaniu wektorów 5-wymiarowych na dwa wymiary przy użyciu transformacji PCA. Punkt oznaczony kolorem niebieskim reprezentuje łożysko uszkodzone, punkty reprezentowane przez kolor czerwony dotyczą stanu normalnego.

Miary odległościowe zastosowane w powyższym przykładzie reprezentują podejście samoorganizujące (bez nauczyciela). Klaster jest automatycznie tworzony przez bliskie sobie dane, biorąc pod uwagę odległości między poszczególnymi wektorami.

W praktyce stosowane są różne miary odległościowe, na podstawie których ocenia się stopień anomalności procesu reprezentowanego przez punkt w przestrzeni wielowymiarowej w stosunku do danych reprezentowanych w klastrach powstałych w wyniku grupowania. Rozważane są między innymi takie podejścia jak: odległość do wszystkich punktów poszczególnych klastrów, odległość do najbliższego lub K najbliższych sąsiadów, odległość od centrum klastra danych, odległość od granicznych elementów klastra, itp. Proces grupowania danych może wykorzystywać różne techniki, w tym ostre podejście *K-means*, rozmyte *c-means* czy grupowanie hierarchiczne [3].

Częściej stosowane jest podejście wykorzystujące informację dotyczącą potencjalnej przynależności danych wejściowych do konkretnej klasy. Jest to tak zwane uczenie z nauczycielem realizowane przez określony rodzaj klasyfikatora. System oceny anomalności bazuje na matematycznym modelu procesu. W tym podejściu dane wejściowe są reprezentowane przez parę wektorów ( $\mathbf{x}$ ,  $\mathbf{d}$ ), gdzie  $\mathbf{x}$  oznacza wektor wejściowy, a  $\mathbf{d}$  reprezentuje klasę (na przykład stan normalny bądź anomalny związany z konkretnym rodzajem uszkodzenia). Stosowane są różne

modele klasyfikatorów, poczynając od KNN (ang. *K nearest neighbors*), klasyfikatory bayesowskie, drzewa decyzyjne, sieci neuronowe MLP (ang. *multilayer perceptron*), sieci RBF (ang. *radial basis function*), sieci SVM (ang. *support vector machine*), czy liczne rozwiązania konwolucyjnych sieci głębokich CNN (ang. *Convolutional Neural Networks*) [3, 4, 5].

Przypisanie danego pomiaru do stanu normalnego bądź anomalnego przez model klasyfikatora może być ostre (stopień przynależności do klasy równy 1, a brak przynależności oznaczony jako 0) lub rozmyte (stopień przynależności do klasy z przedziału  $[0,1]$ ). W drugim przypadku operuje się prawdopodobieństwem przynależności do klasy (większa wartość prawdopodobieństwa oznacza większy stopień anomalności procesu). Taki sposób postępowania pozwala uszeregować stopień anomalności, na podstawie którego użytkownik uzyskuje dodatkową wiedzę o analizowanym procesie.

Wykrywanie anomalności bazujące na modelu procesu może bazować na przestrzeni parametrów modelu bądź na danych uczących [1]. W pierwszym przypadku ocenia się jak pojedyncza para ucząca zmienia wartości parametrów. Duża zmiana tych wartości utożsamiana jest z anomalnością aktualnego pomiaru. Przy wielu obserwowanych parametrach sumuje się wartości absolutne wszystkich zmian. Wartość sumaryczna wskazuje na stopień anomalności (mała wartość sumy – mały stopień, duża wartość sumy – duży stopień anomalności).

W drugim przypadku porównuje się aktualną zmianę wartości sygnału wyjściowego (prognozowanego) przez model w stosunku do wartości rzeczywistej (zadanej). Dla uzyskania dobrej generalizacji preferowane są modele stosunkowo proste o ograniczonej liczbie adaptowanych parametrów. Dla osiągnięcia tego celu w procesie uczenia stosuje się różne rozwiązania regularyzacyjne, prowadzące do polepszenia zdolności generalizacji modelu. Typowym przykładem takiego postępowania jest algorytm uczenia sieci SVM, w którym hiperparametr  $C$  reguluje zależność między szerokością marginesu separacji, a aktualnym akceptowalnym błędem klasyfikacji na danych uczących.

W podejściu modelowym ważną rolę odgrywają cechy diagnostyczne procesu stanowiące atrybuty wejściowe dla klasyfikatora. Generuje się je z oryginalnych pomiarów przy zastosowaniu różnego rodzaju technik przetwarzania. Dominują tu metody aproksymacyjne, autoregresyjne, transformacyjne i dekompozycyjne. Do takich zaliczyć można zastosowanie aproksymacji typu wielomianowego [6] w których buduje się model wielomianowy (np. wielomianów Hermita), przy czym wartości współczynników tego modelu stanowią cechy diagnostyczne. W przypadku

autoregresji typowe jest zastosowanie modelu ARIMA, adaptującego się do zmienności procesu [7]. Transformacja Fouriera [8] stanowi reprezentację przebiegu czasowego procesu w dziedzinie częstotliwości, natomiast transformacja falkowa [9, 10] reprezentuje proces w różnej skali czasu i częstotliwości umożliwiając lepsze wychwycenie chwilowych zmian wartości analizowanego sygnału w różnej skali czasowo/częstotliwościowej. Ważną rolę w tworzeniu cech diagnostycznych odgrywa przekształcenie PCA (ang. *principal component analysis* [2]), które poprzez liniową kombinację wagową poszczególnych składników sygnału oryginalnego tworzy zredukowany zestaw atrybutów tworzony w taki sposób, aby uzyskać najlepsze zachowanie oryginalnej informacji o procesie (rzutowanie oryginalnego wielowymiarowego układu współrzędnych w inny układ zorganizowany według wariancji sygnału oryginalnego). Niezależnie od zastosowanej metody tworzenia cech diagnostycznych podstawową zasadą tych metod jest redukcja wymiarowości oryginalnego wektora sygnałów zmierzonych przy jak najlepszym odwzorowaniu informacji pierwotnej o procesie [11, 12].

Po przetworzeniu sygnałów oryginalnych na cechy diagnostyczne stanowią one atrybuty wejściowe dla klasyfikatorów, których zadaniem jest rozpoznać proces anomalny od normalnego. W praktyce stosowane są różne podejścia do budowania takich klasyfikatorów, poczynając od najprostszych KNN, poprzez drzewa decyzyjne, klasyczne sieci neuronowe, sieci SVM aż do struktur głębokich CNN [1, 13].

### **1.3 Cel i przegląd zawartości pracy**

**Celem pracy jest opracowanie metod wykrywania anomalii procesów reprezentowanych przez szereg danych sygnałów, które pozwolą na uzyskanie polepszonej dokładności wychwycenia anomalii. Podstawą proponowanych rozwiązań będzie podejście modelowe, wykorzystujące transformacje falkowe (ciągłą i dyskretną) do generacji cech diagnostycznych i użycie zespołu klasyfikatorów klasycznych lub głębokich jako głównych narzędzi podejmujących ostateczną decyzję przypisania obserwacji do klasy anomalnej lub normalnej. Zostanie pokazane, że opracowany system wykrywania anomalii bardzo dobrze sprawdza się na problemach z różnych dziedzin inżynierii i może mieć zastosowanie do wykrywania zjawisk anomalii procesów charakteryzowanych zarówno poprzez sygnały jednowymiarowe jak i dwuwymiarowe (obrazy).**

Zaproponowane zostanie zastosowanie wielu rozwiązań klasyfikatorów współpracujących ze sobą w zespole dla wypracowania optymalnej decyzji (poprzez głosowanie większościowe).

W wyniku analizy teoretycznej istniejących rozwiązań i wielu prób wstępnych wyselekcjonowano zespół klasycznych (płytkich) klasyfikatorów, w tym las losowy drzew decyzyjnych (ang. *random forest* - RF), Extra las losowy (ERF), klasyfikator typu Gradient Boosting (GB), perceptron wielowarstwowy (MLP), maszyna wektorów nośnych (ang. *support vector machine* – SVM, klasyfikator K najbliższych sąsiadów (ang. *K nearest neighbors* - KNN), naiwny klasyfikator Bayesa (NB) oraz klasyfikator bazujący na procesach gaussowskich (ang. *Gaussian Proces* - GP). Każde z tych rozwiązań ma swoją własną specyfikę i zasadę działania, pozwalającą na zachowanie dużej niezależności w kształtowaniu werdyktu klasyfikacyjnego (podstawowy warunek dobrego działania zespołu). Niezależnie od powyższego systemu klasycznego zaproponowane zostanie również zastosowanie zespołu bazującego na strukturach głębokich CNN, znacznie różniących się sposobem przetwarzania danych, co daje podstawy do niezależności ich działania (podstawowy warunek włączenia ich w zespół).

Dla uzyskania najlepszych wyników rozpoznania anomalii od procesu normalnego ważną rolę odgrywa wstępne przetworzenie oryginalnych szeregów czasowych reprezentujących proces w zestaw atrybutów wejściowych dla zespołu klasyfikatorów. W pracy podstawę takiego preprocesingu danych stanowić będzie transformacja falkowa [9]. Wykorzystywać będziemy zarówno ciągłą transformację (ang. *Continuous Wavelet Transform* – CWT) jak i dyskretną formę (ang. *Discrete Wavelet Transform* – DWT). W przypadku CWT wynik jest w postaci obrazu, który może być bezpośrednio podany na wejście sieci głębokiej CNN realizującej jednocześnie generację cech diagnostycznych i funkcję klasyfikacyjną. Zastosowanie DWT generuje wiele wyjściowych sygnałów (szeregów czasowych) na z góry przyjętych poziomach dekompozycji. Każdy z tych poziomów podlegać będzie następnie opisowi statystycznemu, a wyselekcjonowane parametry opisu stanowić będą atrybuty wejściowe dla zespołu klasycznych klasyfikatorów.

Praca została podzielona na 5 rozdziałów. Poza wstępem w rozdziale drugim, przedstawione zostały metody wykrywania anomalii bazujące na uczeniu maszynowym. W pierwszej jego części omówiono metody generacji cech diagnostycznych zastosowane w pracy. Bazują na dwóch trybach transformacji falkowej: DWT oraz CWT. Transformacja DWT dekomponuje badany szereg czasowy na wiele poziomów, które następnie są reprezentowane poprzez wybrane parametry statystyczne, w tym wartość średnią, odchylenie standardowe, energię, skośność, kurtozę, itp. stanowiące deskryptory procesu. Następnym etapem jest selekcja deskryptorów umożliwiająca wyselekcjonowanie zredukowanej ich liczby stanowiących cechy diagnostyczne procesu

podawane na klasyfikator jako atrybuty wejściowe. W przypadku zastosowania CWT wynik wyjściowy jest w postaci obrazu we współrzędnych przesunięcie-skala/częstotliwość. Obrazy te były podawane bezpośrednio na wejście głębokich sieci neuronowych CNN. W wyniku ich przetworzenia przez warstwy konwolucyjne lokalnie połączone, wytwarzane są automatycznie numeryczne deskryptory obrazu zasilające końcowy klasyfikator w formie softmaxu.

W drugiej części drugiego rozdziału zaprezentowane zostały podstawowe informacje dotyczące rozwiązania klasyfikatorów zastosowanych w pracy. Dotyczy to zarówno klasyfikatorów klasycznych (RF, ERF, GB, MLP, SVM, KNN, GP) jak i różnych rozwiązań sieci neuronowych głębokich (*alexnet*, *densenet*, *resnet*, *inception*, *efficientnet*, *squeezenet*, *googlenet*, *shufflenet*, *darknet*, *mobilenet*). Niezależnie od rodzaju podejścia (klasyczne czy głębokie) zastosowane klasyfikatory połączone były w zespół, którego wyniki są integrowane w jeden finalny werdykt poprzez głosowanie większościowe.

Rozdział trzeci dotyczy wykrywania anomalii w sygnałach EKG. Rozważono 2 rodzaje anomalii: jedna związana z arytmia i druga odpowiadająca zastoinowej niewydolności serca. Dane zaczerpnięto z bazy „*Research Resource for Complex Physiologic Signals PhysioNet*” obejmującej między innymi stosowaną powszechnie bazę *MIT-BIH Arrhythmia database*. Opracowano 2 systemy wykrywania anomalii: jeden oparty na dyskretnej transformacji falkowej i zespole klasyfikatorów płytkich i drugi wykorzystujący ciągłą transformację falkową i zespół klasyfikatorów neuronowych głębokich CNN. Przedstawiono wyniki badań eksperymentalnych potwierdzające wysoką sprawność opracowanych systemów.

Rozdział czwarty został poświęcony wykrywaniu uszkodzenia łożysk tocznych na podstawie zarejestrowanych sygnałów czujników akcelerometrycznych. Rozważono 4 rodzaje uszkodzeń łożyska: defekty bieżni zewnętrznej, defekty bieżni wewnętrznej, defekty elementów tocznych oraz defekty łączone. W rozwiązaniu problemu zastosowano (podobnie jak w przypadku EKG) dwa rodzaje systemów. System pierwszy oparty jest na falkowej dekompozycji dyskretnej współpracującej z zespołem klasyfikatorów płytkich i system drugi wykorzystujący ciągłą transformację falkową zasilającą zespół głębokich sieci neuronowych CNN. Przy zastosowaniu obu opracowanych systemów wykrywania anomalii na ogólnodostępnej bazie danych [14] uzyskano dokładność rozpoznania anomalii zbliżoną do 100% .

W rozdziale piątym opisano inny rodzaj anomalii. Zadanie dotyczyło wykrywania obrazów twarzy typu deep fake, jako fałszywych obrazów utworzonych z oryginałów przy użyciu

nowoczesnych technik sztucznej inteligencji. Opracowano system ekstrakcji obrazu z klatek video z ogólnodostępnej bazy *Forensics++*, tworząc własny zbiór danych do eksperymentów numerycznych. W rozwiązaniu problemu zaproponowano własną technikę przetwarzania obrazów opartą na ciągłej transformacji falkowej i zastosowaniu zespołu głębokich sieci neuronowych CNN odpowiedzialnych za wykrycie podróbek obrazów. Przedstawione wyniki badań numerycznych wskazują na zadowalającą skuteczność opracowanego rozwiązania.

Zwieńczeniem pracy jest podsumowanie wyników rozprawy z podkreśleniem własnych osiągnięć oryginalnych, stanowiących podstawę rozprawy. Wskazano również na przyszłe kierunki badań w tej tematyce.

W pracy zawarto również wykaz literatury związanej z tematyką rozprawy. Zawiera 96 pozycji dotyczących zarówno treści teoretycznych proponowanych rozwiązań jak i części technicznej związanej z problemami praktycznymi, których dotyczyły eksperymenty numeryczne.



## 2 Metody uczenia maszynowego stosowane w pracy

### 2.1 Wprowadzenie do metod uczenia maszynowego

Główny wysiłek badawczy pracy dotyczący wykrywania anomalii procesów ukierunkowany jest na zastosowanie nieliniowych modeli matematycznych procesu, bazujących na uczeniu z nauczycielem. W takim ujęciu występują dwa podstawowe etapy:

- wstępne przetworzenie danych dla generacji i selekcji cech diagnostycznych procesu,
- wybór zestawu klasyfikatorów połączonych w zespół podejmujący ostateczną decyzję przynależności do klasy normalnej bądź anomalii.

Cechy diagnostyczne wygenerowane w pierwszym etapie stanowią atrybuty wejściowe dla zespołu klasyfikatorów. W generacji cech diagnostycznych podstawowym etapem przetwarzania danych pomiarowych będzie transformacja falkowa, zarówno w wersji dyskretnej jak i ciągłej.

### 2.2 Wstępne przetworzenie danych z użyciem transformacji falkowej

Transformacja falkowa jest przekształceniem całkowym (podobnie jak transformacja Fouriera) różniącym się od niej poprzez jądro przekształcenia. W odróżnieniu od transformacji Fouriera stosującej funkcje o nieskończonej długości nośnika, transformacja falkowa wykorzystuje jądra w postaci falek o skończonym nośniku, spełniające określone wymogi, umożliwiające ich zastosowanie w analizie wielorozdzielczej. Funkcje falek tworzą „rodziny” wywodzące się z funkcji matki (tzw. falki macierzystej) poprzez przesunięcie i skalowanie. O ile w transformacji Fouriera jądro reprezentuje określoną częstotliwość to jądro przekształcenia falkowego reprezentuje przedział częstotliwości o szerokości odwrotnie proporcjonalnej do czasu trwania falki.

#### 2.2.1 Ciągła transformacja falkowa

Ciągła transformacja falkowa (CWT – Continuous Wavelet Transformation) jest zdefiniowana dla sygnałów czasu ciągłego. Wynik CWT dla sygnału  $x(t)$  oznaczony jako  $W_x(a,b)$  jest definiowany w postaci [9, 10, 15, 16]:

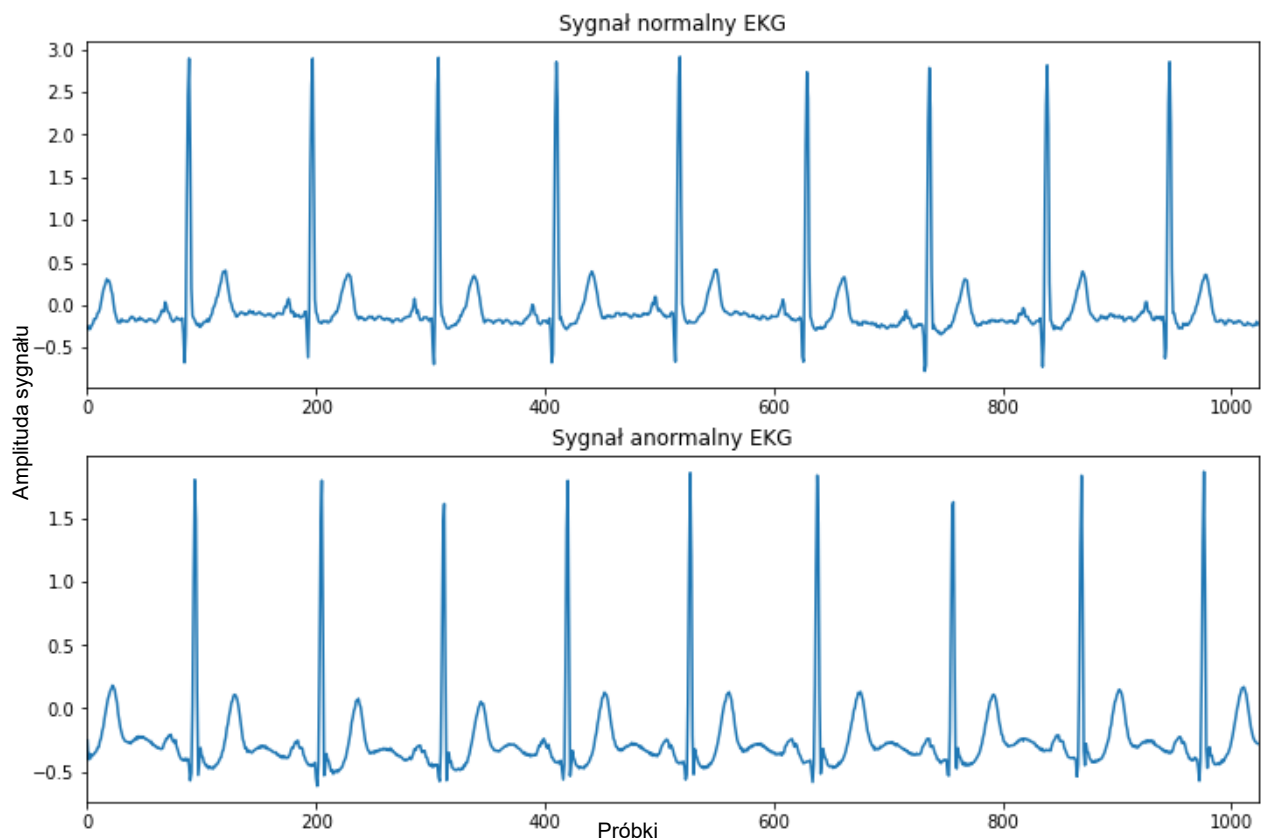
$$W_x(a,b) = |a|^{-\frac{1}{2}} \int_{-\infty}^{\infty} x(t) \tilde{\psi}\left(\frac{t-b}{a}\right) dt \quad (2.1)$$

Gdzie  $\tilde{\psi}$  oznacza falkę użytą w dekompozycji (analizie) sygnału,  $a$  - skalę czasu,  $b$  - wartość przesunięcia falki.

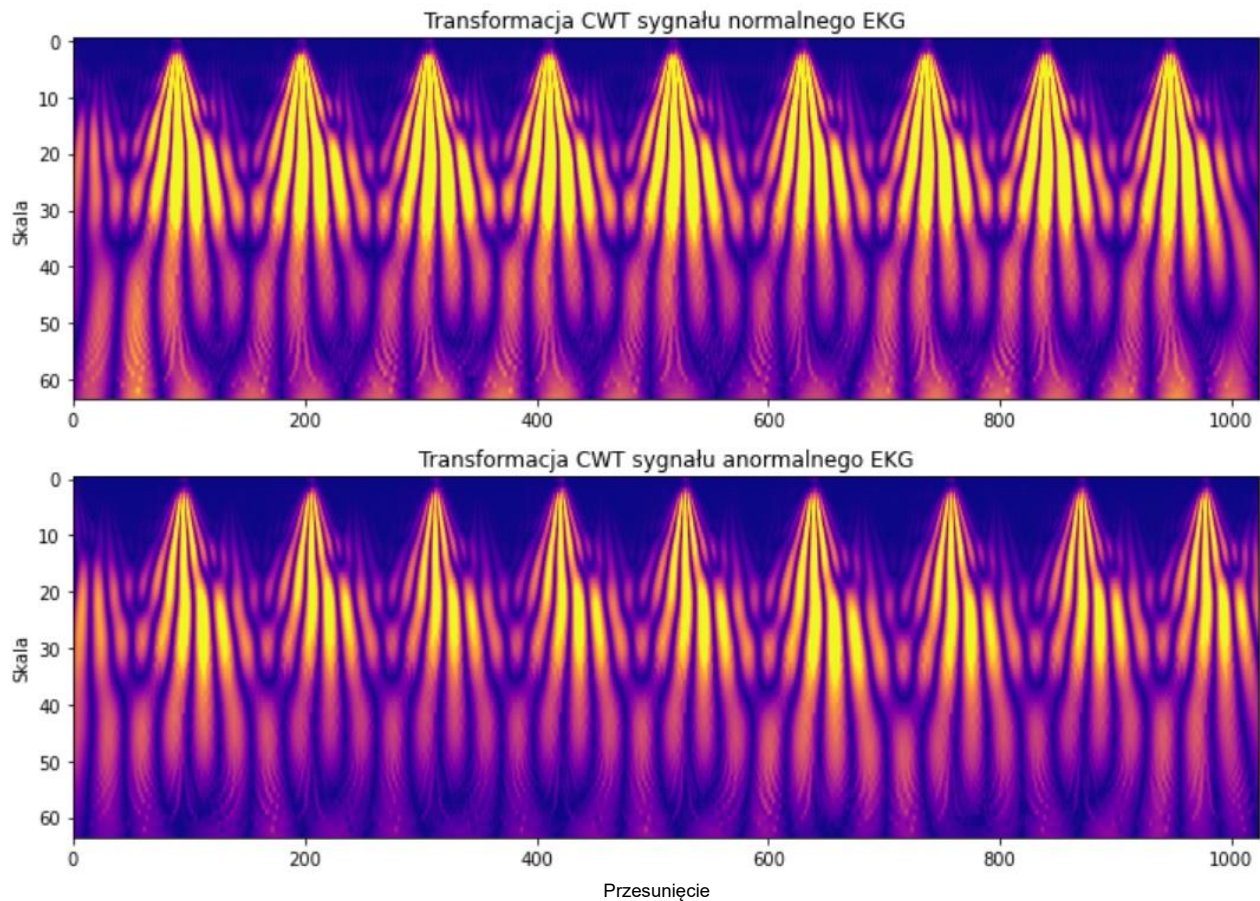
Rozdzielczość czasowa i częstotliwościowa wyniku dekompozycji falkowej jest regulowana poprzez współczynnik skali czasowej  $a$ . Dla małej wartości  $a$  (duża częstotliwość) mamy wysoką rozdzielczość czasową wyniku, ale niską rozdzielczość częstotliwościową. Duże wartości  $a$  generują niską rozdzielczość w czasie, ale wysoką w dziedzinie częstotliwości.

Ciągła transformacja falkowa wykorzystuje falki specjalnie zdefiniowane dla tego rodzaju przekształceń. Przykładami typowych falek stosowanych wyłącznie w transformacji CWT są falka Morleta, kapelusz meksykański („*Mexican hat*”), zespolona falka Morleta czy zespolona falka Gabora [15]. Wszystkie posiadają opis jawny.

Wynik transformacji  $W_x(a,b)$  reprezentowany jest w układzie współrzędnych: skala czasu  $a$  oraz przesunięcie  $b$  (obie wielkości typu ciągłego), co może być odzwierciedlone w postaci obrazu. Przykłady takiego przekształcenia danych jednowymiarowych sygnału EKG (rys. 2.1) w postaci obrazu wynikowego CWT przedstawione są dla sygnału normalnego oraz anomального na rys. 2.2.



Rys. 2.1 Przykładowe przebiegi sygnału EKG poddane analizie CWT.



Rys. 2.2 Wynik transformacji CWT uzyskany dla sygnału EKG z rys. 2.1 przy zastosowaniu falki Morleta.

Obraz uzyskany w wyniku przekształcenia przedstawia zachowanie procesu EKG w skali czasu (oś pozioma reprezentująca przesunięcie) i częstotliwości (oś pionowa przedstawia wielkość współczynnika skali  $a$  – odwrotność częstotliwości). Widoczne są istotne różnice obu obrazów, zwłaszcza dla dużej wartości skali  $a$ .

Obrazy uzyskane w wyniku transformacji CWT stanowią będą dane wejściowe dla zespołu klasyfikatorów CNN. Klasyfikatory te będą odpowiedzialne za wykrycie anomalii. Sieci CNN w swojej strukturze mają wbudowane warstwy konwolucyjne o połączeniu lokalnym odpowiedzialne za automatyczną generację cech diagnostycznych procesu poddanego analizie. Cechy wygenerowane w ten sposób są jednocześnie podawane na końcowy klasyfikator, zwykle typu softmax, którego zadaniem jest rozpoznanie anomalii.

### 2.2.2 Dyskretna transformacja falkowa

Ciągła transformacja falkowa generuje ogromną liczbę szczegółów, niekoniecznie istotnych z technicznego punktu widzenia. Stąd powstała dyskretna forma tej transformacji stosująca dyskretne wartości skali  $a$  i przesunięcia  $b$ , zwykle w postaci diadycznej, w którym wartości parametrów zmieniają się z mnożnikiem dwa. Wprowadzając pojęcie poziomu  $m$ -tego i przesunięcia  $n$ -tego liczonego w liczbie przesunięć okresów  $T$  falki na aktualnym poziomie, współczynniki skali i przesunięcie zapisują się w postaci  $a \rightarrow a_m = 2^m$  oraz  $b \rightarrow b_{m,n} = 2^m \cdot nT$ . Na każdym poziomie  $m$  i przesunięciu  $n$  falka podlega przeskalowaniu [9, 10, 17]:

$$\psi_{a,b}(t) \rightarrow \psi_{m,n}(t) = (a_m)^{-1/2} \cdot \psi\left(\frac{t - b_{m,n}}{a_m}\right) = (2)^{-m/2} \cdot \psi(2^{-m}t - nT) \quad (2.2)$$

Dyskretna transformacja falkowa (DWT) jest wówczas zdefiniowana wzorem:

$$W_x(m, n) = 2^{-m/2} \int_{-\infty}^{\infty} x(t) \tilde{\psi}(2^{-m}(t - 2^m nT)) dt \quad (2.3)$$

Na każdym  $m$ -tym poziomie dekompozycji DWT otrzymuje się szeregi czasowe odpowiednie dla danej skali częstotliwościowej, reprezentujące cechy charakterystyczne procesu w danym zakresie częstotliwości.

Rekonstrukcja sygnału w DWT jest bardzo prosta i ograniczona do sumowania odpowiednich składników rozkładu według wzoru [9]:

$$x(t) = \sum_{-\infty}^{\infty} \sum_{-\infty}^{\infty} W_x(2^m, 2^m nT) \psi_{m,n}(t) \quad (2.4)$$

W rekonstrukcji używa się falki  $\psi(t)$  komplementarnej do  $\tilde{\psi}(t)$  użytej w dekompozycji. W dekompozycji dyskretniej używa się specjalne rodziny falek zdefiniowane przez I. Daubechies [10] w sposób rekurencyjny (niejawny). Zdefiniowanych zostało wiele rodzin falek ortogonalnych i biortogonalnych, w tym falki Daubechies, symlety, coiflety, falki biortogonalne [16].

Zależność dekompozycyjna (2.3) reprezentuje operację splotu, w której sygnał analizowany  $x(t)$  jest splatany z analizującą funkcją falkową  $\tilde{\psi}(t)$ , reprezentując filtrację. To spostrzeżenie dało podstawę do opracowania specjalnej procedury implementującej matematyczny zapis transformacji (2.3) poprzez wielokrotne operacje filtracji realizowanych z użyciem filtrów cyfrowych typu FIR. Podstawę matematyczną takiej ekwiwalentności dały prace I. Daubechies [10] oraz S. Mallata [17]. Implementacja rodziny falek zdefiniowanych przez Daubechies została przez Mallata przetworzona w proces filtracji cyfrowej z użyciem filtrów FIR dolnoprzepustowych i górnoprzepustowych, które pełnią fizycznie rolę dekompozycji (2.3).

W algorytmie Mallata dokonuje się rozkładu analizowanego sygnału na część zawierającą aproksymację niższej rozdzielczości czasowej (operator  $A_jx$ ) oraz szczegóły różnicowe (operator  $D_jx$ ), reprezentujące różnicę sygnału o poprzedniej i aktualnej rozdzielczości [17]:

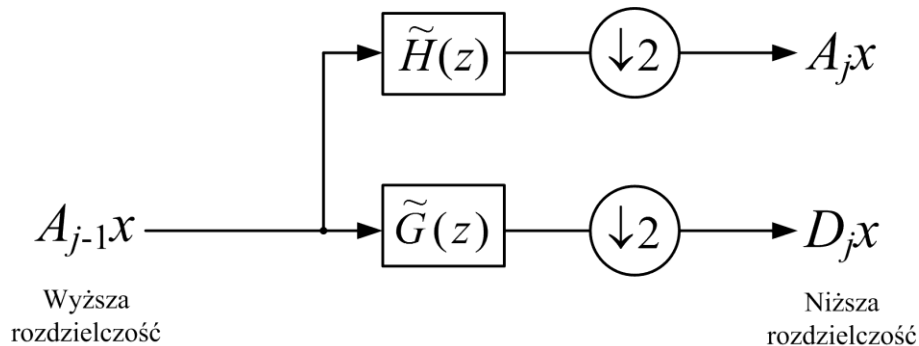
$$x(t) = x_a(t) + x_d(t) \quad (2.5)$$

W równaniu tym sygnał  $x_a(t)$  jest reprezentowany przez operator  $A_jx$ , a  $x_d(t)$  przez operator  $D_jx$ , gdzie  $D_jx = A_{j-1}x - A_jx$ . Sygnał aproksymowany niższej rozdzielczości jest splotem odpowiedzi impulsowej  $\tilde{h}(u)$  filtru dolnoprzepustowego i sygnału aproksymowanego  $A_{j-1}x(u)$  wyższej rozdzielczości (z poprzedniego poziomu dekompozycji), przy zatrzymaniu co drugiego wyniku:

$$A_jx = \tilde{h}(u) * A_{j-1}x(u) \quad (2.6)$$

Analogicznie otrzymuje się sygnał różnicowy poprzez filtrację górnoprzepustową sygnału aproksymowanego wyższej rozdzielczości, co można zapisać w postaci splotu odpowiedzi impulsowej  $\tilde{g}(u)$  filtru górnoprzepustowego i sygnału aproksymowanego  $A_{j-1}x(u)$  (z poprzedniego poziomu dekompozycji), zatrzymując co drugi wynik.

$$D_jx = \tilde{g}(u) * A_{j-1}x(u) \quad (2.7)$$



Rys. 2.3. Schemat blokowy implementacji fizycznej dekompozycji falkowej Mallata na jednym poziomie dekompozycji przy zastosowaniu filtru dolnoprzepustowego  $\tilde{H}(z)$  oraz górnoprzepustowego  $\tilde{G}(z)$ . W wyniku przetworzenia powstaje sygnał aproksymowany niższej rozdzielczości  $A_jx$  oraz sygnał różnicowy  $D_jx$  obu sąsiadujących poziomów rozdzielczości.

Równaniom (2.6) i (2.7) można przyporządkować schemat systemu filtracji, generujący sukcesywnie na kolejnych poziomach dekompozycji sygnał aproksymowany oraz szczegółowy o coraz mniejszej rozdzielczości, jak to pokazano na rys. 2.3 [18]. Na schemacie tym  $\tilde{H}(z)$  reprezentuje transmitancję filtru dolnoprzepustowego o odpowiedzi impulsowej  $\tilde{h}(n)$ , a  $\tilde{G}(z)$

transmitancję filtru górnoprzepustowego o odpowiedzi impulsowej  $\tilde{g}(n)$ . Symbol  $\downarrow 2$  oznacza operację decymacji poprzez zachowanie co drugiej próbki wynikowej (niższa rozdzielczość wyniku w stosunku do sygnału wejściowego).

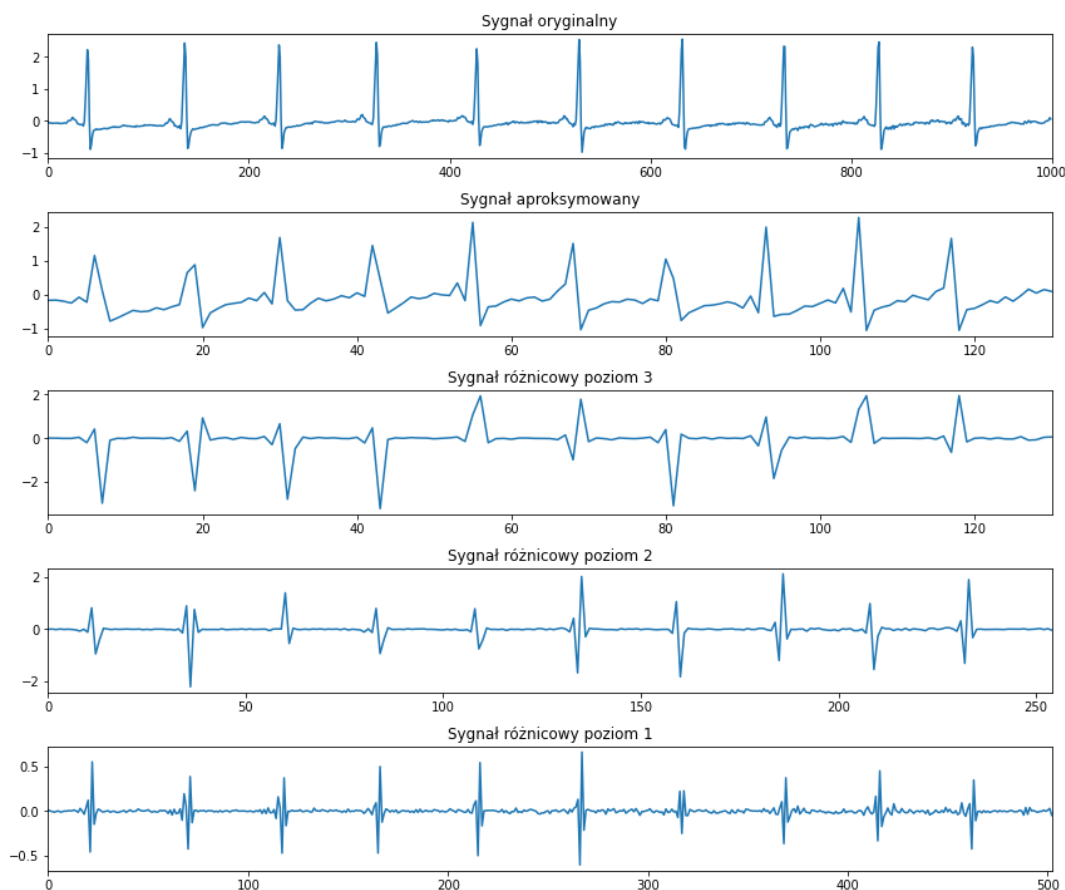
W efekcie zastosowania transformacji DWT i przeskalowania czasowego wyników poszczególnych poziomów do tej samej skali czasu, sygnał oryginalny  $x(t)$  może być przedstawiony jako suma sygnału aproksymacyjnego na końcowym ( $N$ -tym) poziomie dekompozycji oraz sygnałów różnicowych  $D_j x$  na kolejno zadeklarowanych poziomach  $j=1, 2, \dots, N$  [18]:

$$x = A_N x + D_N x + D_{N-1} x + \dots + D_2 x + D_1 x \quad (2.8)$$

Przykład takiej dekompozycji falkowej przy zastosowaniu falki Daubechies na 3 poziomach dekompozycji ( $N=3$ ) dla sygnału EKG przedstawiony jest na rys. 2.4.

W efekcie dekompozycji otrzymuje się zwiększoną ilość informacji uwypuklających zachowanie się sygnału w różnych zakresach częstotliwości (skali). Dzięki temu nasza wiedza o analizowanym procesie zostaje zwiększona. W następnym etapie jest ona przetwarzana na określoną liczbę deskryptorów numerycznych, przy czym każdy poziom dekompozycji reprezentowany jest przez przypisany mu zbiór parametrów uwzględniający różne aspekty częstotliwościowe analizowanego sygnału oryginalnego.

W pracy deskryptory numeryczne będą tworzone poprzez ich opis statystyczny (wartości średnie, odchylenia standardowe, energię, itp.). Analiza tak powstałego zbioru deskryptorów pod kątem ich związku z klasą, przeprowadzona jest przy użyciu wybranej metody selekcji cech. W jej wyniku zbiór zdefiniowanych wcześniej deskryptorów przetworzony jest w zestaw cech diagnostycznych procesu, traktowanych jako atrybuty wejściowe dla zespołu klasyfikatorów odpowiedzialnych za wykrycie anomalii.



Rys. 2.4 Wynik dekompozycji DWT na trzech poziomach dla sygnału EKG. Sygnał górny reprezentuje przebieg oryginalny, sygnały poniżej, kolejno: sygnał aproksymowany na trzecim poziomie oraz sygnały różnicowe na kolejnych poziomach, od trzeciego do pierwszego.

## 2.3 Modele klasyfikatorów zastosowane w pracy

### 2.3.1 Losowy las drzew decyzyjnych

Drzewo decyzyjne stanowi model drzewiasty podejmowania decyzji. W procesie tworzenia tego modelu następuje porównanie wartości określonego atrybutu wejściowego każdej obserwacji z przyjętym progiem. Wynik porównania w postaci 2 podzbiorów jest przekazywany do odpowiedniego węzła niższego poziomu. Proces taki prowadzony jest w ogólności aż do chwili uzyskania jednorodności klasowej podzbioru. Przetwarzanie danych wejściowych odbywa się krok po kroku poprzez analizę wartości atrybutów wejściowych i porównywaniu ich z odpowiednio dobranymi progami [19]. W procesie uczenia równolegle przeprowadzany jest wybór odpowiedniego atrybutu wejściowego w każdym węzle podziałowym i określenie wartości progu, na podstawie którego dokonuje się podział zbioru. W efekcie takiej analizy przyjmuje się wybór

tego atrybutu wejściowego (zmienniej  $x_k$ ) i skojarzonego z nim progu, który gwarantuje najmniejszą wartość zanieczyszczenia klasowego obu podzbiorów (miara Giniego lub miara entropijna), na które dzielony jest aktualny zbiór. Powtórzenie wielokrotne podziału zbiorów na podzbiory, coraz mniej różnorodne pod względem zawartości różnych klas prowadzi do przypisania ostatecznej decyzji klasyfikacyjnej dotyczącej obserwacji wejściowej (przypisania wektora wejściowego  $\mathbf{x}$  do odpowiedniej klasy, reprezentowanej w drzewie przez tak zwany liść).

Wadą pojedynczego drzewa jest duża wrażliwość na wartości atrybutów wejściowych, stąd taki klasyfikator uważa się za słaby (ang. *weak classifier*). Dodatkowo pojedyncze drzewo ma inne wady, w tym dużą czułość na zmiany wartości danych wejściowych, co jest istotnym czynnikiem przy ocenie zdolności generalizacji modelu (dane testujące lub weryfikujące z definicji różnią się od danych uczących). Dochodzi do tego brak współdziałania (synergii) wielu zmiennych na raz przy podejmowaniu decyzji podziałowej zbioru w poszczególnych węzłach, co prowadzi zwykle do znacznej rozbudowy struktury drzewa w przypadku problemów o diagonalnym rozkładzie klas wśród danych pomiarowych. Tym nie mniej metoda drzew decyzyjnych jest szczególnie przydatna w problemach decyzyjnych z licznymi, rozgałęziającymi się wariantami oraz w przypadku podejmowania decyzji w warunkach ryzyka oraz przy braku pewnych danych w bazie.

Stąd powstała koncepcja stworzenia zespołu wielu drzew decyzyjnych współpracujących ze sobą przy podejmowaniu ostatecznej decyzji klasyfikacyjnej. Powstało wiele takich rozwiązań, z których najczęściej używany jest las losowy drzew decyzyjnych Breimana, tak zwany *Random Forest* (RF) [20]. Ustalenie końcowego wyniku klasyfikacji odbywa się tu na podstawie głosowania większościowego. Użytkownik wpływa na przebieg procesu tworzenia modelu poprzez ustalenie wstępne liczby losowo dobieranych zmiennych w węźle drzewa oraz liczby drzew wchodzących w skład zespołu [21].

Zakładając, że liczba obserwacji  $N$ -wymiarowych jest równa  $p$ , w algorytmie uczenia lasu losowego przyjmuje, że na każdym poziomie decyzji korzysta się jedynie z  $m < N$  atrybutów, przy czym proponowana jest wartość  $m$  równa w przybliżeniu pierwiastkowi kwadratowemu z  $N$ . Dobór atrybutów w każdym węźle przeprowadzany jest losowo. Zbiór uczący dla każdego drzewa tworzy się zwykle z około  $2/3p$  obserwacji wybieranych z pełnego zbioru poprzez losowanie z zastępowaniem. Pozostała część służy jako zbiór weryfikacyjny (dla każdego drzewa jest on inny ze względu na losowość). Wynik końcowy klasyfikacji na danych testowych jest ustalany przez głosowanie większościowe wszystkich drzew.



Powstało wiele modyfikacji lasu losowego. Jedną z nich wykorzystywaną w tej pracy jest extra RF (ERF) [22]. Główna różnica polega na wyborze wartości progów w węzłach podziałowych. Klasyczny RF dobiera wartości progów dla każdego zbioru  $m$  zmiennych z zastosowaniem procedury optymalizacyjnej względem uzysku, podczas gdy ERF stosuje wartości losowe progów, dzięki czemu uzyskuje się znaczne przyspieszenie procesu. Tym nie mniej ostateczna decyzja podejmowana jest w obu metodach poprzez wybór najlepszej kombinacji. W modyfikacji ERF wybór losowy obserwacji uczących z pełnego zbioru odbywa się bez zwracania (raz wylosowana obserwacja nie trafia z powrotem do puli zbioru podlegającego losowaniu), podczas gdy w klasycznym rozwiązaniu RF ta sama obserwacja ma szansę być wylosowana wielokrotnie.

### 2.3.2 Zespół klasyfikatorów tworzony z zastosowaniem „gradient boosting”

Inną metodą tworzenia systemu klasyfikacyjnego jest zastosowanie specjalnego algorytmu wzbogacania gradientem (tak zwany *gradient boosting* - GB). Zespół jest tworzony z wielu słabych klasyfikatorów (np. drzew decyzyjnych). Dodawane są sukcesywnie kolejne korekty do istniejącego zestawu modelowego. W efekcie w każdym  $n$ -tym etapie korekty model  $F(\mathbf{x})$  klasyfikatora jest wzbogacany o poprawkę  $h_n(\mathbf{x})$ , gdzie  $\mathbf{x}$  jest wektorem zmiennych wejściowych klasyfikatora [31]:

$$F_{n+1}(\mathbf{x}) = F_n(\mathbf{x}) + \eta_n h_n(\mathbf{x}) \quad (2.9)$$

przy czym poprawka  $h_n(\mathbf{x})$  jest zdefiniowana jako różnica między wartością zadaną  $d$  i aktualną odpowiedzią modelu na  $n$ -tym etapie:

$$h_n(\mathbf{x}) = d - F_n(\mathbf{x}) \quad (2.10)$$

Na podstawie tej różnicy generowana jest funkcja strat  $L(d, F_n(\mathbf{x}))$  (na przykład w postaci kwadratowej), dla której wyznacza się gradient  $\nabla L$  tej funkcji względem odpowiedzi modelu. Skorygowana odpowiedź modelu w kroku  $n$ -tym zgodnie z algorytmem największego spadku będzie opisana w postaci:

$$F_{n+1}(\mathbf{x}) = F_n(\mathbf{x}) - \eta \sum_{i=1}^p \nabla_{F_n} L(d_i, F_n(\mathbf{x}_i)) \quad (2.11)$$

Optymalną wartość kroku  $\eta$  w kierunku największego spadku można otrzymać rozwiązując problem optymalizacji jednowymiarowej względem tego kroku, poszukując rozwiązania zadania minimalizacji:

$$\eta = \operatorname{argmin} \sum_{i=1}^p (L(d_i, F_n(\mathbf{x}_i)) + \eta h_n(\mathbf{x}_i)) \quad (2.12)$$

Liczba kroków korekcyjnych  $n$  jest dobierana przez użytkownika dla osiągnięcia pożądanego poziomu błędu uczenia.

### 2.3.3 Sieć SVM

Maszyna wektorów nośnych (SVM) opracowana przez V. Vapnika [23] należy aktualnie do najlepszych klasyfikatorów silnych o bardzo dobrych zdolnościach generalizacji. Stanowi strukturę neuropodobną stosującą różne rodzaje funkcji aktywacji i implementującą specjalny sposób uczenia sprowadzający się do programowania kwadratowego. SVM z definicji ma jedną warstwę ukrytą i jeden neuron wyjściowy umożliwiający rozpoznanie dwu klas. W przypadku wielu klas należy zbudować zespół wielu klasyfikatorów dwuklasowych, ustalając wynik końcowy klasyfikacji poprzez głosowanie, zwykle większościowe.

Najważniejszą cechą SVM, wpływającą na zdolność generalizacji klasyfikatora, jest zastosowanie specjalnego sposobu uczenia polegającego na maksymalizacji szerokości marginesu separacji między dwoma przeciwstawnymi klasami przy kontrolowanym błędzie uczenia na danych uczących. Funkcję „regulatora” pełni współczynnik regularyzacji  $C$ , dobierany przez użytkownika, stanowiący hiperparametr w procesie uczenia. W efekcie uczenie sprowadza się do minimalizacji wartości wag odpowiedzialnych za szerokość marginesu separacji, z regularyzacją uwzględniającą błędy na danych uczących.

Sieć SVM pozwala na znaczną swobodę przy wyborze rodzaju nieliniowości neuronów ukrytych. Można stosować wiele różnych funkcji jądra spełniających warunki twierdzenia Mercera [23, 24], choć najczęściej używane jest jądro gaussowskie  $K(\mathbf{x}_i, \mathbf{x}) = \exp(-\gamma \|\mathbf{x} - \mathbf{x}_i\|^2)$ . Przy zastosowaniu tego typu jądra użytkownik z góry musi ustalić wartości dwu hiperparametrów: współczynnika regularyzacji  $C$  oraz stałej  $\gamma$  jądra gaussowskiego. Odbywa się to zwykle na wstępnie zdefiniowanym zbiorze wartości  $C$  i  $\gamma$  [23]. Ten etap realizowany jest w fazie wstępnej eksperymentów uczenia z użyciem wydzielonego zbioru walidacyjnego. W jego efekcie wybiera się te wartości hiperparametrów, które gwarantują najlepsze wyniki walidacji.

Istotną zaletą nieliniowej sieci SVM jest zastąpienie w procesie uczenia i testowania funkcji wektorowej  $\varphi(\mathbf{x})$  poprzez funkcję skalarną jądra  $K(\mathbf{x}, \mathbf{x}_j)$ , co znakomicie przyspiesza etap uczenia w zadaniu dualnym. W procesie uczenia początkowa liczba wektorów nośnych jest zwykle równa liczbie danych uczących. W zależności od przyjętych wartości ograniczeń regulowanych przez wartość parametru  $C$ , złożoność sieci jest redukowana i tylko część wektorów uczących pozostaje nadal wektorami nośnymi. Przy małej wartości  $C$  kara za przekroczenie ograniczeń jest niewielka i optymalne wartości wag dobrane przez algorytm umożliwiają klasyfikację z wieloma błędami na danych uczących. Aktualna (zwykle suboptymalna) wartość  $C$  jest ustalana przez użytkownika na

drodze prób. Typowe wartości tego parametru dla przypadku danych znormalizowanych zawierają się w przedziale 100-1000.

Sieci SVM, ze względu na zastosowanie jednego neuronu wyjściowego, dokonują podziału danych na dwie klasy. Rozpoznanie wielu klas przy pomocy tej techniki wymaga przeprowadzenia wielokrotnej klasyfikacji. Do najbardziej znanych podejść należą tu metody: "jeden przeciw pozostałym" i "jeden przeciw jednemu". W metodzie "jeden przeciw jednemu" konstruuje się  $M(M-1)/2$  klasyfikatorów typu SVM, rozróżniających za każdym razem 2 klasy danych ze zbioru uczącego, kolejno parowanych ze sobą. W metodzie "jeden przeciw pozostałym" stosuje się  $M$  dwuklasowych sieci SVM. Każda z nich jest odpowiedzialna za rozpoznanie kolejnej klasy przeciwstawianej pozostałym klasom traktowanym łącznie jako pojedyncza klasa. Sieć  $i$ -ta jest trenowana na danych uczących, w których przykłady  $j$ -tej klasy są skojarzone z klasą  $d_j=1$  a pozostałe z klasą  $d_j=-1$ . Po wytrenowaniu wszystkich sieci następuje etap odtwarzania, w którym ten sam wektor  $\mathbf{x}$  jest podawany na każdą sieć dwuklasową SVM, przypisującą wektor wejściowy do określonej klasy. Ostatecznie przypisuje się wektor  $\mathbf{x}$  do klasy o największej liczbie zwycięstw.

#### 2.3.4 Klasyfikator KNN

Klasyfikator KNN (K najbliższych sąsiadów) stanowi najprostsze, choć zaskakująco skuteczne rozwiązanie problemu klasyfikacyjnego. Nie wymaga trybu uczenia, choć dane uczące są podstawą jego działania. Obiekt testowy podlegający klasyfikacji jest przydzielany do klasy, do której należy większość z jego  $K$  najbliższych sąsiadów [26]. Funkcję dyskryminacyjną klasyfikatora KNN która przypisuje wektor testowy  $\mathbf{x}_t$  podany na jego wejście do klasy  $j$  można przedstawić w postaci:

$$f_j(\mathbf{x}_t) = k_j / K \quad (2.13)$$

w której  $k_j$  oznacza liczbę wektorów ze zbioru uczącego reprezentującą  $j$ -tą klasę, które znajdują się wśród  $K$  najbliższych sąsiadów aktualnego wektora  $\mathbf{x}_t$ . Klasyfikator KNN przypisuje analizowany wektor  $\mathbf{x}_t$  do klasy o największej wartości funkcji dyskryminującej, czyli takiej, która jest najliczniejsza wśród  $K$  najbliższych sąsiadów należących do zbioru uczącego. Przy małej wartości  $K$  algorytm jest wrażliwy na szumy. Jego odporność wzrasta po zwiększeniu wartości  $K$ , ale jest to okupione większą złożonością obliczeniową i wydłużonym czasem podjęcia decyzji. W efekcie ustalenie właściwej wartości  $K$  następuje najczęściej metodą prób i błędów na zbiorze danych walidacyjnych, na przykład metodą  $n$ -krotnej walidacji krzyżowej. Typowe jest przyjęcie że  $K$  jest proporcjonalne do pierwiastka z liczby danych uczących.

### 2.3.5 Klasyfikator bazujący na procesie gaussowskim

Klasyfikator oparty na procesie gaussowskim (*Gaussian Process Classifier* – GPC) bazuje na uogólnieniu pojęć rozkładu gaussowskiego definiowanego dla zmiennej (lub zespołu zmiennych) na funkcje tych zmiennych. Jest rozwiązaniem typu stochastycznego generującym prawdopodobieństwo przynależności obserwacji do określonej klasy.

Wykorzystuje, podobnie jak SVM, pojęcie jądra reprezentującego funkcję kowariancji między dwoma zmiennymi. Funkcja ta może być definiowana w różny sposób. Typowe postaci funkcji jądra stosowane w rozwiązaniu obejmują między innymi: jądro gaussowskie RBF (identycznie jak w SVM), jądro Matтерна, jądro kwadratowe o postaci wymiernej, jądro wykładniczo-sinusoidalne, czy nawet jądro w postaci iloczynu skalarnego [24, 25].

W procesie gaussowskim najważniejszą rolę odgrywają funkcje kowariancji, pokazujące relacje występujące między poszczególnymi obserwacjami wielowymiarowymi. Wartość funkcji jądra  $K(\mathbf{x}_i, \mathbf{x}_j)$  jest miarą podobieństwa wektorów, świadcząca jak blisko te wektory są położone względem siebie, zakładając przy tym, że wektory bliskie sobie reprezentują identyczne klasy. Element decyzyjny przynależności klasowej wykorzystuje funkcję logistyczną (sigmoidalną)  $f(u) = 1/(1 + \exp(-u))$  przetwarzającą argument wejściowy  $u$  na wartość prawdopodobieństwa z przedziału  $[0, 1]$  na binarną przynależność klasową. Uzyskana wartość prawdopodobieństwa przetwarzana jest na przynależność klasową ostrą poprzez porównanie z progiem równym 0.5. W przypadku zadania wieloklasowego stosuje się system wielu klasyfikatorów binarnych działających w trybie jeden przeciw jednemu lub jeden przeciw reszcie (jak w sieci SVM).

Szczegóły matematyczne działania klasyfikatora GPC są złożone i nie będą podlegały tutaj dogłębnej analizie. Można je prześledzić w pracach teoretycznych poświęconych procesowi gaussowskiemu, między innymi w [13, 25].

### 2.3.6 Sieć neuronowa MLP

Sieć perceptronu wielowarstwowego zwana w skrócie MLP (ang. *MultiLayer Perceptron*) stanowi ważne rozwiązanie neuronowego systemu klasyfikacyjnego zdolne do rozpoznania wielu klas jednocześnie [3]. Cecha charakterystyczną jest zastosowanie nieliniowej funkcji aktywacji w postaci sigmoidy. W praktyce sieć jest tworzona przez warstwę sygnałową oraz jedną (wyjątkowo dwie) warstwę ukrytą oraz jedną warstwę wyjściową o liczbie neuronów sigmoidalnych odpowiadających liczbie klas. Sieć jest trenowana na zestawie par danych uczących  $(\mathbf{x}, \mathbf{d})$ , gdzie  $\mathbf{x}$  jest wektorem reprezentującym atrybuty wejściowe, a  $\mathbf{d}$  wektorem wartości zadanych na wyjściu.

Celem uczenia jest taki dobór wag połączeń synaptycznych między kolejnymi warstwami neuronów, który zapewni minimum funkcji błędu (kosztu) na danych uczących, definiowany zwykle z wykorzystaniem metryki euklidesowej.

Jakkolwiek cel uczenia jest implementowany poprzez minimalizację funkcji błędu głównym zadaniem procesu jest taki dobór struktury sieci i jej wag, aby sieć „nauczyła” się pośrednio mechanizmu odwzorowania danych wejściowych  $\mathbf{x}$  w dane wyjściowe  $\mathbf{y}$  (na etapie uczenia wektory  $\mathbf{y}$  są reprezentowane przez wektory zadane  $\mathbf{d}$ ). W teorii stosowane są różne specjalizowane algorytmy gradientowe (metoda największego spadku z momentem rozpędowym, metody pseudo-newtonowskie BFGS lub Levenberga-Marquardt, metoda gradientów sprzężonych), choć w praktyce przy dużych wymiarach sieci i bogatej bazie uczącej stosuje się najczęściej algorytmy stochastyczne największego spadku (ang. *stochastic gradient descent* – SGD) z momentem rozpędowym [27].

### 2.3.7 Naiwny klasyfikator Bayesa

Klasyfikator Bayesa należy do rodziny klasyfikatorów probabilistycznych. Zadaniem tego klasyfikatora Bayesa jest określenie prawdopodobieństwa przynależności danych wejściowych do klasy przy zastosowaniu tak zwanej reguły Bayesa [13, 28]. W pełnym klasyfikatorze Bayesa określana jest dokładna wartość prawdopodobieństwa, natomiast w klasyfikatorze naiwnym wartość proporcjonalna do tego prawdopodobieństwa.

Przy istnieniu  $M$  klas i wystąpieniu jednoczesnym  $N$  atrybutów wejściowych  $X_1, X_2, \dots, X_N$  reprezentujących „prawdę” (wartość 1) prawdopodobieństwo wystąpienia  $i$ -tej klasy  $d_i$  przy  $i = 1, 2, \dots, M$ , określa wzór Bayesa [28]:

$$P(d_i / X_1, X_2, \dots, X_N) = \frac{P(X_1, X_2, \dots, X_N / d_i)P(d_i)}{\sum_{k=1}^M P(X_1, X_2, \dots, X_N / d_k)P(d_k)} \quad (2.14)$$

Zakładając niezależność atrybutów wejściowych wzór powyższy upraszcza się do postaci:

$$P(D_i / X_1, X_2, \dots, X_N) = \frac{P(X_1 / D_i)P(X_2 / D_i) \cdots P(X_N / D_i)P(D_i)}{\sum_{k=1}^M P(X_1 / D_k)P(X_2 / D_k) \cdots P(X_N / D_k)P(D_k)} \quad (2.15)$$

Po wyznaczeniu prawdopodobieństwa wystąpienia każdej klasy, za zwycięską uznaje się tę o największej wartości prawdopodobieństwa. Jest to pełna reguła Bayesa minimalizująca statystyczne ryzyko pomyłki. Biorąc pod uwagę, że przy wyznaczaniu prawdopodobieństwa wystąpienia każdej klasy mianownik wyrażenia jest taki sam, regułą pełną Bayesa można zastąpić

wersją uproszczoną (tak zwany naiwny klasyfikator Bayesa), w której decyzje podejmuje się na podstawie jedynie licznika, zastępując prawdziwą wartość prawdopodobieństwa poprzez współczynnik  $\alpha(d_i/X_1, X_2, \dots, X_N)$  proporcjonalny do tej wartości. Takie założenie upraszcza wzór (2.14) o prawdopodobieństwie warunkowym do postaci:

$$\alpha(d_i/X_1, X_2, \dots, X_N) = P(d_i) \prod_{j=1}^N P(X_j/d_i) \quad (2.16)$$

W efekcie zbiór atrybutów wejściowych  $X_i$  jest przyporządkowany klasie, dla której współczynnik  $\alpha$  przyjął największą wartość.

### 2.3.8 Klasyfikatory głębokie CNN

Konwolucyjne sieci neuronowe (ang. *convolutional neural networks*), zwane również sieciami splotowymi, stworzone zostały pierwotnie do analizy danych 2-wymiarowych (obrazów), choć aktualnie stosowane są również do danych wielowymiarowych, w tym 1-wymiarowych (sygnałów) [21].

Sieci te łączą w swojej strukturze dwie funkcje: automatyczną generację cech diagnostycznych (bez manualnego udziału człowieka) oraz końcową klasyfikację. Poszczególne warstwy sieci CNN przetwarzają obrazy z warstwy poprzedzającej (na wstępie jest to zbiór obrazów oryginalnych) poszukując prymitywnych cech (np. grupy pikseli o podobnym stopniu szarości, krawędzie, przecinające się linie itp.). Kolejne warstwy ukryte generują dalsze uogólnienia cech z warstwy poprzedzającej organizowane w formie obrazów, które w końcowym etapie stanowią atrybuty wejściowe dla klasyfikatora wbudowanego w strukturę sieci [4, 30].

Cechą wspólną tych rozwiązań jest wielowarstwowość ułożenia neuronów (zwana głębokością) i równoległe działanie neuronów w warstwie związane z tworzeniem wielu obrazów równoległe (tak zwana szerokość warstwy). Dla uniknięcia problemu lawinowego narastania liczby adaptowanych wag stosuje się połączenia typu lokalnego. W tego typu rozwiązaniach neuron jest zasilany nie przez wszystkie sygnały (na przykład piksele obrazów) warstwy poprzedzającej, jak to jest zorganizowane w sieciach klasycznych, ale przez wybraną małą grupę neuronów (pikseli) tej warstwy, tworzących maskę filtrującą. Analiza całego obrazu następuje poprzez przesuwanie tej maski z ustalonym krokiem (ang. *stride*) wzdłuż i w szerz obrazu (analiza typu lokalnego) [31].

Typowa warstwa składa się z kilku podwarstw, zawierających obrazy tworzone przy pomocy operacji splotu liniowego, następnie nieliniowego przetworzenia jej wyników z użyciem funkcji ReLU (lub jej licznych modyfikacji), redukcji wymiarów obrazów wynikowych poprzez operacje

„pooling” oraz normalizacji poprzez zastosowanie standaryzacji dla podzbioru aktualnych obrazów tworzących tzw. „mini batch” [32].

W wyniku przetworzenia danych przez wiele warstw sieciowych tensor wyjściowy obrazów przetransformowany jest w postać wektorową. Elementy tego wektora są traktowane jako cechy diagnostyczne stanowiące atrybuty wejściowe dla końcowego stopnia struktury stanowiącego klasyfikator. Jest to zwykle prosty klasyfikator typu *softmax*, w którym sygnały wejściowe są bezpośrednio przetwarzane na prawdopodobieństwo przynależności do określonej klasy. Liczba neuronów wyjściowych jest równa liczbie klas, przy czym każdy neuron podlega adaptacji wag w klasyczny sposób poprzez optymalizację funkcji celu, definiowanej zwykle w postaci funkcji entropii krzyżowej (ang. *cross-entropy*). Wartość sygnału sumacyjnego  $i$ -tego neuronu wyjściowego określona jest wówczas wzorem:

$$u_i(\mathbf{x}) = \sum_j w_{ij}x_j + w_0 \quad (2.17)$$

Prawdopodobieństwo przynależności wektora  $\mathbf{x}$  do  $i$ -tej klasy ( $i=1, 2, \dots, M$ ) zależy od wartości funkcji *softmax* obliczanej dla  $i$ -tej klasy według wzoru [21]:

$$\text{softmax}_i = f(u_i) = \frac{\exp(u_i)}{\sum_{j=1}^M \exp(u_j)} \quad (2.18)$$

Jej wartość największa wyznacza przynależność obserwacji  $\mathbf{x}$  do klasy określonej wskaźnikiem  $i$ .

Funkcja kosztu podlegająca minimalizacji jest definiowana zwykle w postaci entropii krzyżowej (*cross-entropy*), łączącej w sobie aktualną, wskazaną przez sieć przynależność  $f(u_i)$  z wartością prawdziwą  $d_i$ . W zdaniach klasyfikacji wieloklasowej ( $M$  klas) tylko jedna klasa jest prawdziwa (etykieta tej klasy  $d=1$ ). Pozostałe klasy mają etykietę  $d=0$ . W procesie uczenia tylko klasa wskazana przez sieć jako prawdziwa bierze udział w obliczaniu funkcji błędu (kosztu) przyjmując  $d_i=1$ . Pozostałe wyjścia mają etykiety  $d=0$  i nie biorą udziału w procesie propagacji wstecznej. Entropijna definicja funkcji celu zapisana jest wówczas jako [21]:

$$E = -\sum_{i=1}^M d_i \log(f(u)_i) \quad (2.19)$$

która wobec tylko jednej wartości  $d_i=d_p$  różnej od zera (równiej 1) jest uproszczona do postaci:

$$E = -\log(f(u)_i) = -\log\left(\frac{\exp(u_p)}{\sum_{j=1}^M \exp(u_j)}\right) \quad (2.20)$$

Składniki funkcji celu względem klas „negatywnych” są równe zero. Tylko klasa wskazana przez sieć bierze udział w tworzeniu funkcji celu. Tym nie mniej gradient funkcji celu zależy również od składników klas „negatywnych”, ze względu na postać mianownika w wyrażeniu.

Ze względu na ogromną liczbę adaptowanych parametrów sieci stosowany jest prosty stochastyczny algorytm największego spadku z momentem rozpędowym (ang. *Stochastic Gradient Descent* – SGD), w którym adaptacja wag odbywa się iteracyjnie zgodnie ze wzorem [27]:

$$\mathbf{w}(k) = \mathbf{w}(k-1) - \eta \mathbf{g}(k-1) + \alpha [\mathbf{w}(k-1) - \mathbf{w}(k-2)] \quad (2.21)$$

gdzie  $\mathbf{g}$  jest wektorem gradientu a  $\alpha$  jest współczynnikiem momentu dobieranym przez użytkownika w przedziale  $[0, 1]$ . Proces adaptacji parametrów w poszczególnych iteracjach używa losowo wybranego podzbioru danych (ang. *mini batch*).

W ostatnich latach powstał ulepszony algorytm uczący zwany ADAM (*ADaptive Momenet estimation*) [33]. W tej metodzie każdy parametr  $w$  ma indywidualny, adaptacyjnie dobierany współczynnik uczenia uzależniony od momentu statystycznego gradientu pierwszego i drugiego rzędu definiowanych w postaci średniej kroczącej:

- moment pierwszego rzędu

$$m_w(k) = \beta_1 m_w(k-1) + (1 - \beta_1) g(k) \quad (2.22)$$

- moment drugiego rzędu

$$v_w(k) = \beta_2 v_w(k-1) + (1 - \beta_2) g^2(k) \quad (2.23)$$

gdzie  $\beta_1$  i  $\beta_2$  są hiperparametrami dobieranymi przez użytkownika [30]. Z tych wartości określa się estymaty nieobciążone:

$$\hat{m}_w(k) = \frac{m_w(k)}{1 - \beta_1^k} \quad (2.24)$$

$$\hat{v}_w(k) = \frac{v_w(k)}{1 - \beta_2^k} \quad (2.25)$$

Z udziałem których dokonuje się adaptacji wagi w  $k$ -tej iteracji według wzoru:

$$w(k) = w(k-1) - \eta \frac{\hat{m}_w(k)}{\sqrt{\hat{v}_w(k) + \varepsilon}} + \alpha [w(k-1) - w(k-2)] \quad (2.26)$$

Często ostatni człon tego wzoru związany z momentem rozpędowym jest pomijany.

Aktualnie istnieje ogromna liczba różniących się struktur i implementacji komputerowych sieci CNN dostępnych w Internecie [34, 35, 36, 37, 38, 39, 40]. Są to struktury wstępnie wytrenowane na zbiorze *ImageNet*, które mogą być w trybie *Transfer Learning* wykorzystane przez innych



użytkowników dla rozwiązania różnorodnych problemów. Pierwszą taką siecią dostępną w Internecie była sieć ALEXNET, o strukturze 25-warstwowej przedstawionej poniżej [18, 34].

```
1 'data' Image Input 227x227x3 images with 'zerocenter' normalization
2 'conv1' Convolution 96 11x11x3 convolutions with stride [4 4] and padding [0 0]
3 'relu1' ReLU
4 'norm1' Cross Channel Normalization with 5 channels per element
5 'pool1' Max Pooling 3x3 max pooling with stride [2 2] and padding [0 0]
6 'conv2' Convolution 256 5x5x48 convolutions with stride [1 1] and padding [2 2]
7 'relu2' ReLU
8 'norm2' Cross Channel Normalization with 5 channels per element
9 'pool2' Max Pooling 3x3 max pooling with stride [2 2] and padding [0 0]
10 'conv3' Convolution 384 3x3x256 convolutions with stride [1 1] and padding [0 0]
11 'relu3' ReLU
12 'conv4' Convolution 384 3x3x192 convolutions with stride [1 1] and padding [0 0]
13 'relu4' ReLU
14 'conv5' Convolution 256 3x3x192 convolutions with stride [1 1] and padding [1 1]
15 'relu5' ReLU
16 'pool5' Max Pooling 3x3 max pooling with stride [2 2] and padding [0 0]
17 'fc6' Fully Connected 4096 fully connected layer
18 'relu6' ReLU
19 'drop6' Dropout 50% dropout
20 'fc7' Fully Connected 4096 fully connected layer
21 'relu7' ReLU
22 'drop7' Dropout 50% dropout
23 'fc8' Fully Connected 1000 fully connected output layer
24 'prob' Softmax softmax
25 'output' Classification 1000 classes
```

Sieć ta w wersji pre-trenowanej umożliwia rozpoznanie do 1000 klas. Ta liczba klas stała się normą również dla innych implementacji CNN. Tabela 2.1 przedstawia wybrany zestaw aktualnie dostępnych w Matlabie sieci CNN w połączeniu z podstawowymi danymi charakteryzującymi sieć z punktu widzenia użytkownika: głębokość (liczba warstw), *size* (zajętość pamięci w komputerze), *parameters* (liczba parametrów adaptowanych podana w milionach) oraz *Image Input Size* (wymagany wymiar obrazów wejściowych) [34, 36, 37, 38, 39, 40].

Tabela 2.1 Przykładowe aktualnie dostępne w Matlabie pre-trenowane struktury sieci CNN oraz ich podstawowe parametry.

Network	Depth	Size	Parameters (Millions)	Image Input Size
squeezenet	18	5.2 MB	1.24	227-by-227
googlenet	22	27 MB	7.0	224-by-224
inceptionv3	48	89 MB	23.9	299-by-299
densenet201	201	77 MB	20.0	224-by-224
mobilenetv2	53	13 MB	3.5	224-by-224
resnet18	18	44 MB	11.7	224-by-224
resnet50	50	96 MB	25.6	224-by-224
resnet101	101	167 MB	44.6	224-by-224
xception	71	85 MB	22.9	299-by-299
inceptionresnetv2	164	209 MB	55.9	299-by-299
shufflenet	50	5.4 MB	1.4	224-by-224
nasnetmobile	*	20 MB	5.3	224-by-224
nasnetlarge	*	332 MB	88.9	331-by-331
darknet19	19	78 MB	20.8	256-by-256
darknet53	53	155 MB	41.6	256-by-256
efficientnetb0	82	20 MB	5.3	224-by-224
alexnet	8	227 MB	61.0	227-by-227
vgg16	16	515 MB	138	224-by-224
vgg19	19	535 MB	144	224-by-224

Pomimo ogromnej różnicy w liczbie adaptowanych parametrów poszczególne rozwiązania charakteryzują się zbliżoną do siebie skutecznością przetwarzania obrazów. W rozwiązaniu stosowanym w pracy zastosowany został zespół klasyfikatorów CNN złożony z wielu wymienionych wyżej sieci. Ich skład został ustalony w wyniku wielu eksperymentów wstępnych:

- alexnet,
- mobilenetv2,
- resnet50,
- efficientnet,
- squeezenet,
- googlenet,
- shufflenet,
- inceptionresnetv2.

Omówione w tym rozdziale metody przetwarzania danych, oparte na transformacji falkowej i zastosowaniu szeregu różnych rozwiązań klasyfikatorów tworzących zespół podejmujący

ostateczną decyzję przynależności klasowej danych wejściowych, zostaną zastosowane do wykrywania określonych anomalii w trzech różnych rodzajach procesów. Anomalie procesów będą rozpoznawane na podstawie analizy sygnałów charakterystycznych dla analizowanych procesów. Dwa z nich dotyczą szeregów czasowych. Są to sygnały EKG oraz sygnały sensoryczne łożysk tocznych. Trzeci rozważany przypadek związany jest z wykryciem podróbek 3. obrazów typu „*deep fake*”. Niezależnie od rodzaju sygnałów w pracy zaproponowano zastosowanie podobnych procedur przetwarzania danych. Dane numeryczne wykorzystane w tych eksperymentach zostały zaczerpnięte z ogólnodostępnych baz danych.

## 3 Wykrywanie anomalii w sygnałach EKG

### 3.1 Wstęp do przetwarzania sygnałów EKG

Elektrokardiogram (EKG/ECG) reprezentuje elektryczną aktywność serca, odwzorowaną przy pomocy elektrod umieszczonych w określonych punktach ciała pacjenta. Dane uzyskiwane są z elektrod pomiędzy którymi występują zmiany potencjału elektrycznego wraz z aktywnością skurczową i rozkurczową mięśnia sercowego.

Zmiany we wzorcu sygnału EKG mogą być spowodowane wieloma różnymi zaburzeniami kardiologicznymi, takimi jak zaburzenia rytmu pracy serca, zwężenie przepływu krwi w tętnicach, zaburzenia w poziomie elektrolitów, itp. Komputerowo wspomaganą analizę sygnału EKG jest pomocna w detekcji chorób układu krążenia.

Problem wykrywania różnego rodzaju anomalii w sygnałach EKG ma bardzo długą historię, ze względu na praktyczne aspekty tego problemu. W przeszłości stosowane były różnorodne metody przetwarzania sygnałów EKG ukierunkowane na wykrywanie arytmii czy innych schorzeń serca. W obszarze tworzenia cech diagnostycznych stosowane są między innymi metody autoregresyjne, transformacja Fouriera (w szczególności STFT), transformacje falkowe, metody aproksymacyjne, zastosowanie cech statystycznych bazujących na różnego rodzaju dekompozycjach, filtracja morfologiczna [41]. Zwykle w końcowym rozpoznaniu anomalii stosowane są różnego rodzaju rozwiązania klasyfikacyjne, w tym sieci neuronowe MLP, RBF, SVM czy ostatnio popularne rozwiązania sieci neuronowych głębokich. Przegląd tych metod można znaleźć między innymi w publikacji [42]. Wyniki badań prezentowane w licznych publikacjach dotyczą różnego rodzaju anomalii, a badania wykorzystują zwykle ogólnodostępną bazę danych *MIT-BIH arrhythmia database* (MIT-BIH AD).

W pracy [43] rozważano 3 rodzaje anomalii (LBBB, RBBB, P) przy użyciu jako klasyfikatorów MLP i SVM. Na bazie 48 rekordów EKG wziętych z MIT-BIH AD uzyskano błąd względny wykrycia zespołu QRS poniżej 0.42% i dokładność wykrycia anomalii równą 96.67% dla klasyfikatora MLP oraz 98.39% dla SVM.

Praca [44] prezentuje rozwiązanie dotyczące wykrycia 4 rodzajów anomalii w sygnałach EKG. Cechy diagnostyczne w postaci kilku wielkości statystycznych są definiowane na podstawie tak zwanych funkcji trybu wewnętrznego. Zasilają one klasyfikator SMO-SVM (ang. *sequential minimal optimization* - SVM). Deklarowane wyniki dla bazy MIT\_BIH AD to czułość średnia 98.01% , specyficzność 99.49% i dokładność 99.20%.

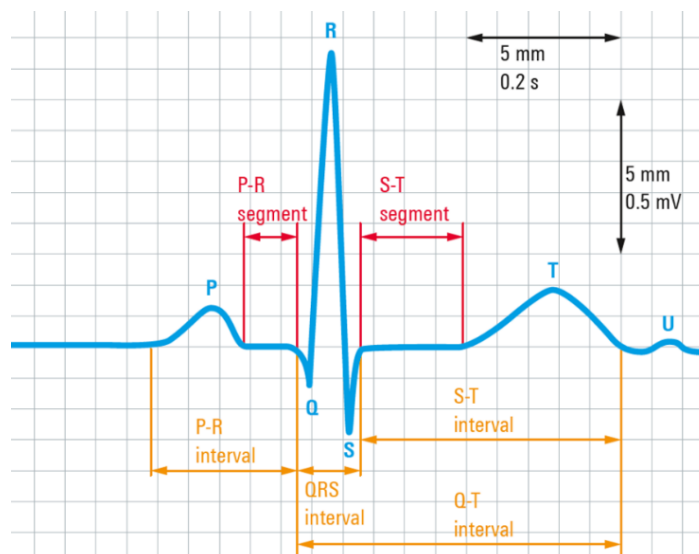
W pracy [42] zaproponowano metodę wykrywania anomalii sygnałów EKG przy wykorzystaniu spektrogramu wynikającego z krótkookresowej transformacji Fouriera (STFT) połączonego z manualnie generowanymi deskryptorami sygnału. Tak wytworzone cechy diagnostyczne wykorzystują sieć głęboką CNN jako końcowy element rozwiązania w wykrywaniu 16 typów anomalii (głównie różnych rodzajów arytmii). Deklarowana dokładność to 99.79% i czułość 99.74% w wykrywaniu anomalii.

Obecnie, najczęściej używana jest transformata falkowa której wyniki stanowią źródło opisu statystycznego sygnału stanowiące sygnały wejściowe dla końcowego systemu klasyfikacyjnego najczęściej w postaci sieci neuronowych, w tym sieci głębokich. Przetworzony rezultat transformaty jest podawany na wejście systemów klasyfikacyjnych odpowiedzialnych za wykrywanie anomalii w przebiegu elektrokardiogramu. Jakość wyników uzyskanych w procesie rozpoznania anomalii jest mocno uzależniona od bazy danych użytej w eksperymentach. Publikowana dokładność i wrażliwość w różnych opracowaniach ogólnodostępnych waha się pomiędzy 95% a 100% w zależności od użytych metod, rodzajów anomalii i bazy danych użytej w eksperymentach [43].

W sygnale EKG wyróżniamy 5 podstawowych załamków (rys. 3.1), czyli wychyleń od linii izoelektrycznej (linia poziomu sygnału podczas braku aktywności mięśnia sercowego) [45]:

- P – reprezentujący depolaryzację mięśnia przedsionków serca,
- Q, R, S – reprezentujący depolaryzację mięśnia komór serca, najbardziej istotne z punktu widzenia analizy sygnału EKG,
- T – reprezentujący repolaryzację mięśnia komór serca.

Zaburzenia rytmu serca reprezentują różne typy anomalii w sygnale. Sygnał w pełni zdrowej osoby jest bardzo regularny, po depolaryzacji przedsionków zawsze następuje depolaryzacja komór. W przypadku arytmii rytm serca staje się nieregularny, częstotliwość jest wyższa lub niższa niż dla zdrowej osoby.



Rys. 3.1 Ilustracja typowego przebiegu sygnału EKG z podziałem na segmenty [46].

Sygnał EKG posiada wiele nagłych zmian przebiegu. Analizując jedynie widmo takiego sygnału, szumy i przejścia normalne przebiegu nie są separowalne przy klasycznych metodach filtracji. Otwiera to szerokie możliwości zastosowania transformaty falkowej w opisie EKG [47].

### 3.2 Baza danych zastosowana w eksperymentach

W badaniach użyto bazy danych pochodzącej z „*Research Resource for Complex Physiologic Signals PhysioNet*” [48]. Baza zawiera 162 nagrania sygnału EKG pochodzące z trzech grup badawczych, dwie grupy przejawiały oznaki nieprawidłowej pracy serca, trzecia pracę normalną:

- Grupa I – sygnały charakteryzujące arytmie, 96 zbiorów sygnałów,
- Grupa II – sygnały charakteryzujące niewydolność serca, 30 zbiorów sygnałów.
- Grupa III – sygnały normalnej pracy serca, 36 zbiorów sygnałów.

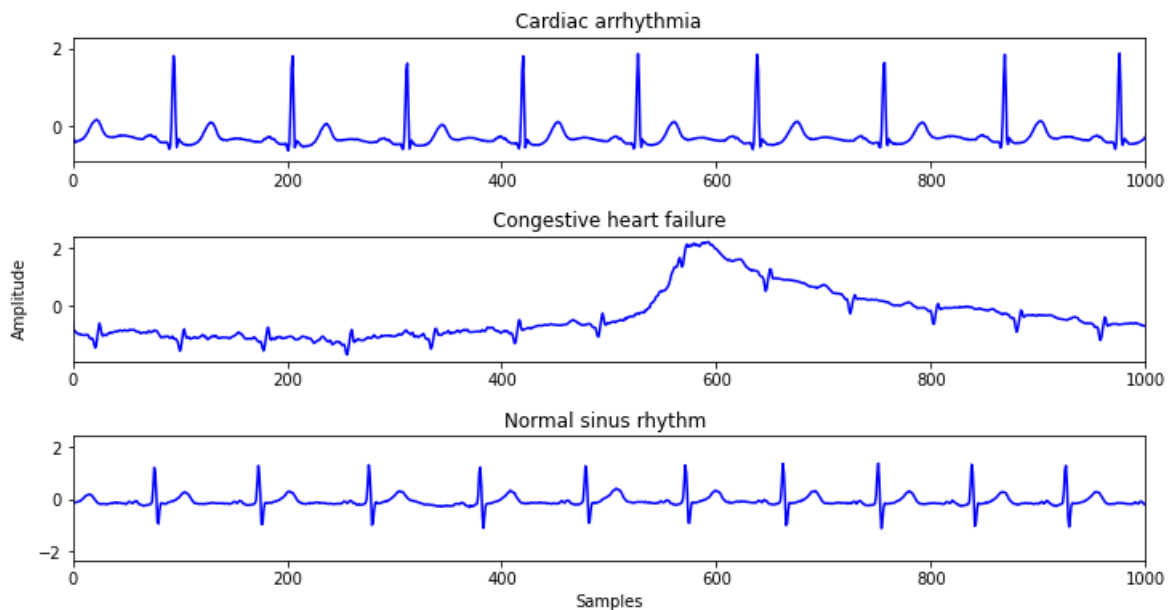
przy czym każdy zbiór składa się z 65536 próbek.

Badana baza składała się z trzech ogólnodostępnych baz:

- „*MIT-BIH Arrhythmia Database*” (arytmia – ARR) – baza składająca się z nagrań sygnałów 60% pacjentów hospitalizowanych i 40% pacjentów ambulatoryjnych w Bostońskim szpitalu „Beth Israel”.
- „*MIT-BIH Normal Sinus Rhythm Database*” (rytm normalny – NSR) – w skład badanej grupy wchodziło 5 mężczyzn w wieku od 26 do 45 lat oraz 13 kobiet w wieku 20 do 50 lat, przy czym żadna badana osoba nie przejawiała objawów nieprawidłowej pracy serca.

- „*The BIDMC Congestive Heart Failure Database*” – (zastoinowa niewydolność serca - CHG) – w skład badanej grupy wchodziło 11 mężczyzn w wieku 22 do 71 lat i 4 kobiety w wieku od 54 do 63 lat, badani mieli stwierdzone liczne zaburzenia niewydolności pracy serca.

Baza została stworzona w Bostońskim szpitalu „Beth Israel” z wykorzystaniem elektrokardiografów ambulatoryjnych. Użyte dane były próbkowane z częstotliwością 128 Hz, a następnie skrócone do długości 65536 próbek. Przykładowe przebiegi sygnałów z tych trzech grup są przedstawione na rys. 3.2.



Rys. 3.2 Przykłady rytmów EKG reprezentujących grupy I, II i III.

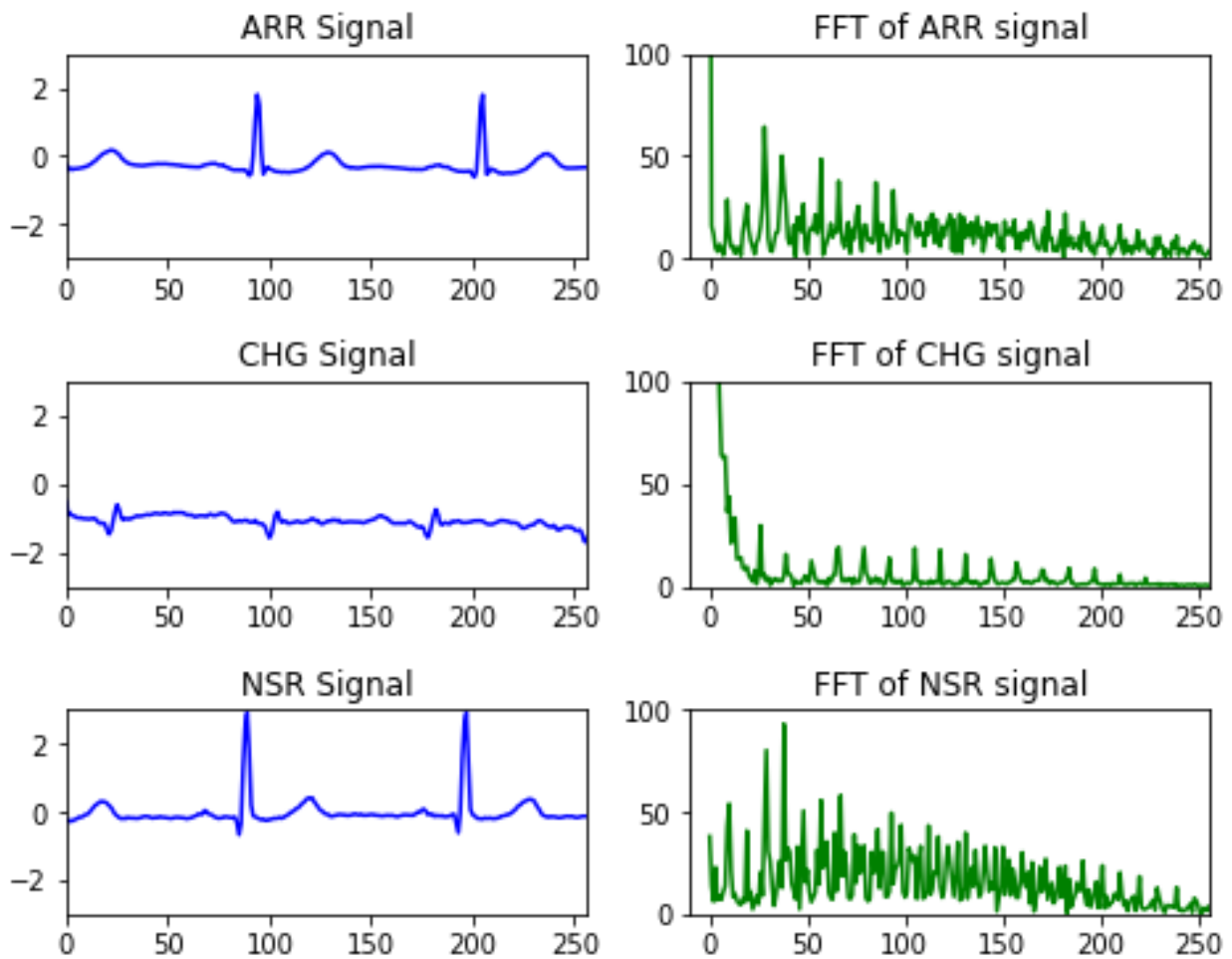
W eksperymentach numerycznych przeprowadzonych w rozprawie 70% dostępnych danych stanowi zbiór uczący, pozostałe 30% dane testujące. Obie grupy były rozłączne, wybrane losowo bez powtórzeń. W sumie dało to 10368 odcinków przebiegu o długości 1024 próbek.

### 3.3 Generacja cech diagnostycznych sygnału EKG z wykorzystaniem parametrów statystycznych

#### 3.3.1 Wstępne przetwarzanie sygnału EKG

Sygnał EKG ze względu na sposób rejestracji zawiera szумы, powodowane czynnikami takimi jak zakłócenia linii zasilającej elektrokardiograf, poruszenie się elektrod wraz z ruchami badanego pacjenta lub tłumienie samych elektrod. Zmiany impedancji elektrod często skutkują fluktuacjami całkowitego poziomu sygnału. Rys. 3.3 ilustruje przykładowo trzy rodzaje badanych sygnałów

(ARR, CHG oraz NSR) i ich widmo częstotliwościowe [8, 18]. Rozkład energii sygnału w poszczególnych pasmach częstotliwości jest zdecydowanie różny dla każdego typu rytmu.



Rys. 3.3 Przykładowe przebiegi sygnałów EKG reprezentujących 3 rozważane grupy oraz ich widma amplitudowe.

Częstotliwość szumów w sygnale EKG spowodowanych fluktuacjami całkowitego poziomu sygnału mieszczą się w zakresie 0.3-1.5 Hz [49]. Po filtracji szumów, powinna nastąpić analiza charakterystyk sygnału. Jedną z metod jest szybka transformacja Fouriera (FFT). Wynik transformacji pozwala określić punkty PQRST sygnału i przy założeniu odpowiedniego przesunięcia, odseparować składowe sygnału uderzeń serca. Jest to istotne, gdyż sygnał został podzielony na odcinki po 10 składowych uderzeń serca, które są następnie przetwarzane w systemie wykrywającym anomalie. Niedopasowanie okna w odcinku skutkowałoby znaczącymi niespójnościami pomiędzy odcinkami sygnału tego samego typu.

Detekcja zespołu QRS wykorzystana w badaniach opiera się o algorytm znany pod nazwą „Pan-Tomkins QRS detection algorithm” [45], jak to przedstawiono na rys. 3.4.





Rys. 3.4 Schemat detekcji zespołu QRS w sygnałach EKG według algorytmu Pana-Tompkinsa.

Algorytm ten zakłada trzystopniowy filtr cyfrowy realizowany programowo. Pierwszy z nich to filtr pasmowo-przepustowy o współczynnikach całkowitych, składających się z kaskadowo połączonych filtrów dolno- i górno-przepustowych. Filtr ten redukuje szumy pochodzące z ruchów mięśni, fluktuacji całkowitej poziomu sygnału i wpływu załamka T. Pasma przepustowe filtra które maksymalizuje energię załamka QRS wynosi od 5 Hz do 15 Hz. Transmitancja filtrów dolnoprzepustowego (indeks dolny d) i górnoprzepustowego (indeks dolny g) określona jest wzorem [45]:

$$H_d(z) = \frac{(1 - z^{-6})^2}{(1 - z^{-1})^2} \quad (3.1)$$

$$H_g(z) = \frac{(1 - 32z^{-16} + z^{-32})}{(1 - z^{-1})} \quad (3.2)$$

Kolejny filtr różniczuje sygnał wejściowy dla uzyskania informacji o zboczach załamka QRS. Transmitancja filtru różniczkującego w tym algorytmie jest określona wzorem [45]:

$$H(z) = \frac{1}{8} T(-z^{-2} - 2z^{-1} + 2z^1 + z^2) \quad (3.3)$$

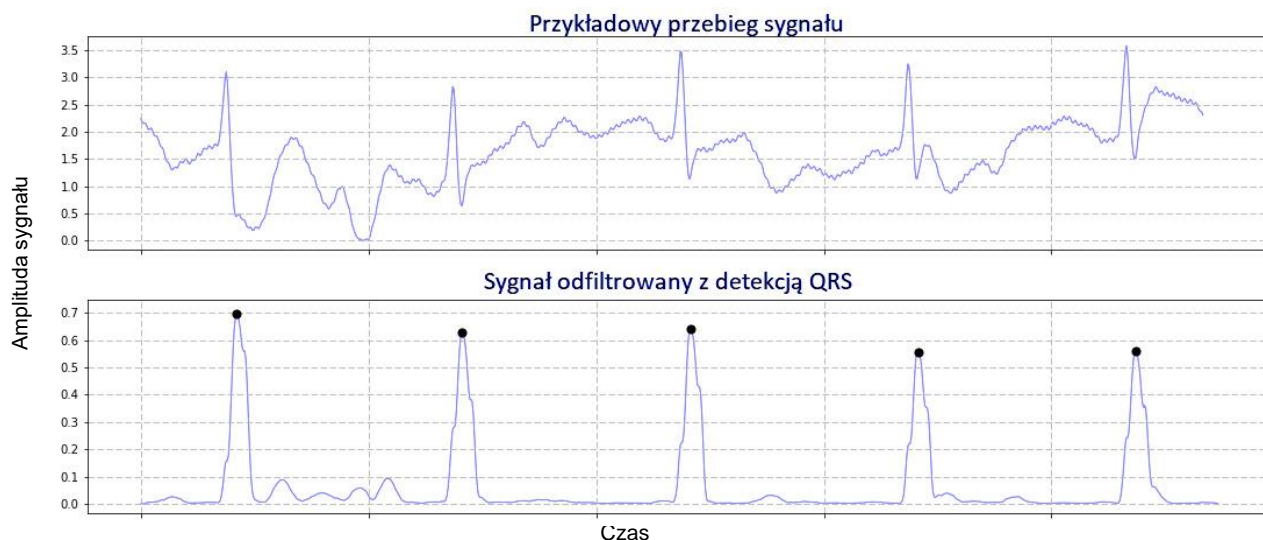
Każda wartość sygnału odfiltrowanego jest podnoszona do kwadratu, aby uzyskać tylko wartości dodatnie i wprowadzić nieliniowe wzmocnienie sygnału. W ostatnim kroku, sygnał jest

całkowany przy użyciu filtru z ruchomym oknem. Jego celem jest uzyskanie informacji o przebiegu funkcji. Sygnał wyjściowy filtru jest generowany według wzoru:

$$y(nT) = [x(nT) - (N - 1)T + x(nT - (N - 2)T) + \dots + x(nT)] \quad (3.4)$$

gdzie  $n$  jest liczbą próbek w oknie filtru całkującego. Wynik całkowitej filtracji sygnału jest klasyfikowany jako załamki QRS lub odrzucany jako szum.

Na rys. 3.5 przedstawiono wynik działania algorytmu wykrywającego zespoły QRS (rysunek dolny) w sygnale EKG (rysunek górny). W pracy wykorzystano algorytm zaimplementowany w języku Python [50]



Rys. 3.6 Ilustracja działania algorytmu wykrywającego zespoły QRS (rysunek dolny) w sygnale EKG (rysunek górny).

### 3.3.2 Zastosowanie transformacji DWT do generacji cech diagnostycznych sygnału EKG

Dyskretna dekompozycja falkowa stanowi dyskretną transformację całkową sygnału w dziedzinie czasu i częstotliwości dokonującą dekompozycji sygnału na wielu poziomach rozdzielczości czasowej/częstotliwościowej [10, 18]. W wyniku jej zastosowania sygnał oryginalny jest reprezentowany na każdym poziomie poprzez sygnały szczegółowe reprezentujące różne zakresy częstotliwościowe oraz sygnał aproksymacyjny na ostatnim poziomie. Poszczególne operacje generujące kolejne sygnały dekompozycyjne wykorzystują dwa rodzaje filtrów FIR. Filtr dolnoprzepustowy odpowiada za generację sygnału aproksymacyjnego na danym poziomie dekompozycji, a filtr górnoprzepustowy odpowiada za generację sygnału szczegółowego (różnicowego). W rozwiązaniu zadania dokonano dekompozycji na 5 poziomach, generując w efekcie 6 zbiorów sygnałowych (5 sygnałów szczegółowych i jeden aproksymacyjny na 5-tym poziomie).

Celem zastosowania transformacji DWT w analizie sygnału EKG jest dekompozycja sygnału na wiele pasm częstotliwości, odpowiadających różnym poziomom dekompozycji, z których każdy reprezentuje różne rozkłady częstotliwościowe. Wyniki transformacji na każdym poziomie mogą zostać scharakteryzowane poprzez zestaw parametrów statystycznych. W wyniku wstępnie przeprowadzonych badań wybrano następujące opisy statystyczne:

- wartość średnia,
- wartość średniokwadratowa,
- mediana,
- odchylenie standardowe,
- kurtoza,
- skośność,
- percentyle 5%, 25%, 75%, 95%,
- ilość punktów przekroczenia poziomu zerowego,
- ilość punktów przekroczenia wartości średniej,
- entropia.

Łączna liczba deskryptorów wyniosła 78 dla wszystkich zbiorów wynikowych (5 sygnałów szczegółowych i jeden aproksymacyjny, każdy charakteryzowany poprzez przyjęte wartości statystyczne).

### **3.3.3 Selekcja cech diagnostycznych procesu**

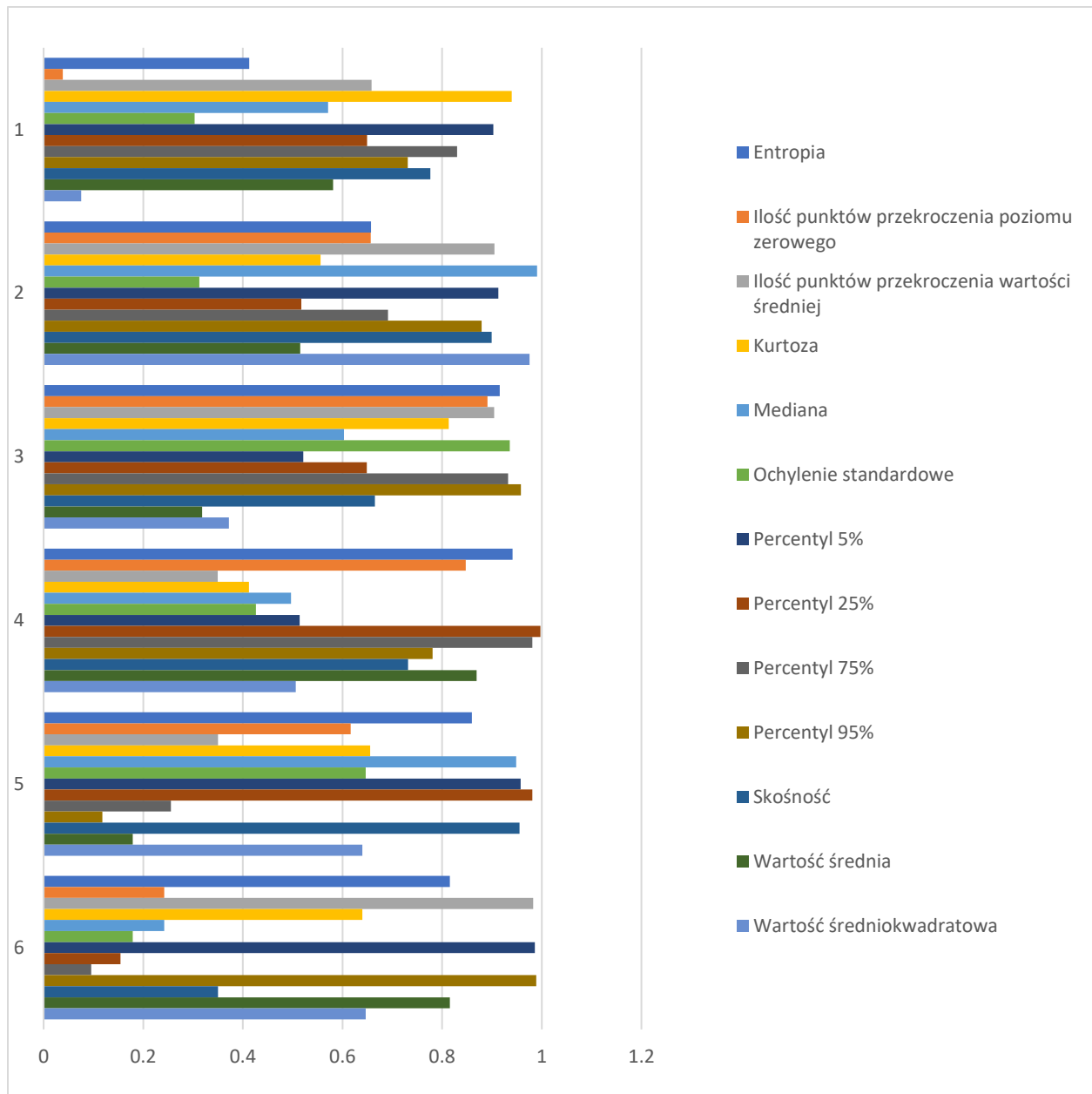
W wyniku automatycznego procesu przetwarzania danych omówionych w punkcie poprzednim utworzony jest zestaw deskryptorów numerycznych, które stanowią potencjalne cechy diagnostyczne. Analiza związku tych parametrów z klasą przypisaną analizowanemu procesowi pokazuje, że nie wszystkie deskryptory numeryczne są równie ważne w procesie rozpoznania. Stąd stosuje się następny etap przetwarzania, polegający na selekcji najważniejszych deskryptorów, które stanowią będą ostateczne cechy diagnostyczne procesu, stanowiące atrybuty wejściowe dla systemu klasyfikacyjnego.

Możliwe jest zastosowanie różnych metod selekcji deskryptorów, które pozwolą dobrze reprezentować modelowany proces przy podejmowaniu decyzji przynależności klasowej [51]. Liczba cech diagnostycznych nie może być przy tym zbyt duża, gdyż ich nadmiarowa populacja pogarsza zdolność generalizacji modelu. Wybór wyselekcjonowanego zbioru cech powinien

zapewniać dobrą korelację cechy z klasą. Spośród wielu istniejących metod selekcji wybrano metodę  $\chi^2$ , ze względu na jej prostotę i jednocześnie dobre wyniki uzyskane we wstępnych eksperymentach porównawczych różnych metod.

Zbiór 78 deskryptorów numerycznych wygenerowanych przy zastosowaniu transformacji DWT poddano analizie przy pomocy testu zgodności  $\chi^2$  [52]. Jest to nieparametryczny test statystyczny sprawdzający hipotezę zerową, że analizowana klasa przynależności danych wejściowych jest niezależna od danej cechy (taka cecha podlega eliminacji ze zbioru potencjalnych cech). Hipotezą alternatywną jest zależność odpowiedzi systemu od danej cechy. Przy niezależności obu zmiennych wartość testu  $\chi^2$  jest mała, co oznacza niezależność wyboru klasy od danej cechy. Duża wartość testu świadczy, że hipoteza zerowa o niezależności jest niewłaściwa, czyli w rzeczywistości cecha ma wpływ na decyzję o przynależności klasowej obserwowanych danych. Im większa jest wartość testu  $\chi^2$  dla analizowanej cechy tym zależność zmiennej wyjściowej (klasy) od danej cechy jest większa. Wynik testu pomiędzy deskrytorem numerycznym (potencjalną cechą diagnostyczną), a klasą przynależności danych wejściowych pozwala ustawić kolejność cech według przypisanej im wartości  $\chi^2$ . W związku z powyższym do cech diagnostycznych procesu wybiera się zbiór  $n$  deskryptorów o wartościach największych testu, eliminując pozostałe, przy czym ich liczebność  $n$  jest dobierana eksperymentalnie. Tak wyselekcjonowany zbiór cech podawany jest na wejście klasyfikatorów w przypadku zastosowania modeli płytkich w rozwiązaniu.

Na rys. 3.7 przedstawiono wykres przedstawiający zbiór estymowanych wartości wynikowych testu  $\chi^2$  dla 78 deskryptorów numerycznych w problemie wykrycia anomalii sygnałów EKG. Przedstawia względne wartości tego testu dla 13 deskryptorów i wyników dekompozycji DWT na pięciu poziomach. Zestaw 1 dotyczy sygnału aproksymacyjnego na piątym poziomie, natomiast zestawy od 2 do 6 dotyczą sygnałów szczegółowych na poszczególnych poziomach, poczynając od piątego a kończąc na pierwszym.



Rys. 3.7 Wykres przedstawiający zbiór estymowanych wartości testu  $\chi^2$  dla 78 deskryptorów numerycznych w problemie wykrycia anomalii sygnałów EKG. Przedstawia względne wartości testu dla 13 deskryptorów i wyników dekompozycji DWT na pięciu poziomach, przy czym zestaw 1 dotyczy sygnału aproksymacyjnego na piątym poziomie, natomiast zestawy od 2 do 6 dotyczą sygnałów szczegółowych na poszczególnych poziomach, poczynając od piątego a kończąc na pierwszym.

Widoczne są znaczące różnice wartości diagnostycznych poszczególnych deskryptorów, przy czym wartości te są w dużej mierze uzależnione od aktualnego poziomu dekompozycji. Zmiany te można zauważyć dla każdego deskryptora. Dla uzyskania najlepszych wyników rozpoznania należy wyselekcjonować te deskryptory, które prezentują największe powiązanie z rozpoznawaną

klasą. Na przykład deskryptor w postaci ilości punktów przekroczenia poziomu zerowego dla sygnału aproksymacyjnego (ale tylko dla niego) prezentuje bardzo małą wartość i może być odrzucony jako cecha diagnostyczna. Istotne staje się wyznaczenie progu ważności poszczególnych deskryptorów i ich wyboru jako cech diagnostycznych procesu. W wyniku wielu eksperymentów wstępnych spośród wielu deskryptorów utworzonych na podstawie DWT wybiera się ich ograniczoną liczbę, tworzącą zbiór cech diagnostycznych. Liczebność tego zbioru była ustalana na podstawie eksperymentów wstępnych.

Zastosowanie wyselekcjonowanych cech diagnostycznych w połączeniu z zestawem klasyfikatorów tworzących zespół pozwoliło na przeprowadzenie eksperymentów numerycznych rozpoznania anomalii w sygnałach EKG. Wyniki eksperymentów numerycznych w tej części pracy jak i w pozostałych będą zilustrowane poprzez zestaw miar jakości zdefiniowanych dla danych testujących nie uczestniczących w procesie uczenia. Obejmują one następujące definicje tych miar [5]:

- Dokładność rozpoznania zbioru danych (ACC )

$$ACC = \frac{TP}{TP+TN+FP+FN} \quad (3.5)$$

- Precyzja rozpoznania klasy (PREC)

$$PREC = \frac{TP}{TP+FP} \quad (3.6)$$

- Czułość klasyfikatora w rozpoznaniu (SENS), zwana w inżynierii technicznej również odtwarzalnością (oznaczenie REC)

$$SENS = \frac{TP}{TP+FN} \quad (3.7)$$

- Miara F1 definiowana w postaci

$$F1 = \frac{PREC \cdot SENS}{0.5(PREC+SENS)} \quad (3.8)$$

W dalszej części pracy wielkości te będą przedstawiane jako wartości średnie testowania dla wszystkich klas i uśrednione po wielokrotnych próbach uruchomienia procesu na danych uczących i testujących dobieranych losowo w każdej próbie.

### 3.4 Wyniki zastosowania modeli płytkich w wykrywaniu anomalii EKG

Wyznaczone deskryptory numeryczne poddane testowi  $\chi^2$  zostały uszeregowane według ich wartości testowej i podlegać będą wyborowi jako sygnały (atrybuty) wejściowe dla klasyfikatorów.

Zastosowane zostały następujące klasyfikatory, które według powszechnej opinii uchodzą za najlepsze:

- Gradient Boosting (GB),
- klasyfikator ADA
- las drzew losowych: random forest (RF) i extra random forest (ERF),
- maszyna SVM,
- klasyfikator oparty o proces gaussowski (GP),
- klasyfikator MLP,
- klasyfikator K-najbliższych sąsiadów (KNN),
- naiwny klasyfikator bayesowski (NB).

W tabeli 3.1 przedstawiono wyniki eksperymentów numerycznych w postaci miar jakości (ACC, SENS, PREC i F1) dla określonej wielkości populacji deskryptorów numerycznych. Wyniki podano dla  $n=78$  (pełny zestaw deskryptorów) oraz wyselekcjonowany (według miary  $\chi^2$ ) zestaw obejmujący 48, 24 i 12 najważniejszych deskryptorów. Pomniejszone zestawy reprezentują  $n$  najlepszych cech diagnostycznych (o wartościach największych testu). 70% dostępnego zbioru danych (wybranych losowo) zostało użyte podczas procesu uczenia, pozostałe 30% posłużyło do testowania modeli. Proces losowania podzbiorów połączony z etapem uczenia i testowania systemu był powtarzany 10-krotnie. Miary jakościowe modeli obejmują 9 klasyfikatorów poddanych treningowi i dotyczą średnich wyników testowania na zbiorze nie uczestniczącym w uczeniu.

Najlepsze wyniki uzyskano dla klasyfikatora lasu losowego drzew decyzyjnych (Extra Random Forest) przy użyciu 24 deskryptorów numerycznych wybranych w teście zgodności  $\chi^2$ . W zdecydowanej większości modeli, pełna liczba deskryptorów (78) powodowała spadek wydajności systemu. Jest to spowodowane tym, że zbyt duża liczba atrybutów wejściowych powoduje zjawisko tak zwanego „przeuczenia” (ang. overfitting) który objawia się spadkiem wartości miar jakości. Najlepsze okazały się klasyfikatory bazujące na lasach losowych, K-najbliższych sąsiadów oraz sieć SVM. Model SVM i KNN wykazał najmniejszą wrażliwość na liczbę deskryptorów numerycznych na wejściu modeli, model klasyfikatora Bayesa uzyskał najgorsze wyniki we wszystkich badanych przypadkach.

Tabela 3.1 Wyniki statystyczne eksperymentu przy zastosowaniu zbioru 9 klasyfikatorów i różnej liczby cech diagnostycznych procesu: 12, 24, 48, 78.

Liczba cech diagnostycznych		GB	ADA	ERF	RF	SVM	GP	MLP	KNN	NB
12	ACC [%]	94.19	83.79	96.07	94.99	93.42	89.76	81.91	95.27	81.3
	PREC [%]	92.68	81.78	94.93	93.55	91.78	87.55	81.8	93.53	78.53
	SENS [%]	93.84	86.18	95.75	94.56	92.82	88.26	76.53	94.57	75.41
	F1	93.26	83.92	95.34	94.05	92.30	87.90	79.08	94.05	76.94
24	ACC [%]	95.12	90.18	<b>97.78</b>	97.61	95.82	92.61	84.5	97.37	80.05
	PREC [%]	95.09	88.56	<b>97.54</b>	97.13	94.28	91.54	81.42	96.3	73.92
	SENS [%]	93.49	89.62	<b>97.20</b>	96.49	94.44	90.34	81.09	96.07	72.32
	F1	94.28	89.09	<b>97.37</b>	96.81	94.36	90.94	81.25	96.18	73.11
48	ACC [%]	93.00	91.88	96.14	92.81	90.79	93.85	85.51	95.85	73.71
	PREC [%]	93.25	93.38	96.58	94.37	93.22	93.50	86.64	95.30	69.08
	SENS [%]	88.90	87.37	94.27	88.74	84.95	89.43	80.05	94.45	65.86
	F1	91.02	90.28	95.41	91.47	88.89	91.42	83.21	94.87	67.43
78	ACC [%]	92.61	92.10	94.27	91.40	90.18	94.08	88.97	96.32	50.65
	PREC [%]	90.53	90.20	94.78	92.70	93.51	94.67	89.50	96.03	70.95
	SENS [%]	91.18	89.47	91.29	86.80	84.71	91.06	84.54	95.11	52.28
	F1	90.85	89.83	93.00	89.65	88.89	92.83	86.95	95.57	60.20

W następnym etapie poddano integracji wyniki działania 6 najlepszych klasyfikatorów, pomijając 3 najslabsze (ADA, MLP i NB) w zespole przy zastosowaniu wyselekcjonowanych 24 najlepszych cech diagnostycznych. Integrację zespołu przeprowadzono przy użyciu głosowania większościowego. W wyniku tej operacji uzyskano poprawę wskaźników jakości. Wyniki liczbowe tej operacji w porównaniu z najlepszym i najslabszym klasyfikatorem w zespole przedstawia tabela 3.2.

Tabela 3.2 Wyniki zastosowania zespołu 7 klasyfikatorów w rozpoznaniu anomalii EKG.

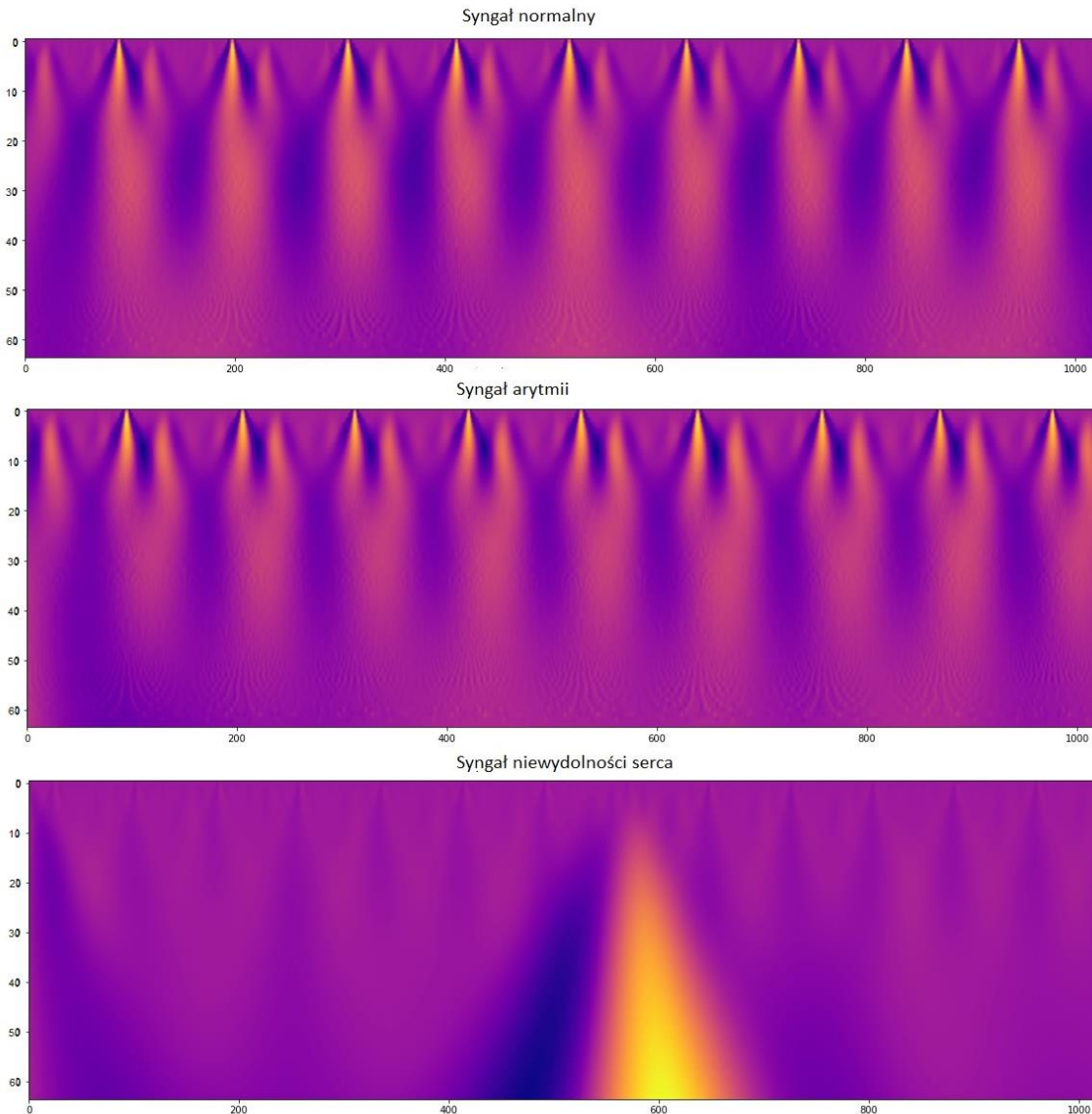
Parametr	Zespół	Najlepszy klasyfikator	Najslabszy klasyfikator
ACC [%]	<b>98.21</b>	97.78	90.18
PREC [%]	<b>97.98</b>	97.54	88.56
SENS [%]	<b>98.11</b>	97.20	89.62
F1	<b>97.88</b>	97.37	89.09

Dzięki zastosowaniu zespołu w każdym przypadku uzyskano poprawę wskaźników jakości. Zespół okazał się lepszy niż najlepszy jego członek (pomimo uwzględnienia wyników działania członków zespołu zdecydowanie słabszych).



### 3.5 Sieci głębokie w wykrywaniu anomalii

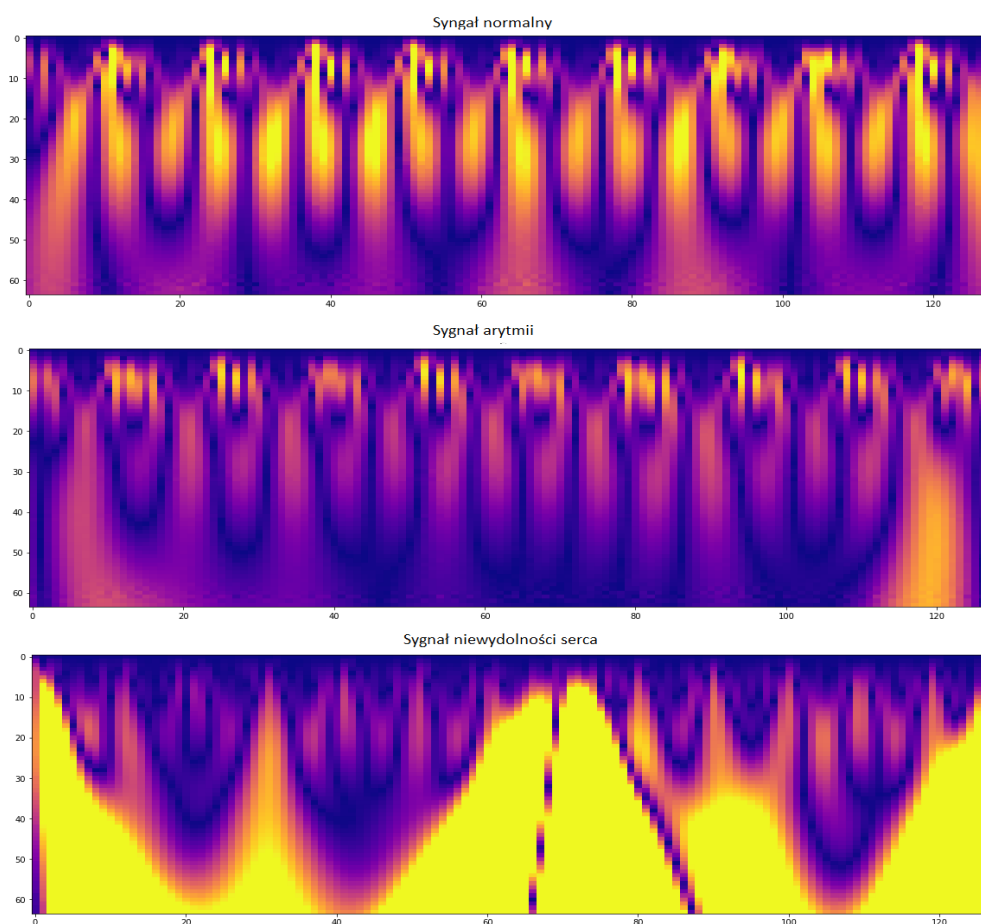
W drugim podejściu do wykrywania anomalii EKG wykorzystano sieci głębokie CNN [53] Atrybuty wejściowe dla tych sieci stanowią obrazy uzyskane w wyniku zastosowania ciągłej transformaty falkowej CWT [54].



Rys. 3.8 Wynik ciągłej transformacji falkowej sygnałów EKG dla trzech badanych grup sygnału. Zobrazowany wynik uzyskany został poprzez dekompozycję CWT przy użyciu falki „Mexican hat”.

Zastosowanie CWT dla sygnału EKG pozwoliło uzyskać dwuwymiarowy zestaw danych (obraz) gdzie oś rzędnych reprezentuje skala czasu  $a$  (odwrotnie proporcjonalna do częstotliwości sygnału) a oś odciętych przesunięcie  $b$  w dziedzinie czasu. Takie dane mogą być odzwierciedlone w postaci obrazu przedstawiającego zachowanie procesu EKG (rys. 3.8).

Rys. 3.8 przedstawia wynik ciągłej transformacji falkowej dla trzech grup badawczych sygnału. Zobrazowany wynik uzyskany został poprzez dekompozycję w skali 64 przy użyciu falki „Mexican hat”. Sygnał EKG został podzielony na sekcje o długości 1024 próbek, generując obraz o rozmiarze 64x1024. Analizując wyniki tych przekształceń, można zauważyć znaczące różnice dotyczące sygnału reprezentującego niewydolność serca. Pozostałe dwie grupy są zbliżone do siebie pod względem rozkładu wartości. Największe różnice obserwuje się w zakresie wysokiej skali (niskich częstotliwości).



Rys. 3.9 Wyniki transformacji CWT po redukcji rozdzielczości.

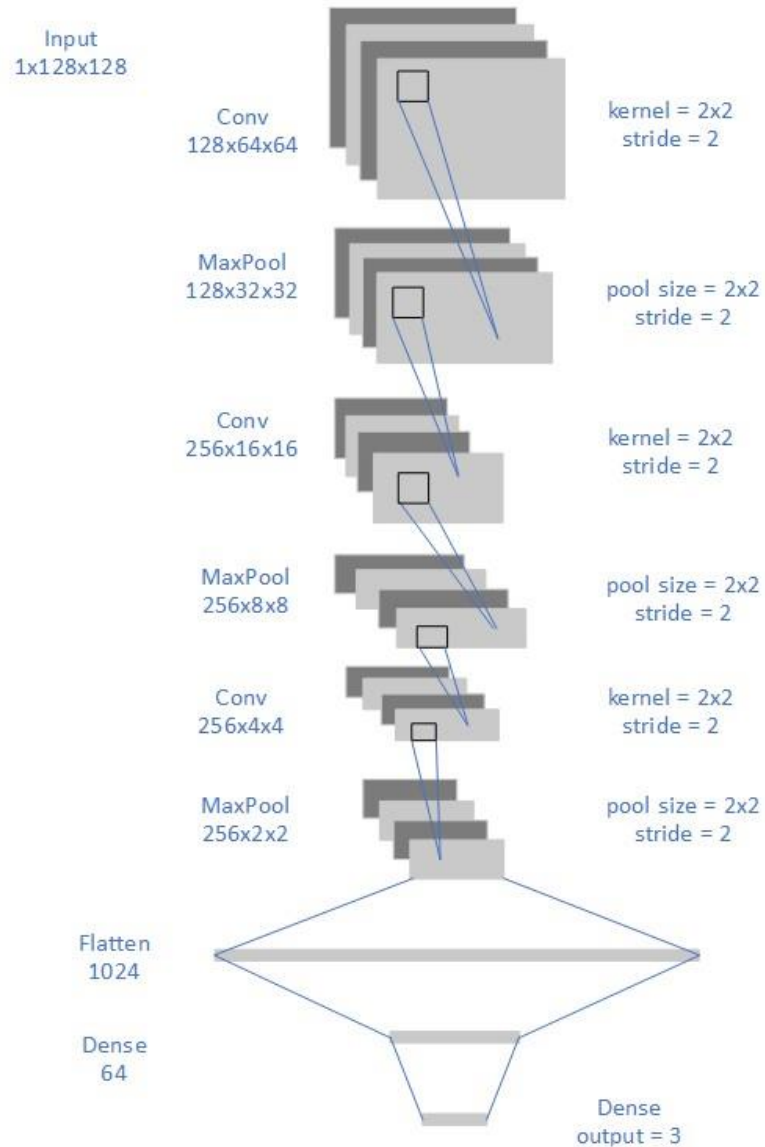
Badania eksperymentalne były przeprowadzone w rozprawie przy użyciu standardowego laptopa z procesorem graficznym. W przypadku sieci głębokich okazało się, że wymiar obrazów 64x1024 jest zbyt duży do przeprowadzenia pełnego uczenia sieci głębokiej od podstaw. Aby zmniejszyć zasoby potrzebne do obliczeń (proces trenowania sieci głębokiej tworzonej od

podstaw), inicjalne zestawy danych po transformacji falkowej zostały zmniejszone radykalnie do wymiaru 64x128. Wynik redukcji rozdzielczości obrazów z rys. 3.8 przedstawiono na rys. 3.9.

Obrazy uzyskane w wyniku transformacji CWT stanowiły wejście dla klasyfikatorów CNN. Klasyfikatory te są odpowiedzialne za wykrycie anomalii. Zmniejszenie rozmiaru pozwoliło zmniejszyć złożoność sieci CNN, ale jednocześnie pogorszyło reprezentatywność danych dla rozpoznania klas.

Stworzona na potrzeby eksperymentu sieć składała się z trzech warstw konwolucyjnych, aktywowanych funkcją ReLU i połączonych przy pomocy warstw MAX POOL. Trójwarstwowa struktura konwolucyjna została wybrana na podstawie wielu przeprowadzonych prób wstępnych. Ostateczny kształt struktury zastosowanej sieci CNN stworzonej i trenowanej od podstaw przedstawia rys. 3.10.

Pierwsza warstwa sieci została zaprojektowana z zastosowaniem 128 filtrów o rozmiarze 2x2 każdy. Jej wynikiem są obrazy reprezentowane w postaci 128x64x64. Rozmiar obrazów jest następnie zredukowany w warstwie MAX POOL do 128x32x32. Kolejna warstwa konwolucyjna oraz MAX POOL generuje zestaw obrazów o zredukowanym wymiarze jednocześnie podwajając ilość obrazów wejściowych (oznaczenie 256x8x8). Trzecia warstwa konwolucyjna i MAX POOL generuje 256 obrazów o rozmiarze 2x2 (oznaczenie 256x2x2). Następnie każdy z tych obrazów zostaje przekształcony w wektor o długości 1024, który pełni rolę wejścia dla w pełni połączonej struktury sieci (ang. *fully connected network*), dokonując ostatecznej klasyfikacji obrazu wejściowego [38].



Rys. 3.10 Struktura sieci głębokiej CNN stworzona od podstaw która została zastosowana w eksperymentach.

W opisanej sieci użyta została zastosowana funkcja klasyfikacyjna typu softmax [55]. W pełni połączona sieć płaska ma strukturę 1024-64-3. Sieć CNN była trenowana dla rozpoznawania 3 grup sygnałów użytych w eksperymencie. Wykorzystano algorytm uczący ADAM i podzbiory „mini batch” zawierające po 30 obrazów. W eksperymentach zbadano zastosowanie różnych funkcji falkowych (gaussowska, „Mexican hat” oraz Morleta). Wyniki klasyfikacji anomalii EKG, przy wykorzystaniu tych trzech funkcji falkowych i 4 różnych wartości skali (16, 32, 64 i 128) przedstawiono w tabeli 3.3.

Tabela 3.3 Wyniki statystyczne rozpoznania anomalii EKG przy zastosowaniu sieci CNN tworzonej od podstaw.

Skala	16			32			64			128		
Falka	ACC [%]	PREC [%]	SENS [%]	ACC [%]	PREC [%]	SENS [%]	ACC [%]	PREC [%]	SENS [%]	ACC [%]	PREC [%]	SENS [%]
Gauss8	69.62	70.53	68.33	71.93	72.37	71.16	81.00	81.31	80.55	81.42	81.69	81.03
Mex hat	72.96	73.47	72.19	75.98	76.38	75.57	80.16	80.51	79.81	<b>82.06</b>	<b>82.28</b>	<b>81.51</b>
Morlet	72.16	72.58	71.32	75.15	75.63	74.83	77.27	77.76	76.43	80.00	80.46	79.55

Najlepsze wyniki uzyskano przy użyciu funkcji „*Mexican hat*” i skali 128. Uzyskane rezultaty są jednak dalekie od oczekiwanych. Są przy tym zdecydowanie gorsze niż te otrzymane przez zastosowanie modeli płytkich. Można tu wymienić dwa główne powody. Pierwszym jest pogorszenie reprezentatywności danych po radykalnej (10-krotnej) kompresji danych. Drugim powodem jest zbyt mała liczebność danych uczących użyta w procesie doboru parametrów całej sieci. Dotyczy to szczególnie sygnałów arytmii i niewydolności serca (odpowiednio 36 i 30 badanych przypadków).

Dla uniknięcia tego problemu zdecydowano się na zastosowanie techniki „*transfer learning*”, wykorzystując dostępne w Internecie modele sieci głębokich CNN. Sieci te w procesie uczenia adaptują jedynie parametry warstwy w pełni połączonej, akceptując z góry wartości parametrów warstw lokalnie połączonych. W procesie tworzenia sieci wstępnie pre-trenowanych dostosowuje się jedynie ich końcową strukturę (ang. *fully connected substructure*) do aktualnego zadania klasyfikacyjnego, zmieniając standardową liczbę neuronów wyjściowych równą 1000 na aktualną liczbę rozpoznawanych klas (w pracy występują 3). Zastosowano zespół 8 sieci CNN o różnej strukturze. W skład zespołu wchodziły następujące sieci: (*alexnet, mobilenetv2, resnet50, efficientnetb0, squeezenet, googlenet, shufflenet, inceptionresnetv2*). Wynik działania zespołu ustalone się poprzez głosowanie większościowe przedstawia tabela 3.4.

Tabela 3.4 Wyniki statystyczne rozpoznania anomalii EKG przy zastosowaniu zespołu 8 pre-trenowanych sieci CNN.

Skala	16			32			64			128		
Falka	ACC [%]	PREC [%]	SENS [%]	ACC [%]	PREC [%]	SENS [%]	ACC [%]	PREC [%]	SENS [%]	ACC [%]	PREC [%]	SENS [%]
Gauss8	84.05	85.66	82.06	87.77	84.80	87.46	94.44	95.88	94.73	93.70	93.36	92.52
Mex hat	83.88	84.13	85.00	87.41	91.40	89.93	<b>97.28</b>	<b>93.54</b>	<b>92.98</b>	95.08	92.65	93.94
Morlet	83.15	83.37	85.18	84.64	86.73	87.56	90.12	95.12	85.98	93.14	93.50	93.01

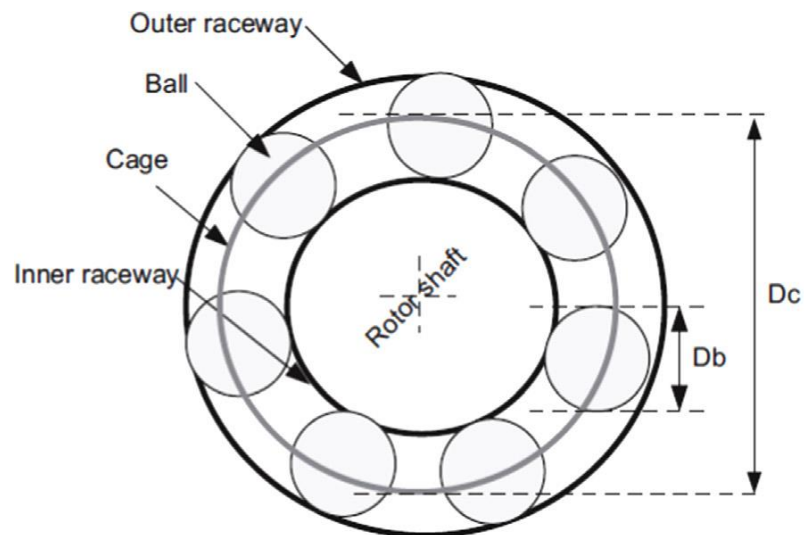
Najlepsze wyniki uzyskano teraz dla skali 64 i falki „Mexican hat”. Dzięki zastosowaniu zespołu pre-trenowanych sieci nastąpiła istotna poprawa wyników numerycznych w stosunku do sieci CNN stworzonej od podstaw. Uzyskane wartości miar jakości są porównywalne z najlepszymi wynikami uzyskanymi przez zespół sieci płytkich.

## 4 Wykrywanie uszkodzeń łożysk

### 4.1 Wprowadzenie do analizy uszkodzeń

Kolejnym problemem rozważanym w pracy jest wykrywanie anomalii w działaniu łożysk tocznych przy wykorzystaniu zarejestrowanych sygnałów akcelerometrycznych i zastosowaniu opracowanej techniki przetwarzania danych. Anomalia jest utożsamiona z określonym rodzajem uszkodzenia łożyska. Łożyska w mechanizmach wirujących stanowią istotny element w prawidłowym działaniu urządzenia. Ich uszkodzenie może spowodować wadliwe działanie całych systemów. Konieczne staje się wczesne rozpoznawanie zmian stanu dynamicznego łożyska maszyny, rodzaju i poziomu uszkodzeń, aby jak najwcześniej podjąć odpowiednie działania zapobiegawcze. Wczesne wykrycie symptomów uszkodzenia pozwoli na przeprowadzenie remontów uwarunkowanych stanem technicznym maszyny i zapobiegnie kosztownym remontom poawaryjnym całego systemu [57].

Uszkodzenie łożyska może być zlokalizowane w różnych jego elementach, w tym uszkodzenie bieżni wewnętrznej lub zewnętrznej, uszkodzenie koszyczka czy też kulki. Geometryczna lokalizacja poszczególnych elementów łożyska, które mogą podlegać uszkodzeniu przedstawiona jest na rys. 4.1 [58].



Rys. 4.1 Ilustracja poszczególnych elementów łożyska, których uszkodzenia będą rozważane w pracy [58].

Najczęściej stosowanym sposobem wykrycia uszkodzenia łożyska jest analiza sygnałów drgań rejestrowanych przez czujniki akcelerometryczne. Powstanie uszkodzenia przejawia się w anomalnej postaci ciągu takich sygnałów, różniących się od stanu normalnego łożyska.

Rejestrowane postacie sygnałów uzależnione są od lokalizacji defektu, szybkości wirowania maszyny oraz od konstrukcji geometrycznej samego łożyska. Ich analiza przy zastosowaniu odpowiednich narzędzi umożliwia wczesne wykrycie anomalii zarejestrowanego przebiegu czasowego i rozpoznanie rodzaju uszkodzenia [59].

## 4.2 Baza danych w eksperymentach

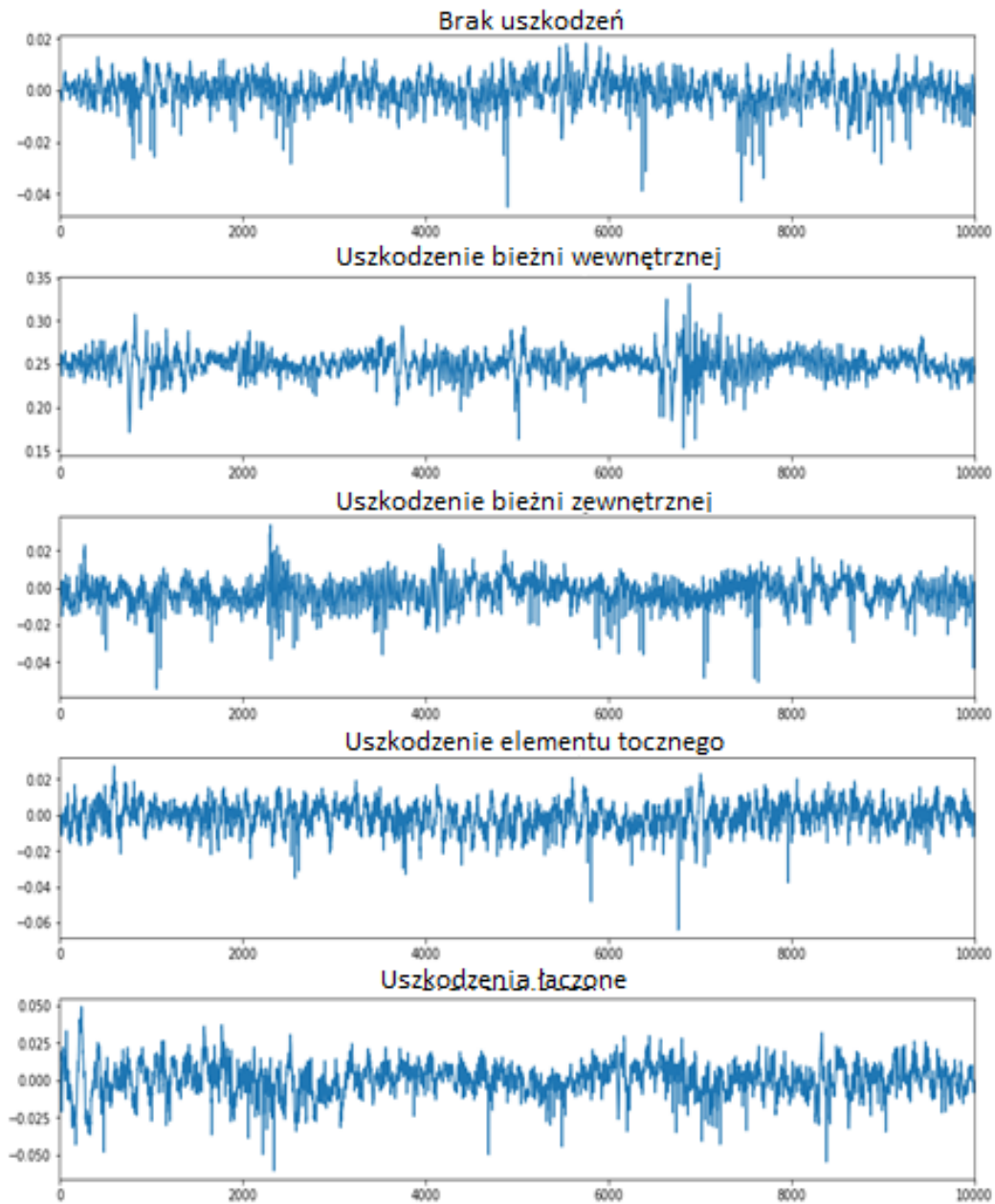
W badaniach użyto bazy danych stworzonej na Uniwersytecie w Ottawie i zamieszczonej w portalu „*Mendeley Data*” [60, 61]. Dane zostały zebrane z łożysk w różnych stanach eksploatacji przy różnych prędkościach obrotowych. Całość składa się z 60 zestawów danych. Reprezentują one sygnał z akcelerometru (wibracje) które mają przełożenie zarówno na typ uszkodzenia jak i prawidłowo działające łożysko. Stany łożyska rozważone w pracy obejmują:

- defekty bieżni zewnętrznej,
- defekty bieżni wewnętrznej,
- defekty elementów tocznych,
- defekty łączone,
- łożysko bez stwierdzonych defektów.

Wszystkie przypadki zawierają po 12 zestawów danych, każdy obejmujący rejestracje o długości 10 sekund (2 000 000 próbek). Sygnały zostały spróbkowane z częstotliwością 200 kHz.

Stanowisko do zebrania danych łożyska ER16K [61] składało się z wałka sterowanego silnikiem elektrycznym, akcelerometru typu ICP oraz enkodera obrotów do kontrolowania prędkości obrotowej wałka. Minimalna częstotliwość obrotów wałka wynosiła 10.3 Hz, a maksymalna 29 Hz. Przykładowe przebiegi sygnałów reprezentujące rozważane 5 klas stanów łożyska zostały przedstawione na rys. 4.2. Jako, że sygnały zostały zarejestrowane w rzeczywistym środowisku pomiarowym, zawierają również określoną składową szumów. Spowodowane są one czynnikami takimi jak niestabilność obrotów wałka, zewnętrznymi wibracjami, szumami nadmiarowymi urządzeń pomiarowych. Wpływają zarówno na kształty sygnałów jak i parametry statystyczne charakteryzujące te sygnały. Ponadto, w każdej ramce pomiarowej utworzonej na zbiorze danych zawartości szumowe mogą być znacznie różniące w ramach jednej klasy, co znacznie utrudnia rozpoznanie typu anomalii.





Rys. 4.2 Przykładowe przebiegi czasowe sygnałów przy różnych stanach uszkodzeń łożyska.

Wartości wybranych parametrów statystycznych charakteryzujących sygnały drgań należących do różnych klas uszkodzeń zostały przedstawione w tabeli 4.1. Analiza ich wartości pozwala zauważyć statystyczne różnice wewnątrz każdej klasy jak również istotne różnice między klasowe.

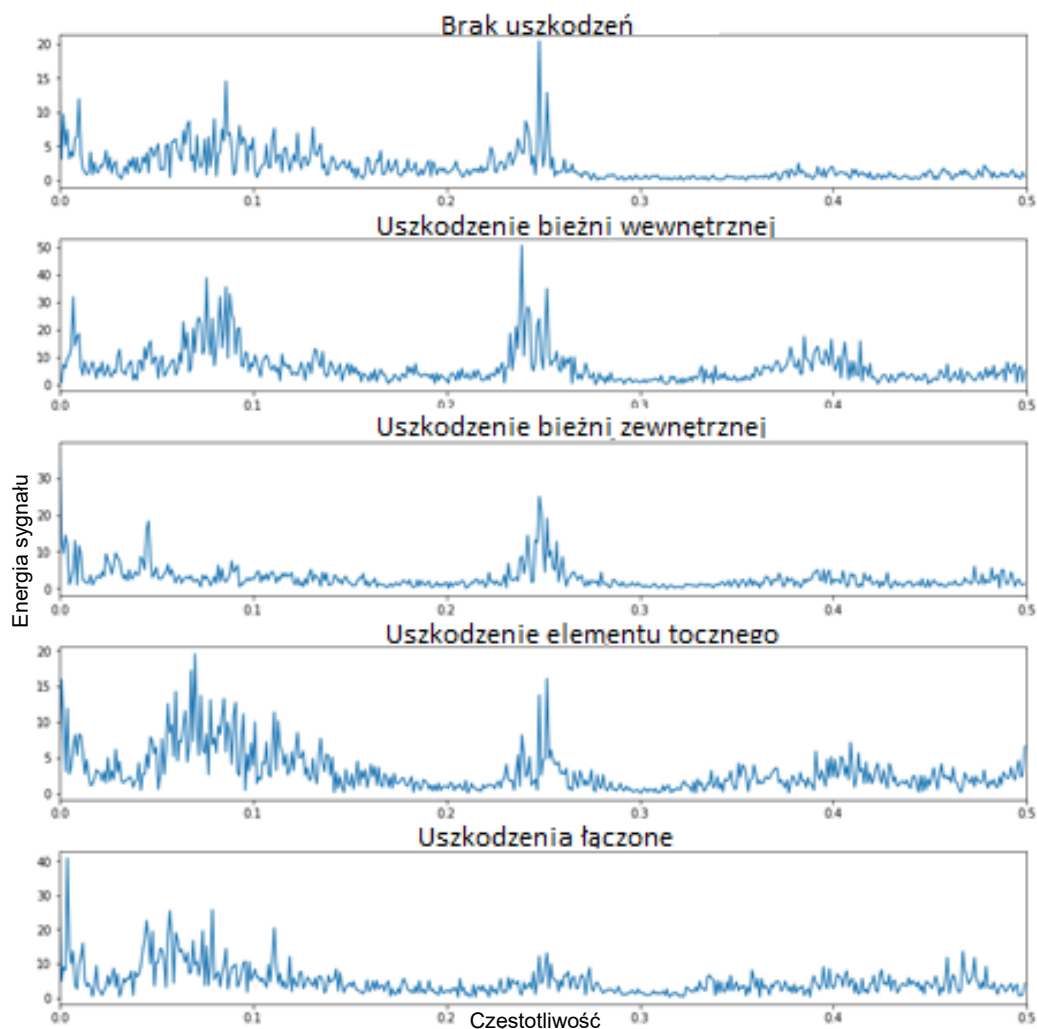
Tabela 4.1 Wybrane parametry statystyczne charakteryzujące sygnały czujników dla różnego stanu łożyska.

Wartości (min, max)	Średnia	Odchylenie standardowe	Energia sygnału	Skośność	Kurtoza
Brak uszkodzeń	-0.0001, 0.0018	0.0045, 0.0149	0.0001, 0.0123	-1.5658, 0.0465	0.3222, 15.8031
Defekty bieżni zewewnętrznej	-0.0019, 0.0031	0.0052, 0.0119	0.0007, 0.0069	-1.1031, 0.1497	1.5659, 9.1148
Defekty bieżni wewnętrznej	0.0002, 0.2458	0.0132, 0.0768	0.0006, 0.2776	-0.2303, 0.1100	5.9528, 30.1610
Defekty elementów tocznych	-0.0006, -0.0005	0.00538, 0.0289	0.0001, 0.0810	-0.4940, 0.2327	0.0731, 21.2105
Defekty łączone	0.0006, 0.0015	0.0128, 0.0535	0.0002, 0.1456	-0.0617, 1.2421	19.3623, 77.6372

### 4.3 Analiza częstotliwościowa sygnałów uszkodzeń łożysk

Analiza częstotliwościowa oparta o transformację Fouriera jest dość często używanym narzędziem w analizie szeregów czasowych. Składowe harmoniczne będące efektem transformacji mogą posłużyć jako cechy diagnostyczne stanu urządzenia. Rys. 4.3 przedstawia rozkład widma częstotliwości dla każdego z pięciu typów stanów łożyska. Wyniki odpowiadają przebiegom czasowym z rys. 4.2. Poszczególne widma częstotliwościowe różnią się zasadniczo, składowe harmoniczne są przesunięte względem siebie w całym paśmie, występują znaczne różnice w zakresie wyższych harmonicznych dla uszkodzonego łożyska. W przypadku defektów łączonych, składowe środka pasma częstotliwości są niższe niż w pozostałych przypadkach jak również widoczna jest wyższa amplituda składowych wolnozmiennych.

Jakkolwiek transformacja Fouriera jest dość często stosowana w rozpoznaniu uszkodzeń łożyska efekty jej zastosowania w dużej mierze zależą od stacjonarności zarejestrowanego procesu, gdyż cechy diagnostyczne bazujące na wynikach transformacji Fouriera dobrze reprezentują proces jedynie dla sygnałów o niskiej zmienności parametrów. Ponadto, przy stosowaniu nieskończonej funkcji bazowej traci się informację o zależności czasowo-częstotliwościowej sygnału [62].



Rys. 4.3 Rozkład widma częstotliwości dla każdego z pięciu typów stanów łożyska.

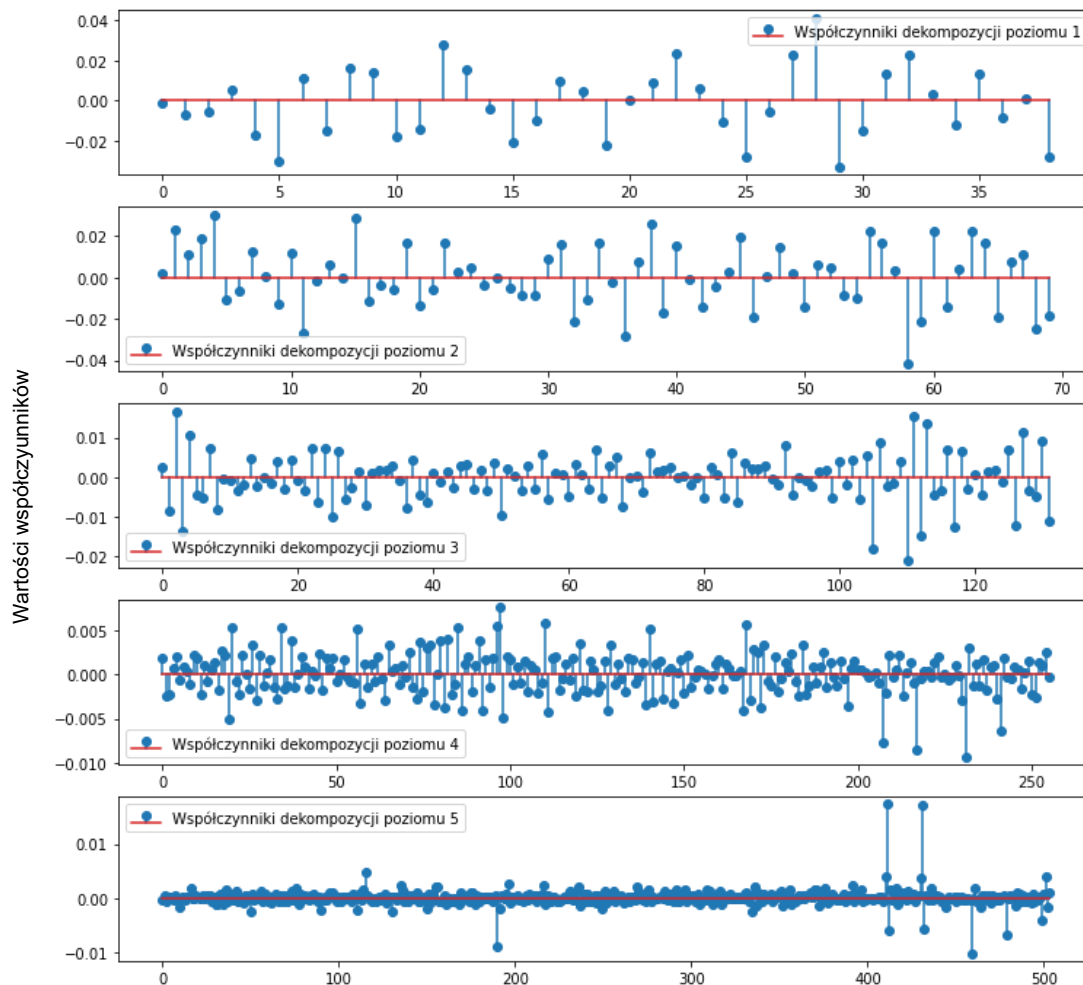
Dlatego podejście Fouriera w tym przypadku ma ograniczone zastosowanie. Zamiast transformacji Fouriera można zastosować transformację falkową, która umożliwi analizę sygnału jednocześnie w dziedzinie czasu i częstotliwości. Ta transformacja, omówiona w rozdziale drugim, stanowić będzie w rozprawie główne źródło cech diagnostycznych procesu.

Podobnie jak w przypadku sygnałów EKG zaproponowano dwa systemy wykrywania anomalii. Pierwszy bazuje na dyskretniej transformacji falkowej do generacji cech diagnostycznych. Cechy te stanowią atrybuty wejściowe dla zespołu klasyfikatorów klasycznych, identycznie jak w przypadku EKG. System drugi wykorzystuje ciągłą transformację falkową, która przetwarza sygnały łożysk na obrazy wynikowe transformacji. Obrazy te stanowią wejście dla zespołu klasyfikatorów głębokich CNN, odpowiedzialnych jednocześnie za generację cech diagnostycznych i końcową klasyfikację.

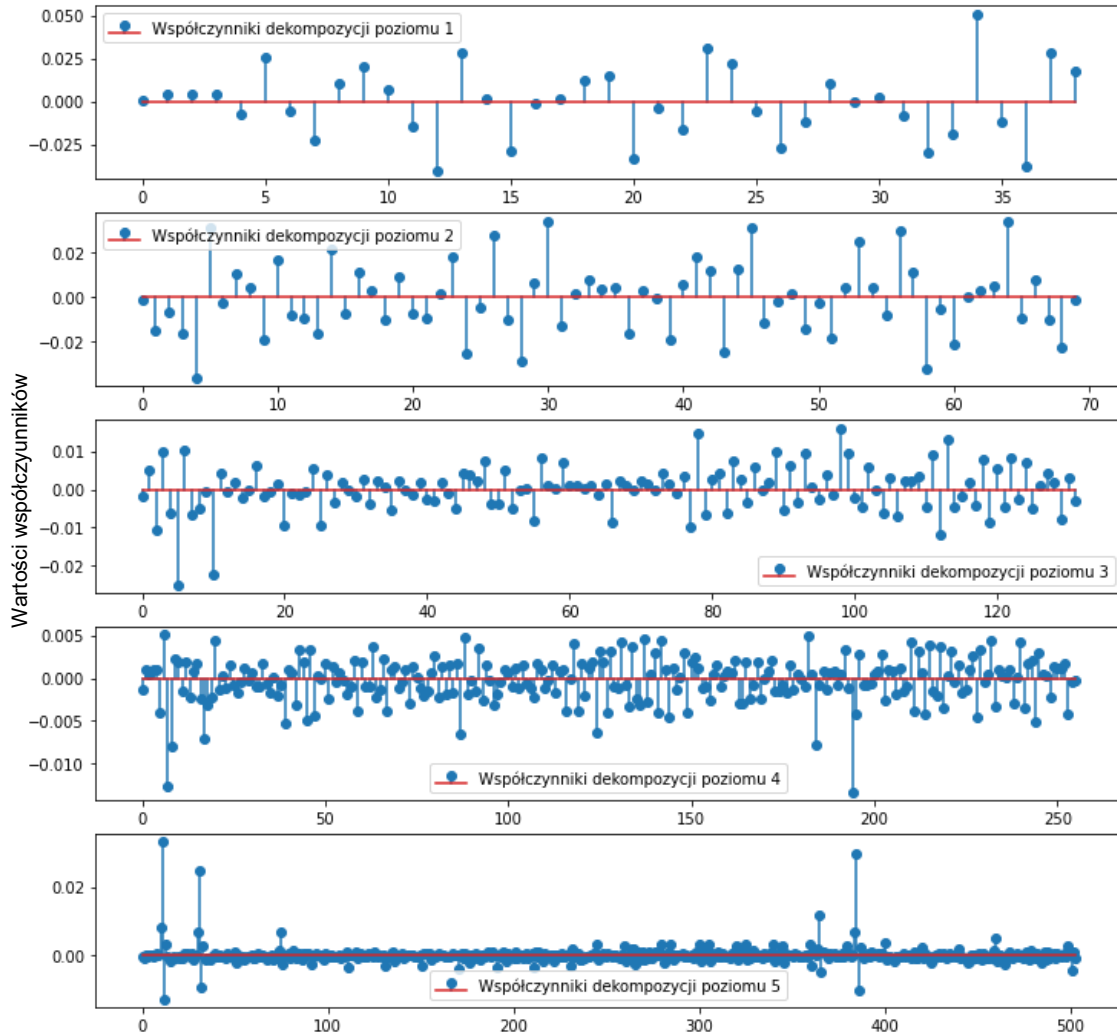
## 4.4 Zastosowanie zespołu klasyfikatorów płytkich w wykrywaniu uszkodzeń

### 4.4.1 Generacja cech diagnostycznych z zastosowaniem DWT

Oryginalne zestawy sygnałów zostały podzielone na odcinki od długości 1 sekundy, a następnie poddane transformacji falkowej DWT na 5 poziomach dekompozycji. Przykładowe wyniki tej analizy dla dwu sygnałów: jeden reprezentujący stan normalny łożyska i drugi odpowiadający uszkodzeniu łącznemu przedstawione są na rys. 4.4 i 4.5. Widoczne są istotne różnice w rozkładzie wartości na poszczególnych poziomach odpowiadające łożysku w stanie normalnym i uszkodzonym.



Rys. 4.4 Przykładowy wynik analizy DWT sygnału łożyska w stanie normalnym na pięciu poziomach dekompozycji.



Rys. 4.5 Przykładowy wynik analizy DWT sygnału łożyska przy 4 typach jednoczesnych uszkodzeń na pięciu poziomach dekompozycji.

W następnym kroku wprowadzono zunifikowany rozmiar współczynników na każdym poziomie do 256 próbek. W wyniku zastosowania 5-poziomowej dekompozycji otrzymano 5 szeregów czasowych sygnałów szczegółowych plus sygnał aproksymacyjny na ostatnim piątym poziomie.

Wyniki dekompozycji poddano analizie statystycznej, podobnej do zastosowanej uprzednio w analizie sygnałów EKG. W ten sposób dla każdego poziomu dekompozycji wygenerowano 13 deskryptorów numerycznych obejmujących: wartość średnią, wartość średniokwadratową, medianę, odchylenie standardowe, kurtozę, skośność, percentyle 5%, 25%, 75%, 95%, ilość punktów przekroczenia poziomu zerowego, ilość punktów przekroczenia wartości średniej oraz entropię. W efekcie wygenerowano 78 deskryptorów statystycznych obejmujących wszystkie

sześć szeregów czasowych (pięć poziomów sygnałów szczegółowych i jeden poziom sygnału aproksymacyjnego na piątym poziomie). Biorąc pod uwagę różną wartość diagnostyczną reprezentowaną przez poszczególne deskryptory następnym etapem jest selekcja ograniczonej liczby  $n$  deskryptorów o wartościach największych testu, eliminując pozostałe.

Deskryptory te poddano selekcji przy zastosowaniu testu  $\chi^2$ , dla uszeregowania ważności poszczególnych deskryptorów [63]. Tak wyselekcjonowany zbiór deskryptorów stanowił zestaw cech diagnostycznych podawanych na wejście zespołu klasyfikatorów, odpowiedzialnych za wykrycie stanu anomalnego procesu.

#### **4.4.2 Wyniki eksperymentów numerycznych**

Badania przeprowadzono w ramach dwu zadań. W zadaniu pierwszym, rozpoznaniu podlegały jedynie dwie klasy, jedna reprezentująca prawidłowo pracujące łożysko i druga reprezentująca łożysko uszkodzone (niezależnie od rodzaju uszkodzenia). Zadanie drugie polegało na rozpoznaniu pięciu klas – cztery klasy uszkodzeń łożyska i jedna klasa reprezentująca łożysko sprawne. Eksperyment przeprowadzono dla różnych postaci falek macierzystych. Wstępne badania miały za zadanie wyłonić najlepszą funkcję falkową. Spośród wielu dostępnych funkcji przebadano cztery rodzaje, w tym symlet (Sym9), Daubechies (db3), Haar and coiflet (coif9). Do treningu klasyfikatorów użyto 70% danych, pozostawiając do testowania pozostałe 30%. Tabela 4.2 przedstawia wyniki numeryczne w rozpoznaniu 2 klas danych. Są one przedstawione w postaci czterech miar jakości (ACC, PREC, SENS i F1 – wartości uśrednione po wszystkich klasach) uzyskane dla sześciu klasyfikatorów i zespołu stworzonego z nich przy zastosowaniu wszystkich 78 cech diagnostycznych) i dwóch klas docelowych.

Ostatnia kolumna w tabeli przedstawia wynik głosowania większościowego dla zespołu klasyfikatorów złożonych z sześciu jednostek (tych samych co w rozpoznaniu sygnałów EKG) charakteryzujących się najlepszymi wynikami indywidualnymi. Wyniki rozpoznania klas przez poszczególne klasyfikatory są na wysokim poziomie, tym nie mniej połączenie ich w zespole pozwoliło na dalszy wzrost wartości wskaźników jakości. Można zauważyć, że funkcja falkowa Coif9 może być uważana za najbardziej odpowiednią do postawionego zadania, choć pozostałe niewiele jej ustępują.

Tabela 4.2 Wyniki działania zespołu klasyfikatorów płytkich w zadaniu rozpoznania 2 klas przy zastosowaniu wszystkich 78 cech diagnostycznych dla różnego wyboru falki macierzystej.

Funkcja falkowa		GB	ERF	RF	SVM r	GP	KNN	Ensemble
Sym9	ACC [%]	99.50	99.58	99.75	99.50	97.78	97.61	99.81
	PREC [%]	99.43	99.34	98.38	99.43	97.54	97.13	99.47
	REC [%]	97.82	98.76	97.90	98.64	97.20	96.49	98.98
	F1	98.62	99.05	98.14	99.03	97.37	96.81	99.22
Db3	ACC [%]	99.58	99.63	99.79	99.58	96.14	97.98	99.91
	PREC [%]	99.54	99.46	98.04	99.54	96.58	99.13	99.60
	REC [%]	99.60	98.53	98.40	99.21	99.79	99.45	99.82
	F1	99.57	98.99	98.22	99.37	98.16	99.29	99.71
Haar	ACC [%]	99.63	99.71	99.83	99.63	98.43	99.13	99.87
	PREC [%]	99.00	99.82	99.27	99.00	98.16	99.69	99.89
	REC [%]	98.63	99.62	98.61	97.79	98.03	98.63	99.70
	F1	98.81	99.72	98.94	98.39	98.09	99.16	99.79
Coif9	ACC [%]	99.13	99.92	100.00	99.13	98.03	98.54	<b>99.95</b>
	PREC [%]	98.79	99.95	98.43	98.79	99.67	99.31	<b>99.95</b>
	REC [%]	97.53	98.22	97.83	97.82	99.62	99.88	<b>99.93</b>
	F1	98.16	99.08	98.13	98.30	99.64	99.59	<b>99.91</b>

Kolejna część eksperymentu miała na celu sprawdzenie, jak liczba deskryptorów statystycznych wpływa na wyniki klasyfikacji. Tabela 4.3 przedstawia uzyskane wartości miar jakości dla trzech różnych populacji cech ( $n=16$ ,  $n=32$ ,  $n=78$ ) i falki macierzystej Coif9. Wybór deskryptorów opierał się o test  $\chi^2$  analogicznie jak w rozdziale 3. Najbardziej czuły na zmiany liczby cech diagnostycznych okazały się klasyfikatory SVM i GP. Na wyniki działania pozostałych klasyfikatorów liczba użytych cech miała marginalny wpływ. Model drzew losowych ERF pozwolił uzyskać praktycznie idealne rozpoznanie klas dla 32 cech. Zespół klasyfikatorów tylko minimalnie polepszył działanie systemu w stosunku do ERF, przy czym różnice wyników w stosunku do zastosowania 78 cech są nieistotne. Badania tego typu zostały powtórzone przy rozpoznaniu 5 klas (4 klasy reprezentujące różne uszkodzenia i jedna odpowiadająca stanowi normalnemu łożysk).

Tabela 4.3 Wyniki działania zespołu klasyfikatorów płytkich w zadaniu rozpoznania 2 klas przy zastosowaniu ograniczonej liczbie cech diagnostycznych przy wyborze Coif9 jako falki macierzystej.

N wybranych parametrów		GB	ERF	RF	SVM	GP	KNN	Ensemble
16	ACC [%]	99.33	99.56	99.22	96.78	96.89	98.33	99.66
	PREC [%]	99.36	99.58	99.25	96.75	96.88	98.48	99.6
	RECALL [%]	99.01	99.22	99.01	97.19	97.30	98.21	99.39
	F1	99.18	99.40	99.13	96.97	97.09	98.34	99.49
32	ACC [%]	99.11	99.99	99.22	99.33	99.00	98.67	<b>99.99</b>
	PREC [%]	99.14	99.99	99.24	99.28	98.94	98.55	<b>99.99</b>
	RECALL [%]	99.23	99.99	99.23	99.30	98.95	98.60	<b>99.99</b>
	F1	99.18	99.99	99.23	99.29	98.94	98.57	<b>99.99</b>
78	ACC [%]	99.63	99.71	99.83	99.63	98.43	99.13	99.89
	PREC [%]	99.00	99.82	99.27	99.00	98.16	99.69	99.86
	RECALL [%]	98.63	99.62	98.61	97.79	98.03	98.63	99.87
	F1	98.81	99.72	98.94	98.39	98.09	99.16	99.86

Tabela 4.4 prezentuje wyniki jakościowe klasyfikacji zespołu klasyfikatorów złożonych z 6 członków (identycznie jak w poprzednim eksperymencie 2-klasowym) przy użyciu DWT i założeniu 5 klas anomalii podlegających rozpoznaniu (wartości precyzji i czułości są uśrednione dla wszystkich klas).

Tabela 4.4 Wyniki działania zespołu klasyfikatorów płytkich w zadaniu rozpoznania 5 klas przy zastosowaniu ograniczonej liczbie cech diagnostycznych i wyborze różnych funkcji falkowych.

N wybranych parametrów	16			32			48			78		
	ACC [%]	PREC [%]	REC [%]	ACC [%]	PREC [%]	REC [%]	ACC [%]	PREC [%]	REC [%]	ACC [%]	PREC [%]	REC [%]
Funkcja falkowa												
Sym9	93.33	94.86	92.22	96.67	95.63	95.21	99.44	98.45	99.11	99.41	98.45	99.11
Db3	93.89	94.89	92.78	93.89	93.21	92.87	98.89	99.44	99.44	96.67	97.77	97.77
Haar	96.67	96.65	96.11	92.78	91.67	94.44	98.88	98.89	98.89	99.44	98.33	96.11
Coif9	80.00	80.92	77.78	98.33	98.33	98.33	<b>99.44</b>	<b>99.44</b>	<b>99.44</b>	97.22	97.22	97.22

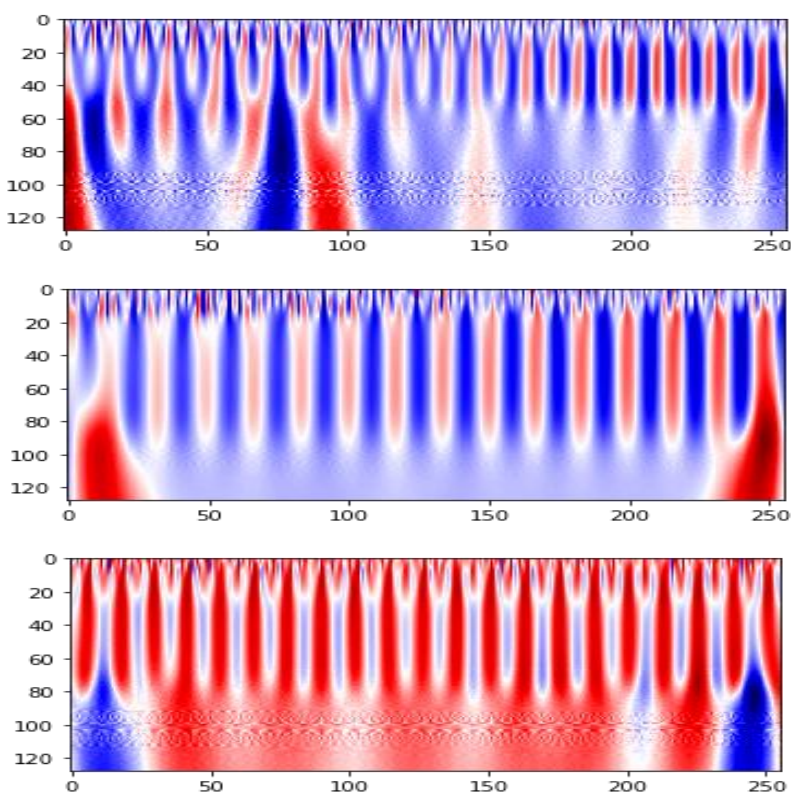
Zbadano zastosowanie różnych funkcji falkowych przy różnej liczebności cech diagnostycznych. Analiza wyników wskazuje, że tym razem optymalne jest zastosowanie 48 cech wybranych przy użyciu testu  $\chi^2$  i zastosowaniu funkcji falkowej Coif9 (podobnie jak w przypadku 2 klas). Interesujący jest fakt, że wartości liczbowe miar jakości przy rozpoznawaniu 5 klas są zbliżone (nieco tylko gorsze) do przypadku rozpoznania dwóch klas. Można stąd wysnuć



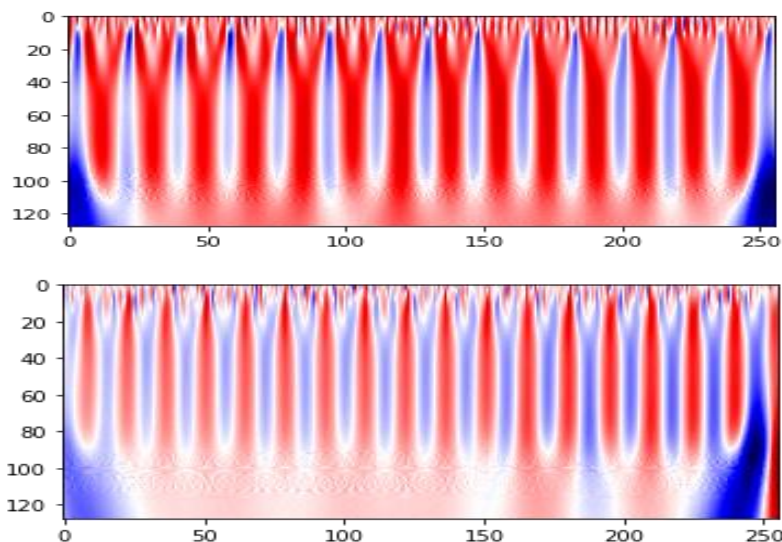
wniosek, że w tych systemach klasyfikacji zasadniczy wpływ ma sama zastosowana transformacja DWT i sposób generacji cech diagnostycznych.

#### 4.5 Zastosowanie sieci głębokich w wykrywaniu anomalii łożysk

W kolejnym eksperymencie zastosowano technikę sieci głębokich w rozpoznaniu anomalii działania łożyska. Tym razem wejście dla sieci stanowią obrazy uzyskane przy użyciu dekompozycji CWT. W wyniku jej zastosowania 1-wymiarowe wektory reprezentujące sygnały czujników (podzielone na odcinki o długości 2000 próbek) zostały przekształcone w obrazy reprezentujące wyniki CWT dla różnej skali i przesunięcia. Skala transformacji zmieniała się do 128. Przykładowe zobrazowanie wyników transformacji 5 sygnałów reprezentujących różne stany łożyska i dekompozycji w skali do 128 (oś x przeskalowana do 256) przy użyciu falki Morleta przedstawia rys. 4.4.



Rys. 4.4 Przykładowe zobrazowanie wyniku transformacji CWT z falką Morleta dla sygnałów łożysk przy dekompozycji w skali 128. Obrazy od góry do dołu reprezentują: defekty bieżni zewnętrznej, defekty bieżni wewnętrznej, defekty elementów tocznych, defekty łączone, łożysko bez stwierdzonych defektów.



Rys. 4.4 Przykładowe zobrazowanie wyniku transformacji CWT z falką Morleta dla sygnałów łożysk przy dekompozycji w skali 128. Obrazy od góry do dołu reprezentują: defekty bieżni zewnętrznej, defekty bieżni wewnętrznej, defekty elementów tocznych, defekty łączone, łożysko bez stwierdzonych defektów.

W pierwszej kolejności przetestowano sieć CNN tworzoną od podstaw. Zastosowano strukturę identyczną jak w badaniu anomalii sygnałów EKG (rys. 3.10). W tabeli 4.5 przedstawiono wyniki klasyfikacji 2-klasowej (łożysko w stanie normalnym przeciwko skumulowanemu zbiorowi uszkodzeń) oraz 5-klasowej (rozpoznanie każdego rodzaju uszkodzenia osobno) przy różnej skali dekompozycji CWT. Wyniki przedstawiono dla najlepszego wyboru funkcji falkowej (*Mexican hat*).

Tabela 4.5 Wyniki rozpoznania 2 i 5 klas w problemie wykrywania uszkodzeń łożyska przy wykorzystaniu sieci CNN z rys. 3.10 trenowanej od podstaw.

Skala	16			32			64			128		
Rodzaj klasyfikacji	ACC [%]	PREC [%]	REC [%]	ACC [%]	PREC [%]	REC [%]	ACC [%]	PREC [%]	REC [%]	ACC [%]	PREC [%]	REC [%]
2 klasy	51.66	87.27	26.66	50.00	79.24	23.33	53.88	86.11	34.44	<b>62.61</b>	<b>85.11</b>	<b>35.55</b>
5 klas	23.63	61.98	23.83	26.88	61.11	20.76	41.09	62.77	22.21	<b>51.40</b>	<b>67.27</b>	<b>23.42</b>

Uzyskane wartości badanych miar jakości są dalekie od oczekiwanych, niezależnie od wyboru wielkości zastosowanej skali. Głównym powodem niepowodzenia jest zbyt mała populacja danych uczących w porównaniu do złożoności sieci CNN. Nawet przy stosunkowo ubogiej strukturze sieci CNN (3 warstwy konwolucyjne) liczba par uczących (równa w eksperymencie 600) była zbyt mała dla osiągnięcia akceptowalnych wyników.

W związku z tym w ostatniej części tego eksperymentu zastosowano modele pre-trenowane (transfer learning), które z definicji są mniej wrażliwe na wielkość populacji danych uczących. Zastosowano zespół złożony z 8 pre-trenowanych sieci (*alexnet, mobilenetv2, resnet50, efficientnetb0, squeezenet, googlenet, shufflenet, inceptionresnetv2*). Wynik działania zespołu jest ustalany poprzez głosowanie większościowe. Ponownie sygnały zostały poddane transformacji CWT, tym razem wyłącznie dla skali 128 która uprzednio pozwalała uzyskać najlepsze efekty. Uzyskano obrazy o rozmiarze 128x2000, przeskalowane automatycznie na wymagany dla sieci przetrenowanych wymiar (najczęściej 224x224).

Wyniki klasyfikacji problemu 2-klasowego oraz 5-klasowego dla falki Morleta i zespołu 8 sieci przedstawiono w tabeli 4.6. Tabela przedstawia średnie wyniki zespołu uzyskane w 10 próbach po integracji metodą głosowania większościowego. Obejmują one wartości średnie i odchylenie standardowe wyników poszczególnych członków zespołu jak również wyniki najlepszego i najgorszego członka zespołu oraz wyniki całego zespołu po integracji.

Tabela 4.6 Wyniki rozpoznania stanu łóżysk przy 2 i 5 klasach podlegających rozpoznaniu przy wykorzystaniu sieci CNN pre-trenowanych w trybie transfer learning.

Miara jakości	2 klasy			5 klas		
	ACC [%]	PREC [%]	SEN [%]	ACC [%]	ACC [%]	SEN [%]
Zespół po integracji	<b>99.99</b>	<b>99.93</b>	<b>99.77</b>	<b>99.59</b>	<b>99.26</b>	<b>99.03</b>
Średnia członków	98.41	98.05	97.76	98.57	98.35	98.33
Odchylenie standardowe członków	2.21	1.81	2.17	1.07	2.15	1.70
Najlepszy indywidualny	98.83	98.91	99.10	99.37	99.02	98.90
Najgorszy indywidualny	98.04	97.75	97.24	97.88	97.79	98.01

Otrzymane w ten sposób wyniki są bliskie ideału (dokładność powyżej 99% niezależnie od liczby klas). Sieci pre-trenowane znacznie lepiej tolerują małą populację danych uczących, gdyż w douczaniu sieci procesowi adaptacji podlegają jedynie wagi neuronów w warstwie w pełni połączonej co przekłada się na wielokrotną redukcję liczby adaptowanych parametrów w stosunku do sieci trenowanej od podstaw). Parametry warstw konwolucyjnych nie podlegają adaptacji, zmniejszając w ten sposób liczbę adaptowanych parametrów, co zwiększa zgodność generalizacji

sieci. Można zauważyć wysoką zgodność wyników sieci głębokiej i zespołu sieci płytkich (wszystkie miary jakości powyżej 99%) niezależnie od liczby klas.

Wyniki uzyskane w rozprawie bardzo dobrze prezentują się w porównaniu z innymi wynikami, które można znaleźć w publikacjach międzynarodowych. Należy przy tym zaznaczyć, że porównanie to dotyczy różnych baz danych, stąd należy podejść do nich z dużą dozą ostrożności, przy czym wszystkie rozwiązania bazują na sygnałach czujników umieszczonych na korpusie urządzenia.

W pracy [64] przedstawiono rozwiązanie systemu do wykrywania anomalii łożysk na bazie Paderborn [65]. Porównano 2 rodzaje systemów. Jeden wykorzystywał sieci głębokie (1D oraz 2D), drugi sieci SVM. Najlepszy deklarowany wynik to dokładność 98.58% uzyskana dla sieci głębokich.

Praca [66] poświęcona jest wykrywaniu trzech rodzajów uszkodzeń łożyska: bieżnia zewnętrzna i wewnętrzna oraz uszkodzenie kulki. Dokładność wykrycia uszkodzeń na bazie „*Case Western Reserve University bearing dataset*” (CWRU) [14] deklarowana przez autorów zmieniała się od 94% do 99% w zależności od szybkości próbkowania.

W pracy [67] przedstawiono system wykrywania anomalii łożyska w układzie napędowym turbiny na podstawie sygnałów generowanych przez zestaw wielu sensorów i przy wykorzystaniu sieci głębokich CNN. Autorzy deklarują jedynie wartość pola AUC pod krzywą ROC zmieniająca się od 0.926 do 0.994 w zależności od wariantu rozwiązania.

Deklarowane wyniki rozpoznania anomalii działania łożysk z bazy danych CWRU w pracy [57] przy zastosowaniu sieci głębokich CNN i autoenkodera wahają się od 88% do 99% dla różnych rodzajów uszkodzeń.

Praca [68] przedstawia zastosowanie sieci 1D CNN w rozpoznaniu 3 typów uszkodzeń łożyska wykorzystując bazę CWRU. Wyniki deklarowane przez autorów obejmują dokładność 93.22%, czułość średnią powyżej 99% i precyzję rozpoznania klas zmieniającą się od 88% do ponad 99%.

W pracy [69] rozważano problem wykrycia anomalii polegającej na przegrzaniu łożysk generatorów przy zastosowaniu takich klasyfikatorów jak SVM, MLP oraz głębokiej sieci typu „*deep believe*”. Błąd prognozy uzyskany na danych pochodzących z 11 turbogeneratorów był rzędu 6%.

Wyniki autora prezentowane w tej rozprawie należą do grupy najlepszych, choć należy podchodzić do nich z dużą ostrożnością, ze względu na różne bazy danych uczestniczące w badaniach.

## 5 Wykrywanie anomalii typu „*deep fake*” w obrazach

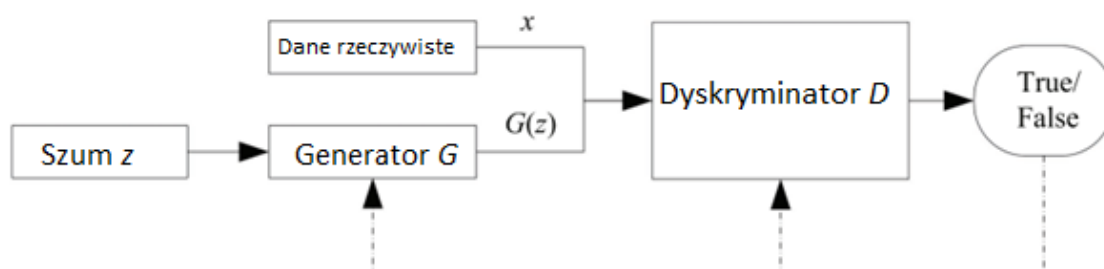
### 5.1 Definicja problemu *deep fake*

Ostatnie zadanie rozważane w pracy dotyczyć będzie wykrywania anomalii w obrazach, zwanych powszechnie „*deep fake*”. Pod tą angielską nazwą kryje się tworzenie fałszywych obrazów i dźwięków przy użyciu technik z zakresu sztucznej inteligencji, zapoczątkowane w roku 2017. Należy zauważyć, że w cyfrowych mediach manipulacja obrazami stała się wszechobecna. Jest to jedno z krytycznych zagadnień w coraz bardziej zdigitalizowanych społecznościach. Materiały tworzone za pomocą technologii *deep fake* mogą mieć różny charakter, w tym edukacyjny, rozrywkowy, dezinformacyjny czy dyskredytacyjny. Dwa ostatnie działania są szczególnie niebezpieczne, gdyż mogą być wykorzystywane do działań kontrowersyjnych czy przestępczych. W tworzeniu obrazów *deep fake* istotną rolę odgrywa twarz człowieka, która stanowić może istotny element w zakresie manipulacji. Wynika to z faktu, że rekonstrukcja i śledzenie twarzy jest szeroko stosowane i wykorzystywane w praktyce codziennej [70].

Zagrożenia powstałe w wyniku rozwoju technik *deep fake* spowodowały powstanie zespołów i grup badawczych, których celem jest stworzenie narzędzi umożliwiających wykrywanie filmów *deep fake*, walka z fałszywymi treściami jak również działania edukacyjne użytkowników Internetu. Początek takich działań dała grupa DFDC (*Deep Fake Detection Challenge*), zawiązana w roku 2019 przez wysokiej klasy specjalistów z Facebook, Microsoft, Amazon, Massachusetts Institute of Technology, Uniwersytetu Oksfordzkiego oraz Uniwersytetów z Berkeley i Maryland [71]. Algorytmy stworzone w roku 2020 pozwoliły uzyskać skuteczność wykrywania takich podróbek video w bazie *Faceforensics++* [72] na poziomie 65%. Należy przy tym zaznaczyć, że deklarowana efektywność wykrywania jest w dużej mierze uzależniona od bazy danych użytej w badaniach, sposobu kompresji i rodzaju wykrywanych obiektów poddanych operacji *deep fake*. W chwili obecnej prace są prowadzone przez wiele ośrodków naukowych, pozwalając istotnie zwiększyć efektywność wykrywania treści typu *deep fake* [73, 74, 75, 76].

Do tworzenia obrazów *deep fake* wykorzystywane są modele sieci głębokich, przy czym efekty ich działań są bardzo trudne do odróżnienia przez człowieka. W odpowiedzi na coraz powszechniejsze generatory tworzące obrazy *deep fake* podjęto szereg działań mających na celu stworzenie narzędzi do ich wykrywania. Największe instytucje i firmy niezależnie publikują bazy danych do badań nad tym zagadnieniem. Wykrywanie ich to problem klasyfikacji binarnej gdzie dany obraz (także video) powinien zostać sklasyfikowany jako prawdziwy lub fałszywy [77].

Najczęściej stosowane do generowania obrazów typu *deep fake* w praktyce są sieci GAN (ang. *Generative Adversarial Network*) [4]. Ich główne założenie pochodzi z teorii gier o profilu strategii równowagi Nasha (gra o sumie zerowej). Zakłada dwóch uczestników gry złożonych z generatora i dyskryminatora. Celem generatora jest odtworzenie rozkładu danych oryginalnych przy wykorzystaniu na starcie wektora losowego, podczas gdy dyskryminator stara się prawidłowo określić czy dane podane na jego wejście są prawdziwe czy też stworzone przez generator. W procesie uczenia uczestnicy (generator i dyskryminator) stale optymalizują swoje parametry z wykorzystaniem metody propagacji wstecznej błędu.



Rys. 5.1 Podstawowa struktura sieci GAN.

Struktura sieci GAN jest przedstawiona na rysunku 5.1. Funkcje **D** i **G** reprezentują dyskryminator i generator, ich wejściami są dane rzeczywiste (tu obrazy) i wektor losowy **z**. W procesie uczenia obraz utworzony przez generator **G(z)** stara się mieć taki sam rozkład statystyczny jak rzeczywiste dane wejściowe **x**. Jeśli na wejście dyskryminatora podane zostaną dane oryginalne, powinien on sklasyfikować je jako prawdziwe, w przypadku danych **G(z)** wygenerowanych sztucznie - jako fałszywe.

Proces uczenia GAN polega na maksymalizacji prawdopodobieństwa, że dyskryminator popełni błąd przyjmując, że obie grupy danych (oryginalne i sztucznie wygenerowane) są identyczne pod względem rozkładu statystycznego (maksymalizacja prawdopodobieństwa zaliczenia obu obrazów: oryginalnego i wygenerowanego sztucznie do tej samej klasy). Podstawą procesu jest uczenie pośrednie generatora poprzez dyskryminator, który ocenia jak realistyczny jest wzorzec wejściowy w stosunku do oryginalnego. Dane **G(z)** wytworzone przez generator podlegają dynamicznej adaptacji, przy czym jej celem nie jest minimalizacja odległości od wzorca oryginalnego, ale oszukanie dyskryminatora. Proces uczenia systemu można zapisać w postaci programowania minimaxowego zdefiniowanego w postaci [78]:

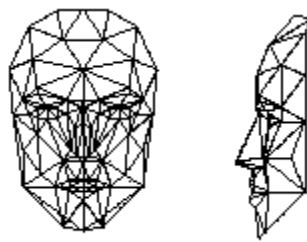
$$\min_G \max_D V(G, D) = E_{\mathbf{x} \sim p_x} [\log D(\mathbf{X})] + E_{\mathbf{z} \sim p_z} [\log(1 - D(G(\mathbf{z})))] \quad (5.1)$$

gdzie symbol  $\mathbf{E}$  reprezentuje wartość oczekiwaną, natomiast  $\mathbf{p}_x$  i  $\mathbf{p}_z$  rozkłady statystyczne odpowiednio danych rzeczywistych (oryginalnych) i wygenerowanych sztucznie.  $D(\mathbf{X})$  oraz  $G(\mathbf{X})$  są wartościami generowanymi odpowiednio przez dyskryminator  $\mathbf{D}$  i generator  $\mathbf{G}$  dla danych  $\mathbf{X}$ . W efekcie proces optymalizacyjny ukierunkowany jest na poszukiwanie minimum względem generatora  $\mathbf{G}$  i maksimum względem dyskryminatora  $\mathbf{D}$ .

## 5.2 Baza danych użytych w eksperymentach

Baza danych użyta w eksperymentach pochodziła ze zbioru *Faceforensics++* zatytuowanego „*Learning to Detect Manipulated Facial Images*” [79]. Zawiera ona 1000 oryginalnych sekwencji wideo, które zostały poddane różnym sposobom generowania sztucznych twarzy, w tym *FaceSwap*, *DeepFake* (algorytm *FakeApp*) oraz *Face2Face*. Wszystkie filmy pochodzą z serwisu Youtube i zawierają frontalnie ustawione profile twarzy. Taka konfiguracja umożliwia łatwiejsze z punktu widzenia algorytmu, generowanie realistycznie wyglądających sztucznych twarzy. W pracy badaniom poddane zostaną trzy zestawy danych z bazy *Faceforensics++* wygenerowane przy użyciu wzmiankowanych wyżej technik tworzenia podróbek [80].

*FaceSwap* to metoda oparta o graficzny transfer obszaru wykrytego jako twarz z wideo wejściowego do wideo wyjściowego [81, 82, 83]. Kopiowanie twarzy polega na wykryciu punktów charakterystycznych (*landmarks*) i dostosowaniu wybranego w ten sposób obszaru w postaci wielokąta do modelu 3D „*Candidie*”.



Rys. 5.2 Przykład tworzenia graficznych punktów charakterystycznych twarzy w metodzie *FaceSwap* [82].

Modelem jest sparametryzowana maska odwzorowująca kształt twarzy i składająca się z około 100 wielokątów opisanych numerycznie (rys. 5.2). Następnie model jest transferowany wstecznie do obrazu wejściowego a celem jest minimalizacja różnicy pomiędzy wygenerowanym w ten sposób kształtem twarzy a zlokalizowanym wcześniej obszarem ograniczonym punktami charakterystycznymi. Ostatecznie, stworzony model jest łączony z obrazem wejściowym, a kolory



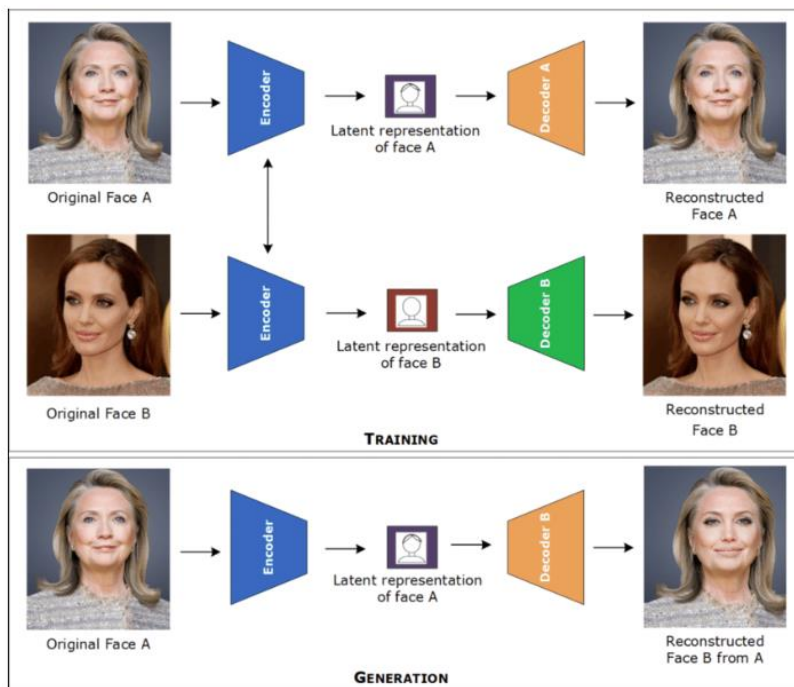
są korygowane. Ten proces powtarza się dla każdej pary obrazów wejściowych w sekwencji wideo. Implementacja tego algorytmu jest efektywna nawet przy użyciu tylko jednostki CPU bez akceleracji graficznej. Na rys. 5.3 przedstawiono przykłady obrazów *deep fake* (wiersz dolny) z bazy *Faceforensics++* utworzonych z obrazu oryginalnego (wiersz górny) metodą *FaceSwap* [82].



Rys. 5.3 Przykłady obrazów *deep fake* (wiersz dolny) z bazy *Faceforensics++* utworzonych z obrazu oryginalnego (wiersz górny) metodą *FaceSwap*.

**DeepFake** jest synonimem powszechnie używanym w tworzeniu obrazów zmodyfikowanych przez sztuczną inteligencję jak również szczególną metodą manipulacji obrazami. Istnieje wiele ich implementacji, najbardziej znaną jest *FakeApp* [84], której wyniki będą podlegały badaniom eksperymentalnym w ramach rozprawy. Metoda wykorzystuje technikę auto-enkodera i dekodera w procesie tworzenia podróbki obrazu. Para enkoder-dekoder jest trenowana w procesie uczenia obrazu A i B dla odtworzenia cech charakterystycznych obu obrazów. W procesie generacji właściwej podróbki obrazu decoder A jest zamieniany z dekoderek B. W ten sposób cechy charakterystyczne obrazu A przenoszone są na obraz B. Wynik działania dekodera jest łączony

z pozostałą częścią obrazu przy użyciu edycji Poissona [84] i zastosowaniu ogólnych algorytmów interpolacji w edycji elementów obrazów. Przykład działania tej metody pokazany jest na rys. 5.4 [85].



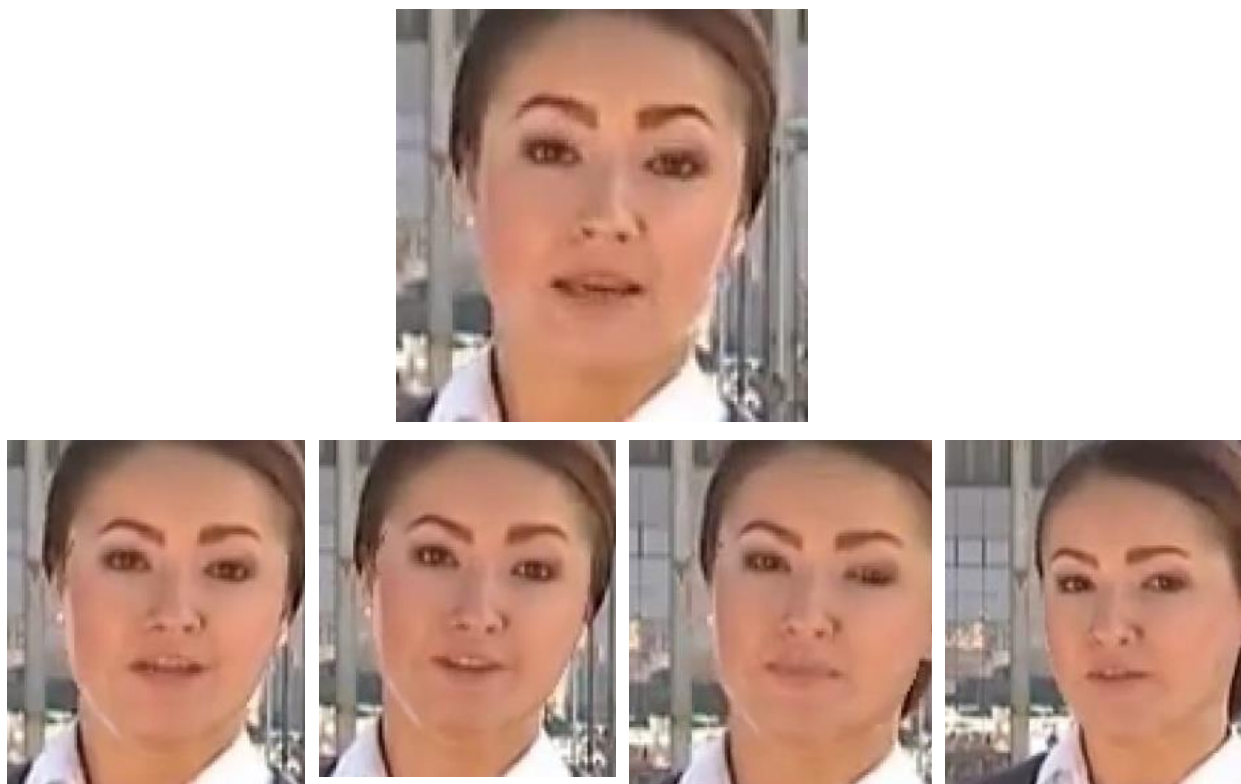
Rys. 5.4 Ilustracja sposobu tworzenia fałszywego obrazu metodą DeepFake (algorytm FakeApp). W procesie uczenia auto-enkoderów uczestniczą dwa obrazy A i B. Przy generacji podróbki obrazu B właściwy dekodery B jest zastąpiony poprzez dekodery A, przenosząc w ten sposób cechy A na obraz B [85].

Na rys. 5.5 przedstawiono typowe przykłady fałszywych twarzy z bazy danych [72] wygenerowane tą metodą.



Rys. 5.5 Przykłady obrazów *deep fake* (wiersz dolny) z bazy *Faceforensics++* utworzonych z obrazu oryginalnego (wiersz górny) metodą *DeepFake* (algorytm *FakeApp*).

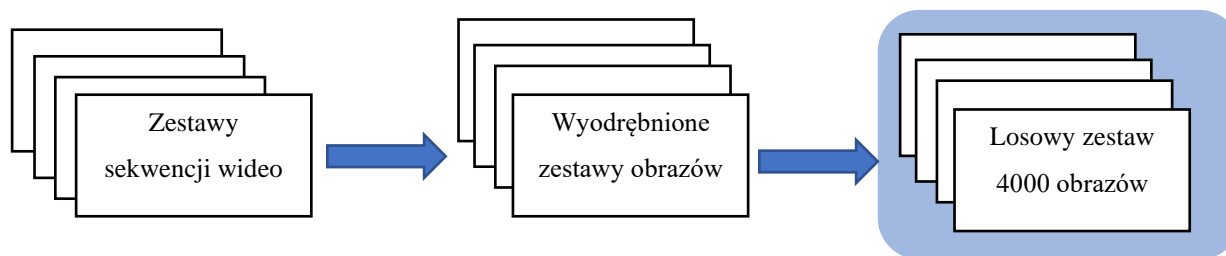
**Face2Face** to system rekonstrukcji twarzy, który przenosi ekspresję postaci z wideo źródłowego na wideo docelowe, zachowując przy tym tożsamość osoby docelowej [86]. Oryginalna implementacja opiera się na dwóch strumieniach wejściowych wideo, z ręcznym wyborem klatek kluczowych. Klatki te są używane do generowania jak najdokładniejszej rekonstrukcji twarzy, która może być użyta do ponownej syntezy. W przetwarzaniu wykorzystuje się pierwsze klatki w celu uzyskania tymczasowej tożsamości twarzy (modelu 3D), a następnie śledzi się ekspresję na pozostałych klatkach. Wygenerowane sekwencje wideo powstają poprzez przeniesienie źródłowych parametrów ekspresji każdej klatki (współczynników *Blendshape*) do docelowego wideo. Na rys. 5.6 przedstawiono przykłady obrazów *deep fake* (wiersz dolny) utworzonych z obrazu oryginalnego (wiersz górny) metodą *Face2Face*.



Rys. 5.6 Przykłady obrazów deep fake (wiersz dolny) z bazy Faceforensics++ utworzonych z obrazu oryginalnego (wiersz górny) metodą Face2Face.

Baza *Faceforensics++* [72] zawiera 4000 sekwencji wideo, w tym 1000 filmów oryginalnych oraz po 1000 filmów zmodyfikowanych każdą z wyżej opisanych metod. Pozwala to uzyskać w sumie 509 914 obrazów (klatek filmowych). Wszystkie pliki w bazie zostały skompresowane kodekiem H.264 (MPEG-4 AVC) szeroko używanym w Internecie w mediach społecznościowych i informacyjnych. W badaniach tej rozprawy zastosowano średni poziom kompresji, oznaczony jako c23. Obraz skompresowany w ten sposób wizualnie nie odbiega zasadniczo od oryginalnego (surowego). Spośród dostępnych obrazów (sekwencji wideo) do badań w rozprawie wybrano po 1000 obrazów reprezentujących każdą kategorię.

Biorąc pod uwagę, że dostępna baza jest w postaci sekwencji wideo, niezbędny jest etap wstępny polegający na ekstrakcji pojedynczych obrazów w postaci klatki z nagrań. W tym celu z każdego filmu i każdej kategorii (oryginalnej i trzech zmodyfikowanych) wybrano losowo obrazy występujące po sobie w okresie jednej sekundy. Następnie wybrane zostały także losowo zestawy danych, po jednym obrazie z każdej sekwencji tworząc 4 zestawy obrazów po 4000 elementów każdy.



Rys. 5.7 Ilustracja zastosowanego sposobu wyboru obrazów z bazy Faceforensics++ w tworzeniu zestawów obrazów do eksperymentów numerycznych.

W efekcie w każdym zestawie 4000 obrazów (klatek) występowały twarze oryginalne z ich podróbkami. Liczba oryginalnych twarzy w każdym zestawie była taka sama. Każdej oryginalnej twarzy towarzyszyły 3 rodzaje podróbek, każda wygenerowana losowo przy użyciu innej metody: **FaceSwap, DeepFake (FakeApp) oraz Face2Face.**

### 5.3 Detekcja obrazu twarzy z klatki video

Stworzone zestawy danych reprezentują klatki video jak na rys 5.8. W większości klatek twarz stanowi niewielką część obrazu. Niezbędnym etapem jest więc detekcja i ekstrakcja obrazu jedynie interesującej nas twarzy z klatki filmowej [87].



Rys. 5.8 Przykłady klatek video z bazy Faceforensics++.

W tym celu zastosowano metodę HoG (ang. *Histogram of Oriented Gradients*) [87 88 89]. Metoda określa deskryptory numeryczne lokalizujące wystąpienie orientacji gradientu w zlokalizowanym fragmencie obrazu. Deskryptor HOG skupia się na strukturze lub kształcie obrazu. Obraz jest dzielony na małe połączone rejony zwane komórkami. Dla tych rejonów obrazu generowane są histogramy wartości i orientacji gradientu poszczególnych pikseli. Deskryptory powstają jako złączenie (ang. *concatenation*) tych histogramów. Zwykle łączy się komórki w większe obszary zwane blokami tworząc deskryptory blokowe. W ogólności procedurę HoG [89] zastosowaną w pracy można przedstawić w postaci:

1. Wczytanie obrazu w skali stopnia szarości.
2. Obliczenie gradientu  $G$  obrazu. Gradient tworzy się przez połączenie wartości i kąta (kierunku) zmian stopnia szarości pikseli w obrazie. Wektory gradientu  $G$  w osi  $x$  i  $y$  są obliczane dla każdego piksela  $(w,k)$  obrazu  $I$ :

$$G_x(w, k) = I(w, k + 1) - I(w, k - 1) \quad (5.1)$$

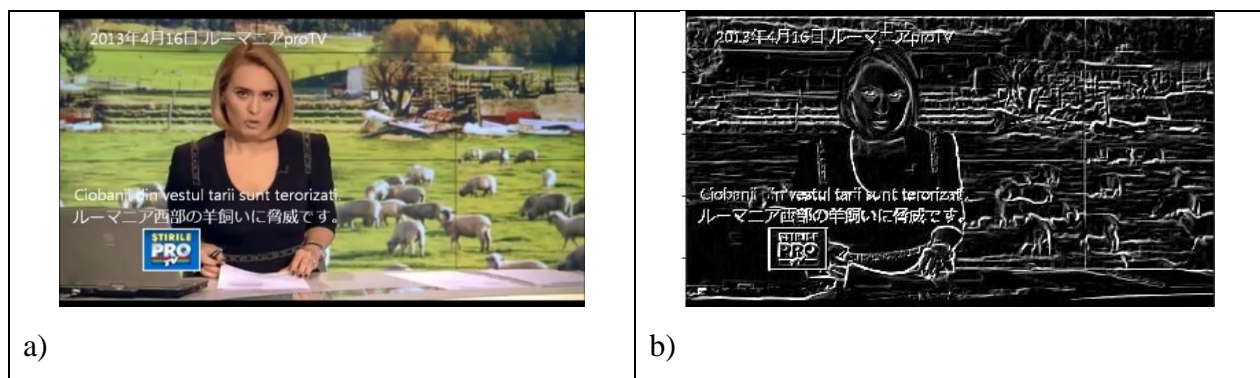
$$G_y(w, k) = I(w + 1, k) - I(w - 1, k) \quad (5.2)$$

Następnie wyznaczana jest wartość amplitudy  $\mu$  i kąta  $\theta$  gradientu dla każdego piksela

$$\mu = \sqrt{G_x^2 + G_y^2} \quad (5.3)$$

$$\theta = \arctg\left(\frac{G_y}{G_x}\right) \quad (5.4)$$

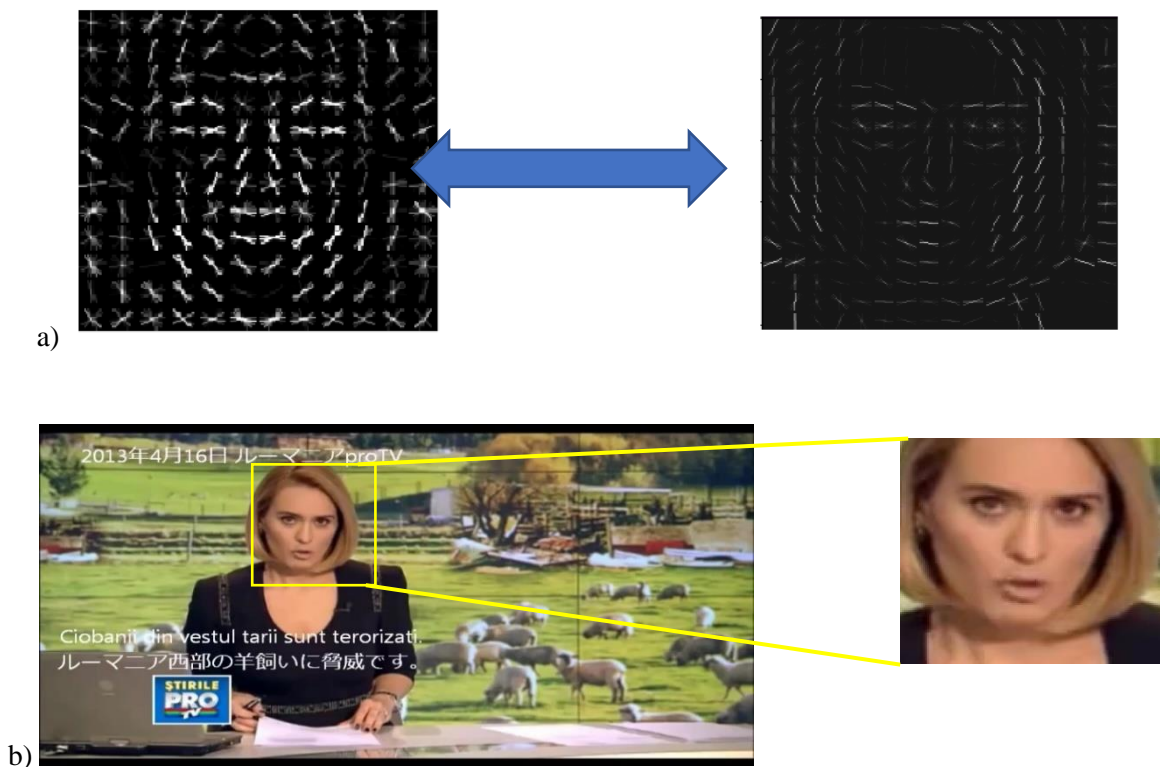
Na rys. 5.8 przedstawiono przykładowy obraz wejściowy z klatki video (a) oraz jego wizualizację w postaci gradientowej (b).



Rys. 5.8 Przykładowy obraz wejściowy z klatki video (a) i wizualizacja wartości jego gradientów (b).

3. Obraz reprezentowany przez wartości amplitudy i kąta gradientu jest dzielony na komórki o rozmiarze  $8 \times 8$  pikseli tworząc strukturę blokową. Dla każdego bloku wyznaczany jest histogram o liczbie przedziałów równej liczbie komórek tworzących blok.
4. Bloki histogramu są łączone w nowe bloki o rozmiarze  $2 \times 2$  z przesunięciem jednego pierwotnego bloku (nakłada się 8 pikseli). Wartości gradientu nowo powstałych bloków są normalizowane przy użyciu normy  $L_2$  aby zniwelować wpływ zmiany kontrastu między sąsiednimi komórkami.
5. Ostatecznie wynikowe wycinki obrazu (macierz cech gradientowych) są podawane na wejście klasyfikatora SVM dla porównania z zaimplementowanymi programowo wzorcami twarzy. Wyznaczone współrzędne obrazu oryginalnego są przenoszone na obraz

wejściowy, a z niego wyodrębniana jest sama twarz, która będzie podlegać zapisowi w bazie danych. Rys. 5.9 przedstawia końcowy etap procedury HoG. Wiersz górny przedstawia zaimplementowany gradientowy wzorzec twarzy (po lewej) oraz porównywana macierz cech gradientowych (po prawej) wygenerowana przez procedurę HoG. Wiersz dolny na rysunku prezentuje obraz samej twarzy wyekstrahowany z aktualnej [90]klatki filmowej zilustrowanej z lewej strony.



Rys. 5.9 Ilustracja końcowych etapów procedury HoG dla ekstrakcji twarzy z klatki video: a) przetworzenie wektora gradientów przez sieć SVM, b) Wynik końcowy ekstrakcji twarzy z klatki video [91].



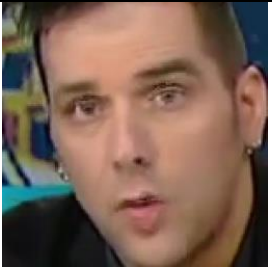




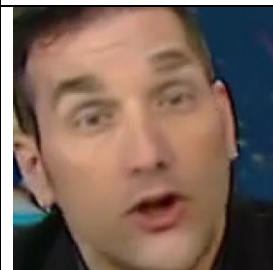
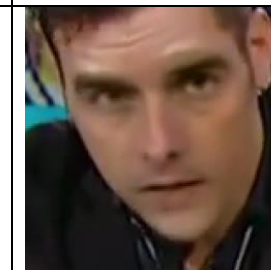
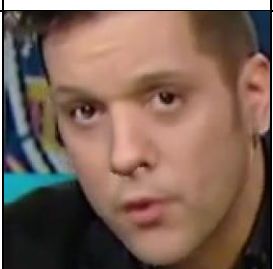
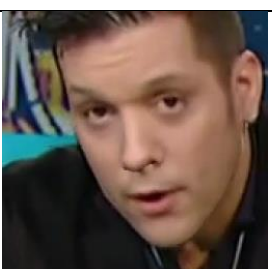
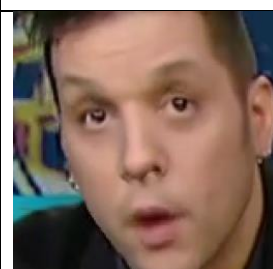
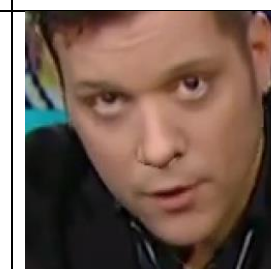
Stworzona w ten sposób baza obrazów twarzy, użyta w eksperymentach numerycznych, zawierała obrazy twarzy różnych osób wyekstrahowane z filmów video, dla których zostały stworzone podróbki przy użyciu trzech wzmiankowanych metod fałszowania. Poniżej na rys. 5.10 przedstawiono 10 przykładowych obrazów oryginalnych twarzy różnych osób zaczerpniętych z tego stworzonego zestawu danych, dla których przeprowadzono wykrywania podróbek obrazów.



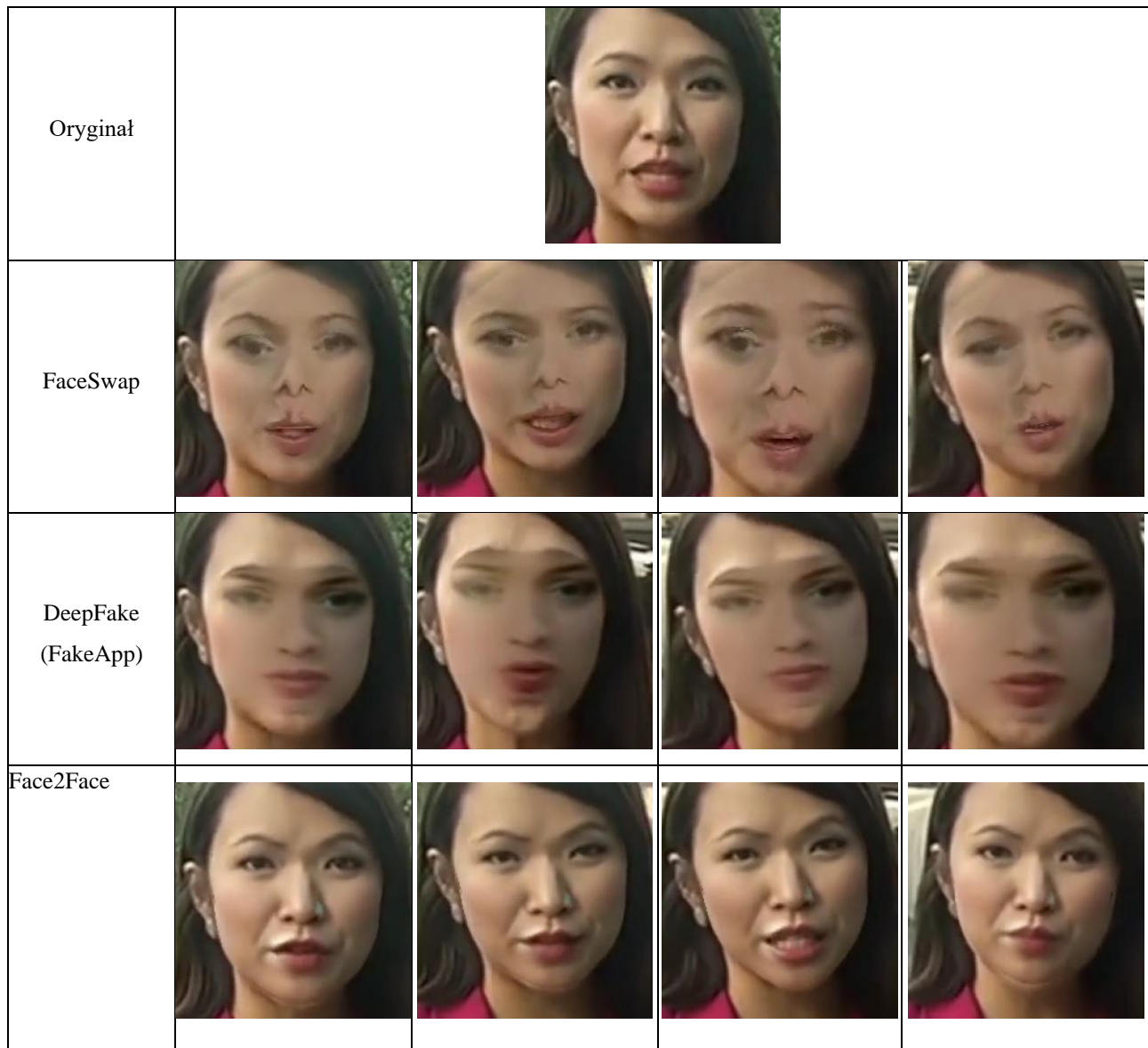
Rys. 5.10 Przykładowe obrazy z bazy danych użyte w eksperymentach numerycznych.

Na rysunkach 5.11 i 5.12 przedstawiono kilka przykładów obrazów twarzy deep fake utworzonych sztucznie przy użyciu trzech wymienionych metod fałszowania (*FaceSwap*, *Deepfake* (algorytm *FakeApp*) i *Face2Face*) dla dwu przykładów oryginału z rys. 5.10. Podróbki obrazów są zamieszczone w trzech wierszach dolnych i dotyczą obrazu oryginalnego prezentowanego w wierszu górnym. Dane pochodzą z bazy *Faceforensics++*. Szczególnie zestaw 1 obrazów może stanowić problem w rozpoznaniu podróbki od oryginału ze względu na niewielkie zmiany wprowadzone przez algorytmy fałszowania.



Oryginał				
FaceSwap				
Deepfake (FakeApp)				
Face2Face				

Rys. 5.11 Przykłady obrazów deep fake utworzonych z obrazu oryginalnego prezentowanego w pierwszym wierszu (zestaw 1).



Rys. 5.12 Przykłady obrazów *deep fake* utworzonych z obrazu oryginalnego prezentowanego w pierwszym wierszu (zestaw 2).

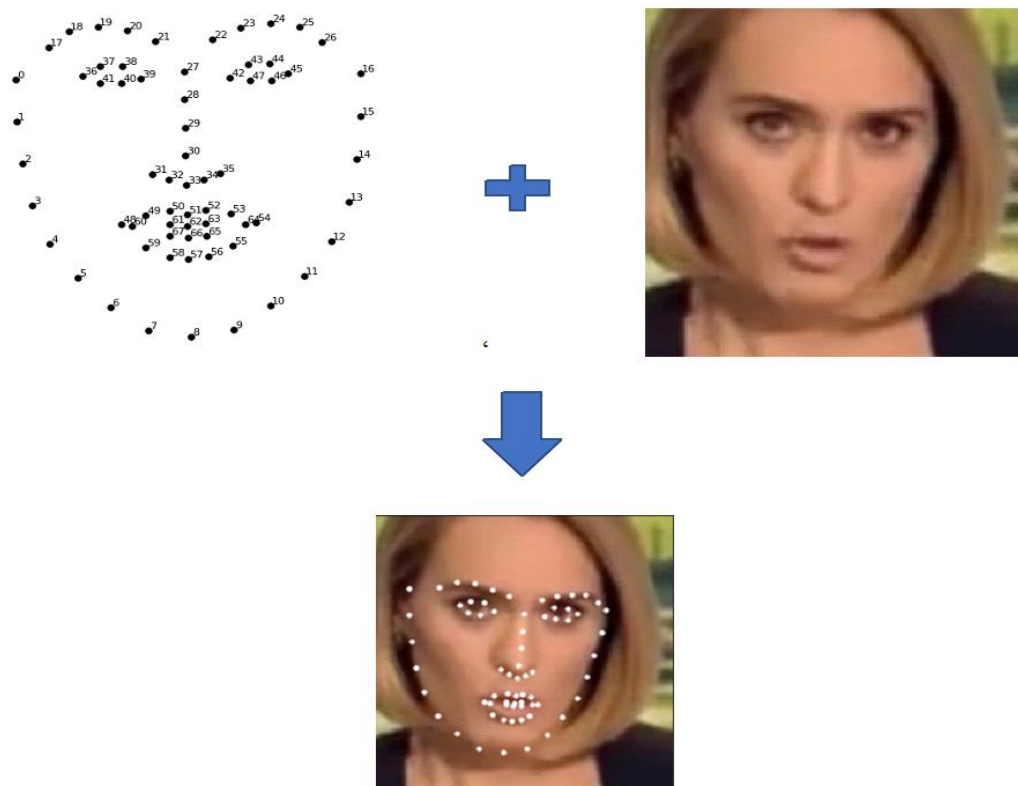
#### 5.4 Proponowana procedura wykrywania obrazów twarzy typu *deep fake*

W wyniku wstępnego przetwarzania klatek video zostały utworzone zbiory obrazów twarzy reprezentujące oryginały oraz trzy rodzaje podróbek bazujące na procedurach *FaceSwap*, *Deepfake* (algorytm *FakeApp*) oraz *Face2Face*. Stanowią one bazę danych poddanych eksperymentom numerycznym przy użyciu procedury wykorzystującej transformację falkową i zespół głębokich sieci CNN jako klasyfikatorów.

Na proponowaną procedurę wykrywania obrazów typu *deep fake* składają się trzy podstawowe etapy przetwarzania:

- 1) Wykrywanie punktów orientacyjnych twarzy (ang. *face landmarks*) w obrazie.
- 2) Zastosowanie ciągłej transformacji falkowej obrazu z nałożeniem punktów orientacyjnych na wynik transformacji CWT.
- 3) Zastosowanie obrazów wygenerowanych poprzez CWT jako atrybutów wejściowych dla zespołu głębokich klasyfikatorów CNN odpowiedzialnych za wykrycie podróbek obrazów.





Pierwszym pomocniczym elementem przetwarzania jest wykrycie punktów orientacyjnych twarzy pozwalające na lokalizację podstawowych elementów struktury twarzy, istotnych w predykcji kształtu twarzy. Algorytm wykorzystuje uprzednio wyznaczone macierze HoG wewnątrz biblioteki dlib [92]. Do reprezentacji struktury twarzy użyto 68 punktów reprezentowanych przez współrzędne (x, y) obrazu. Rys 5.13 przedstawia przykładowo graficzną wizualizację punktów orientacyjnych oraz wynik operacji nałożenia ich na obraz wejściowy twarzy. Białe punkty na rysunku są celowo powiększone dla lepszej wizualizacji.



Rys. 5.13 Graficzna wizualizacja punktów orientacyjnych twarzy oraz wynik ich nałożenia na obraz wejściowy.

Następnym etapem procedury jest zastosowanie transformacji falkowej CWT w stosunku do obrazów z bazy danych. Wyniki tej transformacji są zespolone i będą dalej reprezentowane poprzez obrazy amplitud oraz kątów (jeden obraz wejściowy jest w wyniku CWT transformowany na dwa obrazy wynikowe). Transformacja CWT została przeprowadzona dla różnych wartości skali (od 1 do 12), przy czym w ostatecznym rozwiązaniu wystarczające okazało się ograniczenia skali do wartości 6. Zastosowanie wyższych wartości skali nie prowadziło do poprawy a raczej pogarszało wyniki rozpoznania.

Przykładowe zobrazowanie wyników transformacji CWT (obrazy amplitud i kątów) uzyskane dla obrazu oryginalnego przy różnych wartościach zastosowanej skali dekompozycji (1, 2, 4, 6) i użyciu falki Morleta przedstawia rys. 5.14.

Oryginał			
	Obraz amplitudowy	Obraz kątowy	Obraz amplitudowy z nałożonymi punktami orientacyjnymi
Skala 1			

Rys. 5.14 Przykładowe wyniki transformacji CWT dla czterech wartości skali dekompozycji CWT. Wiersz górny reprezentuje obraz oryginalny, pozostałe wiersze przedstawiają wyniki CWT dla skali zmieniającej się kolejno 1, 2, 4 i 6.



Rys. 5.14 Przykładowe wyniki transformacji CWT dla czterech wartości skali dekompozycji CWT. Wiersz górny reprezentuje obraz oryginalny, pozostałe wiersze przedstawiają wyniki CWT dla skali zmieniającej się kolejno 1, 2, 4 i 6.

W zastosowanej transformacji CWT analizowany obraz w pierwszej kolejności jest przeskalowany do rozmiaru 256x256 pikseli i przekształcany do skali szarości. Tak zmodyfikowany obraz podlega przekształceniu CWT w odpowiedniej skali. W jej wyniku dla każdej skali przekształcenia otrzymywane są 2 obrazy wynikowe (amplitudowy i kątowy) o wymiarach 256x256. Oba są reprezentowane przez macierze liczbowe o tych samych wymiarach.

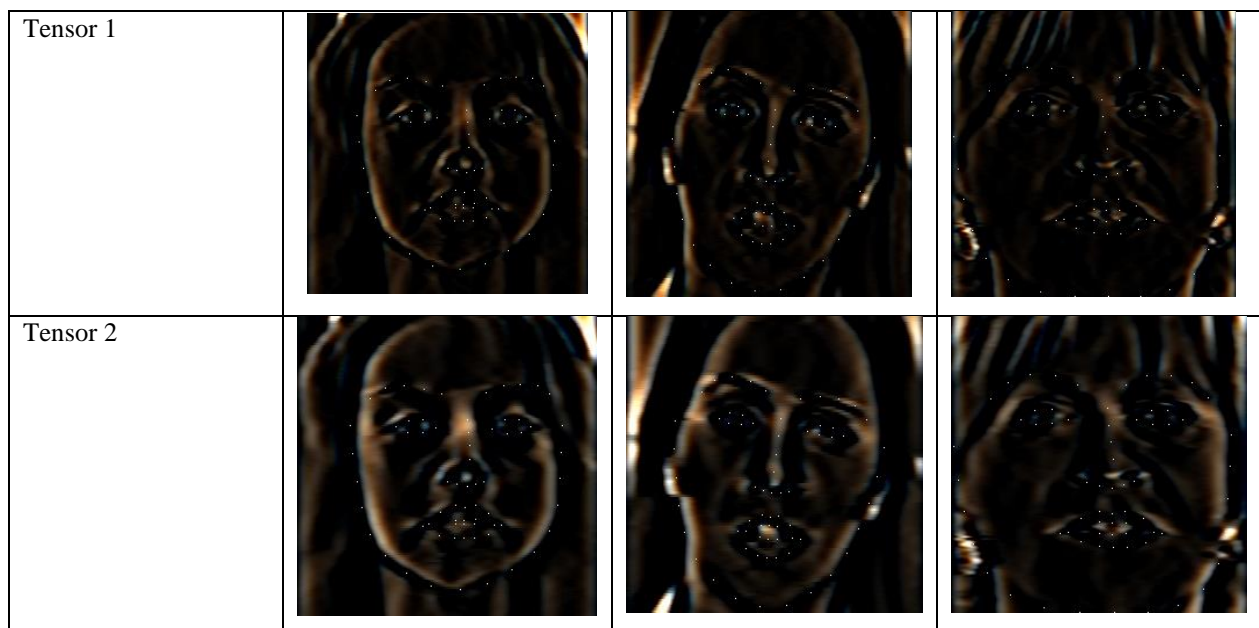
W ostatnim etapie procedury wykrywania podróbek obrazów zastosowano zespół pre-trenowanych sieci CNN z wykorzystaniem techniki transfer learningu. Zespół złożony jest z 8 pre-trenowanych struktur CNN (*alexnet*, *mobilenetv2*, *resnet50*, *efficientnet*, *squeezenet*, *googlenet*, *shufflenet* oraz *inceptionresnetv2*), identycznie jak w rozwiązaniach z poprzednich rozdziałów.

Zastosowane sieci pre-trenowane pozwalają na użycie na wejściu tensora o głębokości 3 (zwykle są to reprezentacje RGB). Możliwe jest zatem przystosowanie takich sieci do sytuacji powstałej w wyniku transformacji CWT, generującej tylko dwa obrazy wynikowe. Przetestowano trzy rodzaje rozwiązań:

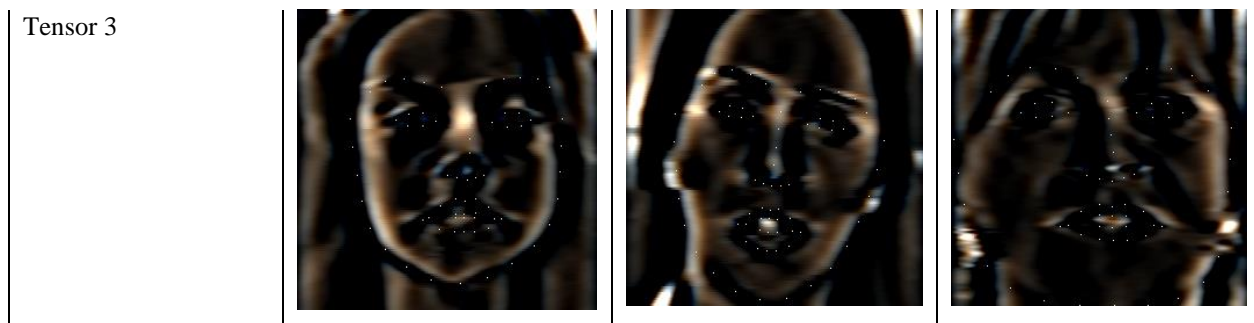
- syntetyczne złożenie 3 takich samych obrazów wynikowych CWT (modułów lub kątowych) jako tensora wejściowego
- dodanie do dwu obrazów (moduł i kąt) dodatkowego obrazu reprezentowanego przez macierze zerowe – pozostawienie 2 obrazów wynikowych CWT na wejściu sieci bez zmian
- złożenie kolejnych 3 obrazów (macierzy) wynikowych CWT w kolejnych skalach jako tensor wejściowy dla sieci. Może to dotyczyć zarówno obrazów amplitudowych lub kątowych.

W wyniku przeprowadzonych eksperymentów wstępnych okazało się, że trzeci sposób jest najbardziej skuteczny. Można w ten sposób stworzyć wiele różnych zestawów tensorów 3x256x256 wejściowych dla sieci. Stworzono i przebadano wstępnie wiele różnych kombinacji, z których najlepsze wyniki uzyskano dla trzech kombinacji

- Tensor 1 – wyniki CWT skala 1 + skala 2 + skala 3
- Tensor 2 – wyniki CWT skala 3 + skala 4 + skala 5
- Tensor 3 – wyniki CWT skala 5 + skala 6 + skala 7



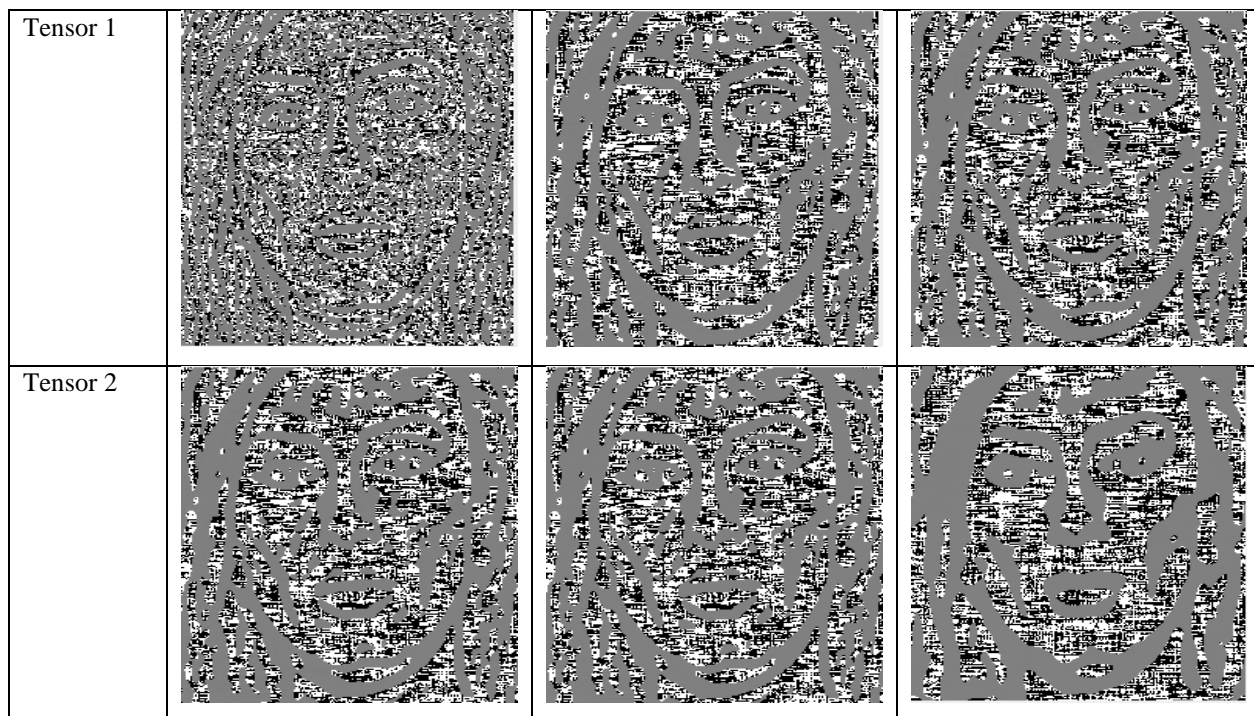
Rys. 5.15 Przykładowe obrazy tworzące tensory wejściowe amplitudowe dla pre-trenowanych sieci CNN uzyskane przy zastosowaniu falki Moreleta.



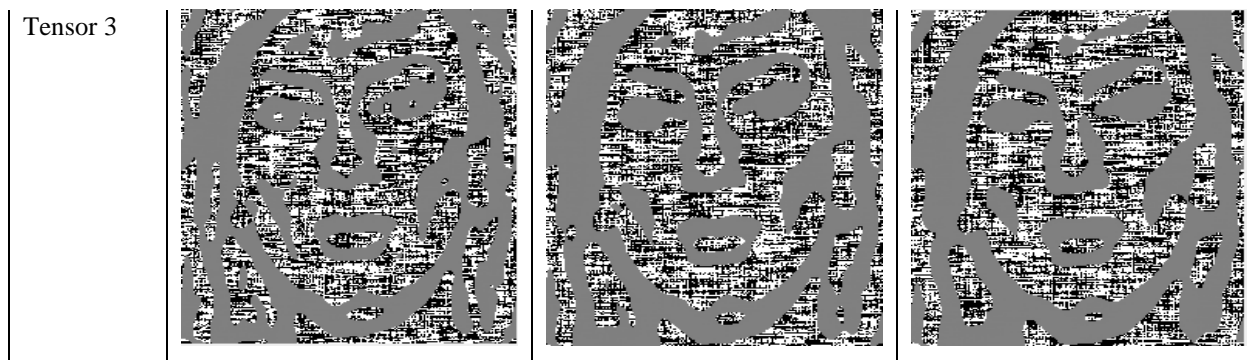
Rys. 5.15 Przykładowe obrazy tworzące tensory wejściowe amplitudowe dla pre-trenowanych sieci CNN uzyskane przy zastosowaniu falki Moreleta.

Przykładowe obrazy dotyczące zestawów trzech wejściowych tensorów amplitudowych utworzonych w ten sposób przedstawia rys. 5.15.

Przykładowe obrazy dotyczące zestawów trzech wejściowych tensorów utworzonych z obrazów amplitudowych przedstawiono na rys. 5.15, a dla obrazów utworzonych poprzez reprezentacje kątowe na rys. 5.16.



Rys. 5.16 Przykładowe obrazy tworzące tensory wejściowe kątowe dla pre-trenowanych sieci CNN uzyskane przy zastosowaniu falki Morleta.



Rys. 5.16 Przykładowe obrazy tworzące tensory wejściowe kątowe dla pre-trenowanych sieci CNN uzyskane przy zastosowaniu falki Morleta.

## 5.5 Wyniki eksperymentów numerycznych

Ekspertymy numeryczne dotyczące wykrywania obrazów *deep fake* przeprowadzono na wstępnie losowo wyselekcjonowanym zestawie obrazów wziętych z bazy *Forensics++* (procedury przedstawione w poprzednich punktach rozdziału). Zgromadzony zestaw obrazów zawierał po 2000 obrazów (1000 oryginalnych i 1000 podróbek) dla każdej badanej osoby wykonanych przy użyciu trzech różnych algorytmów tworzenia obrazów *deep fake* (*FaceSwap*, *FakeApp* oraz *Face2Face*).

Jako klasyfikatory zastosowano wybrane w wyniku eksperymentów wstępnych osiem pre-trenowanych struktur CNN dostępnych w Internecie (*alexnet*, *mobilenetv2*, *resnet50*, *efficientnetb0*, *squeezenet*, *googlenet*, *shufflenet*, *inceptionresnetv2*) złączonych w zespół ekspertowy. Wynik działania zespołu jest ustalany poprzez głosowanie większościowe. Przy trzech metodach tworzenia danych wejściowych dla zespołu tworzy się zestaw 3 zespołów, każdy złożony z ośmiu sieci głębokich, ustalających wynik poprzez głosowanie.

Zbiór danych obrazów był dzielony na podzbiór uczący (70%) oraz testujący nie uczestniczący w procesie uczenia (pozostałe 30%). Ekspertymy były powtórzone 10 razy za każdym razem zmieniając losowo kolejne zestawy oryginalnych obrazów i ich podróbek dla każdej z trzech wymienionych metod ich tworzenia. Wyniki średnie zostaną przedstawione tylko dla danych testujących oddzielnie dla każdego z trzech sposobów tworzenia podróbek.



Tabela 5.1 Wyniki rozpoznania obrazów *deep fake* wytworzonych przy pomocy algorytmu *FakeApp* przy zastosowaniu trzech wartości skali w transformacji CWT (obrazy amplitudowe).

Tensor wejściowy	DeepFake (FakeApp)								
	1			2			3		
Parametr	ACC [%]	PREC [%]	REC [%]	ACC [%]	PREC [%]	REC [%]	ACC [%]	PREC [%]	REC [%]
Zespół	90.16	91.18	90.17	91.33	92.61	91.34	<b>97.33</b>	<b>97.47</b>	<b>97.33</b>
Średnia członków	78.78	86.42	86.98	87.11	85.65	79.25	94.39	85.50	94.67
Std członków	5.00	7.38	2.38	8.88	8.97	6.55	6.71	7.41	5.33
Najlepszy	87.43	90.16	88.82	90.22	91.09	87.17	96.79	95.10	96.73
Najgorszy	78.33	76.53	84.66	82.71	81.89	78.67	92.23	84.28	93.33

Tabele 5.1, 5.2 i 5.3 przedstawiają wartości podstawowych miar jakości rozpoznania obrazu *deep fake* od oryginalnego, w tym średnią dokładność klasową (ACC), średnią precyzję obu klas (PREC) oraz średnią czułość wykrycia obu klas (REC) dla obrazów *deep fake* utworzonych przy zastosowaniu metody *Deepfake* (algorytm *FakeApp*). Dodatkowo zamieszczono wyniki średnie klasyfikatorów (bez ich integracji), standardowe odchylenie wyników a także wynik najlepszego i najgorszego klasyfikatora. W tworzeniu tensorów wejściowych wykorzystano obrazy wynikowe CWT dotyczące amplitudy.

Tabela 5.2 Wyniki rozpoznania obrazów deep fake wytworzonych przy pomocy algorytmu FaceSwap przy zastosowaniu trzech wartości skali w transformacji CWT (obrazy amplitudowe).

Tensor wejściowy	FaceSwap								
	1			2			3		
Parametr	ACC [%]	PREC [%]	REC [%]	ACC [%]	PREC [%]	REC [%]	ACC [%]	PREC [%]	REC [%]
Zespół	84.12	84.19	82.04	87.50	8868	8894	<b>93.16</b>	<b>92.67</b>	<b>93.60</b>
Średnia członków	79.36	77.8	75.87	83.21	84.05	83.32	85.85	86.51	88.60
Std członków	4.11	5.94	5.91	3.48	2.54	3.02	5.99	6.03	5.19
Najlepszy	82.78	82.13	81.17	85.67	84.62	85.45	90.04	89.29	92.24
Najgorszy	77.02	74.26	71.96	80.22	80.03	81.62	81.07	82.30	77.93

Tabela 5.3 Wyniki rozpoznania obrazów deep fake wytworzonych przy pomocy algorytmu Face2Face przy zastosowaniu trzech wartości skali w transformacji CWT (obrazy amplitudowe).

Tensor wejściowy	Face2Face								
	1			2			3		
Parametr	ACC [%]	PREC [%]	REC [%]	ACC [%]	PREC [%]	REC [%]	ACC [%]	PREC [%]	REC [%]
Zespół	71.72	67.02	67.50	75.09	72.31	72.83	<b>83.50</b>	<b>82.68</b>	<b>81.15</b>
Średnia członków	64.56	66.14	62.89	68.33	68.42	69.62	78.25	79.05	78.73
Std członków	9.73	7.65	4.21	7.16	3.42	2.75	4.62	4.66	6.21
Najlepszy	69.34	66.40	66.98	73.61	70.59	70.63	82.07	81.58	80.54
Najgorszy	59.50	60.95	60.60	63.04	66.32	65.71	76.81	76.94	71.01

Najlepsze wyniki uzyskano przy użyciu systemu dla obrazów zmodyfikowanych algorytmem *Deepfake (FakeApp)*, najłabsze dla algorytmu *Face2Face*. Słabsze wyniki dla algorytmu *Face2Face* są spowodowane sposobem tworzenia podróbek obrazu. Nie zmienia on zasadniczo całej twarzy a jedynie jej niewielką część, np. usta. Skutkiem tego są bardzo małe rozbieżności pomiędzy oryginałem i modyfikacją obrazu, trudne to rozróżnienia przy porównywaniu

pojedynczych klasek. Dla wszystkich metod obserwuje się wzrost wartości miar jakości klasyfikatorów i ich zespołów przy wzroście wartości skali CWT uwzględnianej w obrazach tworzących tensory wejściowe. Wyniki podane w rozprawie ograniczono do trzech postaci tensorów, gdyż postać tensora trzeciego (tworzonego z wyników CWT dotyczących obrazów modułu skali 5, 6 i 7) okazała się najlepsza pod względem uzyskanych wskaźników jakości.

Tabela 5.4 Wyniki porównawcze rozpoznania obrazów deep fake reprezentowanych przez obrazy kątowe, wytworzonych przy pomocy trzech algorytmów (*FaceApp*, *FaceSwap*, *Face2Face*). Wyniki dotyczą najlepszej reprezentacji tensorowej (tensor nr 3). Dla uzyskania najlepszych wyników każdej reprezentacji dobrano różne rodziny falek – *Morlet*, *Mexican Hat*, *Coiflet5*.

Rodzaj algorytmu	FaceSwap			FakeApp			Face2Face		
Parametr	ACC [%]	PREC [%]	REC [%]	ACC [%]	PREC [%]	REC [%]	ACC [%]	PREC [%]	REC [%]
Zespół	75.35	76.20	77.03	76.99	74.12	74.38	70.82	73.03	70.47
Średnia członków	70.54	72.96	72.00	71.37	69.27	67.93	64.50	64.89	62.69
Std członków	5.65	5.10	3.16	7.98	5.27	7.01	4.64	8.58	9.11
Najlepszy	71.86	71.09	70.60	73.81	71.59	71.67	66.47	69.64	69.20
Najgorszy	65.01	65.67	66.68	63.50	65.23	66.82	60.58	58.24	55.06

Dla porównania wyników eksperymenty powtórzono przy reprezentacji kątowej obrazów, tworząc tensory w podobny sposób jak dla reprezentacji amplitudowej (Tabele 5.1-5.3). Najlepsze wyniki uzyskano dla tensora numer 3, ale ich poziom jest zdecydowanie niższy w porównaniu do reprezentacji amplitudowej. W tabeli 5.4 przedstawiono zbiorczo wyniki działania zespołu oraz wyniki statystyczne indywidualnych klasyfikatorów tworzących zespół przy rozpoznaniu trzech rodzajów podróbek (*FaceApp*, *FaceSwap*, *Face2Face*). Wszystkie obrazy tworzone były przy użyciu różnych rodzin falek: faleki Morleta (*FaceSwap*), „*Mexican Hat*” (*FaceApp*) oraz *Coiflet5* (*Face2face*).

Wyniki uzyskane przy reprezentacji kątowej obrazów są zdecydowanie gorsze niż przy reprezentacji amplitudowej. Uzyskane wyniki wskazują, że wykrywanie podróbek obrazów przy użyciu tensorów zbudowanych z reprezentacji kątowych w nie jest efektywne (taka reprezentacja tensorowa traci wiele użytecznych informacji o obrazach poddanych analizie).

Wyniki zaprezentowane w rozprawie można porównać z innymi reprezentowanymi w literaturze światowej. W tabeli 5.5 przedstawiono zbiorczo wyniki rozpoznania obrazów deep fake z bazy *Faceforensics++* uzyskane przez różnych autorów [93] i przy zastosowaniu różnych metod przetwarzania (sieć kopułowa, optyczna, syjamska CNN, *Xception*, *MesoInceptionv4*, CNN+GRU+STN oraz CNN plus sygnały biologiczne). Deklarowana dokładność rozpoznania waha się od 70.47% przy bardzo wysokiej kompresji danych (oznaczenie c40) do 99.26%, przy obrazach surowych (bez kompresji – oznaczenie raw).

Tabela 5.5 Wyniki porównawcze uzyskane na bazie *Faceforensics++* różnymi metodami. Oznaczenie c23 dotyczy średniej kompresji danych, c40 – wysokiej kompresji, raw – bez kompresji.

Publikacja	Metoda	Baza danych	Dokładność [%]
Nguyen et al. [94]	Sieć kopułowa	Faceforensics++Face2face	93.11
Zhao et al.[70]	Sieć optyczna	Faceforensics++DeepFake	98.10
Corozzino et al.[77]	Sieć syjamska CNN	Faceforensics++	92.14
Rossler et al. [79]	Xception CNN	Faceforensics++(raw)	99.26
		Faceforensics++(c23)	95.73
		Faceforensics++(c40)	81.00
Afchar et al. [73]	MesoInceptionv4	Faceforensics++(raw)	95.23
		Faceforensics++(c23)	83.10
		Faceforensics++(c40)	70.47
Sabir et al. [76]	CNN+GRU+STN	Faceforensics++(DeepFake)	96.90
		Faceforensics++(Face2Face)	94.35
		Faceforensics++(FaceSwap)	96.3
Ciftci et al. [74]	CNN plus sygnały biologiczne	Faceforensics++(DeepFake)	93.75
		Faceforensics++(Face2Face)	96.25
		Faceforensics++(FaceSwap)	95.25

Najlepsze wyniki uzyskane w rozprawie na bazie obrazów skompresowanych (c23) są zbliżone do wyników aktualnie najlepszych prezentowanych w literaturze światowej.

Należy podkreślić, że proponowany w rozprawie sposób przetwarzania obrazów jest złożony obliczeniowo i wymaga dużych zasobów obliczeniowych i stosunkowo długiego czasu dla uzyskania satysfakcjonujących wyników. Stąd w badaniach ograniczono się do obrazów wstępnie skompresowanych (model c23), w których z definicji traci się pewną ilość informacji oryginalnej. Uwzględniając wszystkie etapy przetwarzania danych, przyjmując jako dane wejściowe strumień wideo, jest procesem złożonym i niełatwym w praktycznej implementacji on-line.

## 6 Podsumowanie i wnioski końcowe

Rozprawa dotyczy opracowania metod i algorytmów wykrywania anomalii procesów na podstawie zarejestrowanych sygnałów (jednowymiarowych ciągów czasowych oraz obrazów typu deep fake). Anomalia procesu jest rozumiana jako wystąpienie nietypowego zbioru zarejestrowanych wartości obserwowanych zmiennych w procesie. W rozwiązaniach zaproponowanych w pracy zastosowano podejście wykorzystujące uczenie z nauczycielem bazujące na matematycznym modelu procesu zbudowanym w oparciu o informacje dotyczące potencjalnej przynależności danych wejściowych do określonej klasy (normalnej bądź anomalnej).

Opracowano szereg indywidualnych rozwiązań modeli klasyfikatora zintegrowanych w zespół poprzez głosowanie większościowe. Zastosowano różne rozwiązania klasyfikatorów bazujące na odmiennych metodach podejmowania decyzji, czynnika ważnego w zapewnieniu niezależności działania poszczególnych jednostek zespołu. Zastosowane rozwiązania klasyfikatorów należą do typu płytkiego (KNN, SVM, MLP, las losowy, naiwny klasyfikator Bayesa, gradient boosting) oraz głębokiego w postaci różnych struktur sieci neuronowej CNN (*alexnet, mobilenetv2, resnet50, efficientnetb0, squeezeNet, googlenet, shufflenet, inceptionresnetv2*), wszystkie zintegrowane w zespole.

W każdym rozwiązaniu problemu klasyfikacyjnego bardzo ważną rolę odgrywają cechy diagnostyczne procesu, stanowiące atrybuty wejściowe dla klasyfikatorów. Podstawą ich tworzenia w pracy są wyniki transformacji falkowej zastosowanej do danych pomiarowych. Wykorzystano zarówno ciągłą transformację CWT jak i dyskretną formę DWT. W przypadku zastosowania CWT wynik transformacji jest w postaci obrazu, który może być bezpośrednio podany na wejście sieci głębokiej CNN realizującej zarówno generację cech diagnostycznych jak i funkcję klasyfikatora. W przypadku DWT generuje się wiele wyjściowych sygnałów (szeregów czasowych) na z góry zdefiniowanych poziomach dekompozycji. W wyniku zastosowania opisu statystycznego sygnałów na poszczególnych poziomach i procedury selekcji generuje się cechy diagnostyczne (atrybuty wejściowe) dla zespołu klasycznych klasyfikatorów.

Zaproponowano i przebadano dwa znacznie różniące się podejścia do rozwiązania postawionego problemu wykrycia anomalii. Pierwszy system wykorzystuje wyniki DWT przy tworzeniu cech diagnostycznych procesu. Na podstawie analizy sygnałowej na każdym poziomie dekompozycji zdefiniowano w sumie 78 deskryptorów statystycznych z których w wyniku selekcji

metodą  $\chi^2$  wybiera się zredukowany zbiór cech stanowiących atrybuty wejściowe dla zbioru klasyfikatorów płytkich [52].

Równolegle opracowany system drugi wykorzystuje transformację CWT, której wyniki w postaci obrazów zasilają zespół sieci głębokich CNN podejmujących decyzję przynależności do klasy (normalnej bądź anomalnej). W tym rozwiązaniu cechy diagnostyczne procesu tworzone są automatycznie poprzez zastosowaną strukturę wielowarstwową sieci o połączeniu lokalnym. Zastosowanie różnych rozwiązań strukturalnych pre-trenowanych sieci CNN (tryb transfer learning) tworzących zespół ekspertowy pozwoliło zwiększyć niezależność działania poszczególnych jednostek tworzących zespół i polepszyć jakość działania systemu.

Opracowana metodologia przetwarzania danych pomiarowych została sprawdzona i przetestowana na trzech różnych problemach praktycznych:

- Wykrywanie anomalii w sygnałach EKG.
- Wykrywanie uszkodzeń łożyska tocznego na podstawie zarejestrowanych sygnałów czujników akcelerometrycznych.
- Wykrywanie sfałszowanych obrazów typu deep fake na przykładzie obrazów twarzy wyekstrahowanych z filmów video.

Przebadane zostały różne warianty doboru parametrów obu systemów, uzyskując w efekcie bardzo dobre wyniki wykrycia anomalii, lepsze lub porównywalne z najlepszymi rezultatami prezentowanymi w literaturze światowej. Autor uważa, że cel pracy sformułowany we rozdziale pierwszym został osiągnięty.

Za główne osiągnięcia rozprawy autor uważa:

- Opracowanie systemu komputerowego do wykrywania anomalii procesów reprezentowanych przez ciągi czasowe. System bazuje na zastosowaniu transformacji falkowej DWT, której wyniki stanowią podstawę definicji cech diagnostycznych, stanowiących atrybuty wejściowe dla zespołu klasyfikatorów typu płytkiego.
- Opracowanie systemu bazującego na transformacji falkowej CWT i zespole sieci głębokich CNN. W wyniku eksperymentów numerycznych udowodniono, że system ten jest uniwersalny i ma zastosowanie zarówno do procesów charakteryzowanych przez ciągi czasowe jak i obrazów.
- Przeprowadzenie ogromnej liczby eksperymentów numerycznych dla sprawdzenia skuteczności działania obu zaproponowanych systemów. Wyniki badań potwierdziły

przydatność opracowanych systemów w wykryciu obu typów anomalii (sygnałowej i obrazowej). W przypadku analizy wybranych typów anomalii dotyczącej ciągów czasowych EKG i sygnałów łożysk tocznych uzyskano dokładność rozpoznawania anomalii zbliżoną do 100%. Dokładność wykrywania sfałszowanych obrazów typu *deep fake* jest również wysoka, choć zależna od typu zniekształcenia obrazu. Najlepszy wynik ACC = 97.33 % uzyskano w bazie Forensics++ dla zniekształceń obrazu uzyskanych przy użyciu algorytmu FakeApp.

Problemy związane z wykrywaniem anomalii procesów są nadal aktualne i mogą podlegać dalszej optymalizacji [95]. Ulepszenie zaproponowanych algorytmów wymaga dalszych prac ukierunkowanych na przebadanie zaproponowanych rozwiązań na szerszej bazie danych dostępnych w Internecie. Interesujący jest zwłaszcza problem wykrywania podróbek obrazów typu deep fake. Jest to problem nowy, dla którego podstawy zostały zdefiniowane w roku 2019 przez najważniejsze instytucje światowe w tej dziedzinie [96].

Problemem szczególnie istotnym jest przyspieszenie procesu przetwarzania danych umożliwiające przejście do trybu on-line. Obecne algorytmy zaprezentowane w rozprawie należałoby znacznie przyspieszyć, choćby przez zastosowanie podejścia równoległego.

## Literatura:

- [1] Mehrotra K., Mohan C. and Huang H., *Anomaly detection principles and algorithms*, Springer, 2017.
- [2] Shlens J., *A Tutorial on Principal Component Analysis*, Google Research, 2014.
- [3] Osowski S., *Sieci neuronowe do przetwarzania informacji*, Warszawa: OWPW, 2021.
- [4] Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A. and Bengio Y., *Generative Adversarial Nets (PDF)*, Proceedings of the International Conference on Neural Information Processing Systems, 2014.
- [5] Tan P. N., Steinbach M. and Kumar V., *Introduction to data mining*, Boston: Pearson Education Inc., 2014.
- [6] Czajkowski A., Skorny G. and Olaszak W., *Hermite polynomials application for expanding functions in the series by these polynomials*, Problems of Applied Sciences, vol. 6, s. 67-76, 2017.
- [7] Box E. and Jenkins G., *Szeregi czasowe – analiza i prognozowanie*, Warszawa: PWN, 1983.
- [8] Goodman W. and Gray R., *Fourier Transforms*, Kluwer, 1995.
- [9] Mertins A., *Signal analysis: wavelets, filter banks, time frequency transform and applications*, Nowy Jork: Wiley, 1999.
- [10] Daubechies I., *Ten lectures on wavelets*, Filadelfia: SIAM, 1992.
- [11] van der Maaten L. and Hinton G., *Visualising data using t-SNE*, Journal of Machine Learning Research, vol. 9, n. 2579-2602, 2008.
- [12] Roelants P., Slater D., Spacagna G. and Zocca V., *Deep Learning. Uczenie głębokie z językiem Python. Sztuczna inteligencja i sieci neuronowe*, Gliwice: Helion, 2018.
- [13] Murphy K. P., *Probabilistic Machine Learning: An Introduction (Adaptive Computation and Machine Learning)*, MIT Press Ltd, 2022.



- [14] Case Western Reserve University, *Bearing Data Center*, [Online]: <https://engineering.case.edu/bearingdatacenter> [Dostęp: Maj 2022].
- [15] Lai M-J., *Popular Wavelet Families and Filters and Their Use*, Computational Complexity, 2012.
- [16] Xian-Zhi Y., *An introduction to wavelet theory and its applications in statistics*, Halifax: Dalhousie University, 1997.
- [17] Mallat S., *A wavelet tour of signal processing*, Amsterdam: Elsevier, 1999.
- [18] Matlab, Natick: MathWorks, 2021.
- [19] Cutler A., Cutler D. and Stevens J., *Random Forests*, Ensemble Machine Learning: Methods and Applications, Springer, 2011, s. 157-176.
- [20] Breiman L., *Random forests*, Machine Learning, vol. 45, n. 11, s. 5-32, 2001.
- [21] Goodfellow I., Bendio Y. and Courville A., *Deep learning*, Massachusetts: MIT Press, 2016.
- [22] Geurts P., Ernst D, and Wehenkel L., *Extremely randomized trees*, Springer Science, 2006.
- [23] Cortes C. and Vapnik V., *Support-Vector Networks*, Machine Learning, vol. 20, s. 273-297, 1995.
- [24] Vapnik V., Golowich S. and Smola A., *Support vector method for function approximation, regression*, Advances in Neural Information Processing Systems, vol. 9, s. 281– 287, 1997.
- [25] Rasmussen C. E., *Gaussian Processes for Machine Learning*, MIT Press Ltd, 2006.
- [26] Wettschereck D. and Dietterich T., *On nearest neighbor classification using adaptive choice of k*, Journal of computational and graphical statistics, 2007 s. 482–502, 2007.
- [27] Breuel T. M., *On the Convergence of SGD Training of Neural Networks*, Google Inc., 2015.
- [28] Berrar D., *Bayes' Theorem and Naive Bayes Classifier*, Tokyo Institute of Technology, Tokyo, 2018.

- [29] Brownlee J., *Deep Learning for Natural Language Processing: Develop Deep Learning Models for your Natural Language Problems*, Machine Learning Mastery, 2017.
- [30] Le Cun Y. and Bengio Y., *Convolutional networks for images, speech, and time-series*, The handbook of brain theory and neural networks, M. Arbib, Ed., Massachusetts, MIT Press, 1995.
- [31] Brownlee J., *Master machine learning algorithms*, Ebook, 2020.
- [32] Tuffery S., *Data mining and statistics for decision making*, Nowy Jork: Wiley, 2011.
- [33] Kingma D. and Ba J., *Adam: a method for stochastic optimization*, arXiv:1412.6980, 2014.
- [34] Krizhevsky A., Sutskever I. and Hinton G., *Image net classification with deep convolutional neural networks*, Advances in Neural Information Processing Systems, vol. 25, s. 1-9, 2012.
- [35] Schölkopf B. and Smola A., *Learning with kernels*, Cambridge MA: MIT Press, 2002.
- [36] Iandola F., Han S., Moskevycz M., Ashraf K., Dally W. and Kreutzer K., *Squeezenet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size*, Conference ICLR, 2017, s. 1-13.
- [37] Szegedy C., Ioffe S. and Vanhoucke V., *Inveption-v4, Inception-ResNet and the impact of residual connections on learning*, arXiv:1602.07261, 2016.
- [38] Huang G., Liu Z., van der Maaten L. and Weinberger K., *Densely connected convolutional networks*, arXiv:1608.06993, 2018.
- [39] Zhang X., Zhou X., Lin M. and Sun J., *ShuffleNet: an extremely efficient convolutional neural network for mobile devices*, arXiv:1707.01083, 2017.
- [40] Howard A., Zhu M., Chen B. and Kalenichenko D., *MobileNets: Efficient convolutional neural networks for mobile vision applications*, arXiv:1704.04861, 2017.
- [41] Li H. and Boulanger P., *Structural Anomalies Detection from Electrocardiogram (ECG) with Spectrogram and Handcrafted Features*, Sensors, vol. 22, n. 7, s. 1-22, 2022.

- [42] Li H. and Boulanger P., *A Survey of Heart Anomaly Detection Using Ambulatory Electrocardiogram (ECG)*, Sensors (Basel), vol. 20, s. 1461, 2020.
- [43] Sahoo S., Kanungo B., Behera S. and Sabut S., *Multiresolution wavelet transform based feature extraction and ECG classification to detect cardiac abnormalities*, Measurement, vol. 108, s. 55-66, 2017.
- [44] Rejesh K. N. and Dhuli R., *Classification of ECG heartbeats using nonlinear decomposition methods and support vector machine*, Journal Computers in Biology and Medicine, vol. 87, s. 271-284, 2017.
- [45] Pan J. and Tompkins W., *A Real-Time QRS Detection Algorithm*, IEEE Transactions on biomedical engineering, BME-32, 1985.
- [46] Gao X., *Non-invasive Detection and Compression of Fetal Electrocardiogram*, Interpreting Cardiac Electrograms - From Skin to Endocardium, 2017.
- [47] Gołgowski M. and Osowski S., *Classical versus deep learning methods for anomaly detection in ECG using wavelet transformation*, Przegląd Elektrotechniczny, vol. 97, n. 6/2021.
- [48] PhysioNet, *The Research Resource for Complex Physiologic Signals*, [Online]: <https://physionet.org/>. [Dostęp 07 2020].
- [49] van Alste J., van Eck W. and Herrman O., *ECG baseline wander reduction using linear phase filters*, Comput. Biomed. Res, vol. 19, s. 417-427, 1986.
- [50] Sharma K.. [Online]: [https://github.com/antimattercorrade/Pan\\_Tompkins\\_QRS\\_Detection](https://github.com/antimattercorrade/Pan_Tompkins_QRS_Detection). [Dostęp 04 2021].
- [51] Gołgowski M. and Osowski S., *Anomaly detection in ECG using wavelet transformation*, Computational Problems of Electrical Engineering (CPEE), s. 1-4, 2020.
- [52] Lyons J., *Chi-squared Statistic*, Practical Cryptography, 2015.
- [53] Ribeiro A., Ribeiro M. and Paixão G., *Automatic diagnosis of the 12-lead ECG using a deep neural network*, Nature Communications, vol. 1760, 2020.

- [54] SciPy Community, *SciPy documentation - CWT*, [Online]: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.cwt.html>. [Dostęp 07 2021].
- [55] Memisevic R., Zach C., Hinton G. and Pollefeys M., *Gated Softmax Classification*, 24th Annual Conference on Neural Information Processing Systems, Toronto 2010.
- [56] Stanford - Spring 2022, *CS231n: Deep Learning for Computer Vision*, [Online]: <http://cs231n.stanford.edu/>. [Dostęp 01 2022].
- [57] Zhang J., Yang K., Jiang Y. and Xia L., *A method for bearing fault diagnosis of mine hoist using convolutional attention autoencoder*, [Ebook] 2021.
- [58] Said L., Fnaiech F., Capolino G. and Henao H., *Stator Current Bi-Spectrum Patterns for Induction Machines Multiple-Faults Detection*, IECON 2012, 2012.
- [59] Wang S., Wang D., Kong D., Wang J., Li W. and Zhou S., *Few-Shot Rolling Bearing Fault Diagnosis with Metric-Based Meta Learning*, *Sensors*, vol. 20, 2020.
- [60] Huang H. and Baddour N., *Bearing Vibration Data under Time-varying Rotational Speed Conditions*, 2018. [Online]: <https://data.mendeley.com/datasets/v43hmbwxpm>. [Dostęp 01 2021].
- [61] Huang H. and Baddour N., *Bearing vibration data collected under time-varying rotational speed conditions*, *Journal Data in Brief*, vol. 21, s. 1745-1749, 2018.
- [62] Golgowski M. and Osowski S., *Detection of bearing failures using wavelet transformation and machine learning*, IEEE International Joint Conference on Neural Networks (IJCNN), Padua, 2022.
- [63] SciPy Community, *SciPy documentation - chi2*, [Online]: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.chisquare.html>. [Dostęp 05 2022].

- [64] Wang D., Guo Q., Song Y., Gao S. and Li Y., *Application of multiscale learning neural network based on CNN in bearing fault diagnosis*, J. Signal Processing Systems, vol. 91, n. 10, s. 1205-1217, 2019.
- [65] Paderborn University, *Bearing DataCenter*, [Online]: <https://mb.uni-paderborn.de/en/kat/main-research/datacenter/bearing-datacenter/data-sets-and-download>. [Dostęp 02 2021].
- [66] Han B., Hui Z., Ming S. and Wu F., *A new bearing fault diagnosis method based on capsule network and Markov transition field/Gramian angular field*, Sensors, vol. 21, n. 22, s. 7762, 2021.
- [67] Bach-Andersen M., Rømer-Odgaard B. and Winther O., *Deep Learning for Automated Drivetrain Fault Detection*, Wing Energy, vol. 21, n. 1, s. 29-41, 2018.
- [68] Eren L., Ince T. and Kiranyaz S., *A generic intelligent bearing fault diagnosis system using compact adaptive 1D CNN classifier*, J. Signal Processing Systems, vol. 91, n. 2, s. 179-189, 2019.
- [69] Li H., Deng J., Yuan S., Feng P. and Arachige D.K., *Monitoring and Identifying Wind Turbine Generator Bearing Faults Using Deep Belief Network and EWMA Control Charts*, Frontiers in Energy Research, vol. 9, n. 1-10, 2021.
- [70] Zhao Y., Ge W., Li W., Wang R., Zhao L. and Ming J., *Capturing the Persistence of Facial Expression Features for Deepfake Video Detection*, Information and Communications Security. ICICS 2019. Lecture Notes in Computer Science, vol. 11999.
- [71] MetaAI, *Deepfake Detection Challenge Results: An open initiative to advance AI*, 2020. [Online]: <https://ai.facebook.com/blog/deepfake-detection-challenge-results-an-open-initiative-to-advance-ai/>. [Dostęp 04 2022].
- [72] *FaceForensics++*, [Online]: <https://github.com/ondyari/FaceForensics>. [Dostęp 04 2022]

- [73] Afchart D., Nozick V., Yamahishi J. and Echizen I., *MesoNet: a Compact Facial Video Forgery Detection Network*, 2018 IEEE International Workshop on Information Forensics and Security (WIFS), 2018.
- [74] Ciftci U. A., Demir I. and Yin L., *FakeCatcher: Detection of Synthetic Portrait Videos using Biological Signals*, IEEE Transactions on pattern analysis and machine intelligence, vol. 10, n. 10, 2020.
- [75] *DARPA Is Taking On the Deepfake Problem*, [Online]: [www.nextgov.com](http://www.nextgov.com). [Dostęp 03 2021].
- [76] Sabir E., Cheng J., Jaiswal A., AbdAlmageed W., Masi I. and Natarajan P., *Recurrent convolutional strategies for face manipulation detection in videos*, Interfaces (GUI), vol. 3, n. 1, s. 80-87, 2019.
- [77] Cozzolino D., Poggi G. and Verdoliva L., *Extracting camera-based fingerprints for video forensics*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Nowy Orlean 2019.
- [78] Wang K., Gou C., Duan Y., Lin Y., Zheng X. and Wang F., *Generative Adversarial Networks: Introduction and Outlook*, IEEE/CAA Journal of Automatica Sinica, vol. 4, 2017.
- [79] Rossler A., Cozzolino D., Verdoliva L., Riess C., Thies J. and Niessner M., *Faceforensics++: Learning to detect manipulated facial images*, Proceedings of the IEEE/CVF international conference on computer vision (ICCV), 2019.
- [80] Kazemi V. and Sullivan J., *One Millisecond Face Alignment with an Ensemble of Regression Trees*, Royal Institute of Technology, 2015.
- [81] Li L., Bao J., Yang H., Chen D. and Wen F., *Advancing high fidelity identity swapping for forgery detection*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, s. 5074–5083, 2020.

- [82] Heo J. and Savvides M., *Gender and ethnicity specific generic elastic models from a single 2D image for novel 2D pose face synthesis and recognition*, IEEE Trans Pattern Anal Mach Intell., 2012.
- [83] Chesakov D., Maltseva A., Groshev A., Kuznetsov A. and Dimitrov D., *A new face swap method for image and video domains: a technical report*, arXiv:2202.03046, 2022.
- [84] Perov I., Gao D., Chervoniy N. and Liu K., *DeepFaceLab: Integrated, flexible and extensible face-swapping framework*, arXiv:2005.05535, 2020.
- [85] Masood M., Nawaz M., Malik K., Javed A, and Irtaza A., *Deepfakes generation and detection: state-of-the-art, open challenges, countermeasures, and way forward*, Projects: Multimedia Forensics, 2021.
- [86] Tran D., *face2face-demo*, 2017. [Online]: <https://github.com/datitran/face2face-demo>. [Dostęp 05 2022].
- [87] Deniz O. and Bueno G., *Face recognition with histograms of oriented gradients*, Proceedings of the International Conference on Computer Vision Theory and Applications, s. 339-344, 2010.
- [88] Dalal N. and Triggs B., *Histograms of Oriented Gradients for Human Detection*, Conference on Computer Vision and Pattern Recognition (CVPR), s. 1-10, 2005.
- [89] Swarnima S., Durgesh S. and Vikashv Y., *Face Recognition Using HOG Feature Extraction and SVM classifier*, International Journal of Emerging Trends in Engineering Research, vol. 8, n. 9, 2020.
- [90] He K., Zhang X., Ren S. and Sun J., *Deep Residual Learning for Image Recognition*, arXiv:1512.03385, 2015.
- [91] S. Harihara and P. Gopala, *Improved Face Recognition Rate Using HOG Features and SVM Classifier*, IOSR Journal of Electronics and Communication Engineering, vol. 11, n. 4, 2016.
- [92] *Biblioteka dlib*, [Online]: <http://www.dlib.net>, [Dostęp 01 2022].

- [93] Yu P., Xia Z, Fei J. and Lu Y., *A survey on deepfake video detection*, ETBiometrics, vol. 10, n. 6, s. 607-624, 2021.
- [94] Nguyen H. H., Yamahishi J. and Echizen I., *Capsule-forensics: Using capsule networks to detect forged images and videos*, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019.
- [95] Dąbrowska I., *Deepfake – nowy wymiar internetowej manipulacji.*, Zarządzanie Mediami, vol. 8, n. 2, 2020.
- [96] *Facebook i Microsoft chcą skutecznie walczyć z deepfake'ami. 10 mln dolarów na nowe narzędzia*, [Online]: [www.wirtualnemedi.pl](http://www.wirtualnemedi.pl). [Dostęp 03 2021].