

Politechnika Śląska  
Wydział Mechaniczny Technologiczny  
Katedra Podstaw Konstrukcji Maszyn

Mateusz Kalisch

**METODA  
DETEKCJI USZKODZEŃ  
W DIAGNOSTYCE  
PROCESÓW  
Z UWZGLĘDNIENIEM  
KONTEKSTU**

Gliwice 2023

### *Recenzenci*

Prof. dr hab. inż. Wojciech Batko - Zakład Mechaniki i Budowy Maszyn,  
Karpacka Państwowa Uczelnia w Krośnie

Dr hab. inż. Jędrzej Mączak, prof. PW - Zakład Ciągników i Napędów  
Hydraulicznych, Wydział Samochodów i Maszyn Roboczych,  
Politechnika Warszawska

### *Redaktor zeszytów*

Marek Wyleżoł

### *Redaktor techniczny*

Marek Wyleżoł

### *Projekt okładki*

Wojciech Cholewa, Marek Wyleżoł

## Zeszyt Naukowy Katedry Podstaw Konstrukcji Maszyn

### *BibTeX*

```
@BOOK{, title = {Metoda detekcji uszkodzeń w diagnostyce  
procesów z uwzględnieniem kontekstu}, publisher = {Politechnika  
Śląska, Katedra Podstaw Konstrukcji Maszyn}, year = {2023,  
author = {Kalisch, M.}, volume = {157}, series = {Zeszyty  
Naukowe}, address = {Gliwice}}
```

**ISBN 978-83-60759-32-5**

### *Wydawca*

Katedra Podstaw Konstrukcji Maszyn

Wydział Mechaniczny Technologiczny

Politechnika Śląska

ul. Konarskiego 18a

44-100 Gliwice

Tel. (32) 237-14-67

<https://kpkkm.polsl.pl>

# Od autora

Zeszyt został opracowany na podstawie rozprawy doktorskiej, którą wykonałem pod kierunkiem dr hab. inż. Anna Timofiejczuk, prof. PŚ oraz dr hab. inż. Piotr Przystałka, prof. PŚ. Publiczna obrona rozprawy odbyła się 11. lipca 2023 roku przed Komisją powołaną przez Radę Wydziału Mechanicznego Technologicznego.

Składam serdeczne podziękowania recenzentom rozprawy doktorskiej, Prof. dr hab. inż. Wojciechowi Batko oraz dr hab. inż. Jędrzejowi Mączakowi, prof. Politechniki Warszawskiej za cenne uwagi, które starałem się uwzględnić w niniejszym zeszycie.

Pragnę wyrazić swoją wdzięczność mojej Pani promotor prof. Annie Timofiejczuk oraz promotorowi pomocniczemu prof. Piotrowi Przystałce za cenne wskazówki, inspirację i wsparcie podczas przygotowywania rozprawy doktorskiej. Składam serdeczne podziękowania Koleżankom i Kolegom z Katedry Podstaw Konstrukcji Maszyn Politechniki Śląskiej za okazaną mi pomoc i życzliwość w czasie wykonywania pracy.

Niniejszą książkę dedykuję mojej najbliższej rodzinie oraz mojej partnerce Beacie, dziękując za ich wsparcie, wyrozumiałość i cierpliwość.

*Gliwice, lipiec 2023*

*Mateusz Kalisch*



# Spis treści

<b>Od autora</b>	<b>3</b>
<b>Wykaz ważniejszych oznaczeń</b>	<b>7</b>
<b>Rozdział 1. Wstęp</b>	<b>9</b>
1.1. Problem badawczy i cel rozprawy	10
1.2. Zakres rozprawy	11
<b>Rozdział 2. Wybrane zagadnienia diagnostyki procesów przemysłowych</b>	<b>13</b>
2.1. Pojęcia podstawowe	13
2.1.1. Obiekt diagnozowania	13
2.1.2. Sygnał, cecha sygnału i sygnał diagnostyczny	14
2.1.3. Fuzja danych	15
2.1.4. Kontekst	15
2.2. Zadania diagnostyki procesów przemysłowych	16
2.2.1. Diagnostyka wsparta modelami	17
2.2.2. Diagnostyka symptomowa	19
2.3. Zadanie klasyfikacji stanu technicznego obiektu	20
2.3.1. Klasyfikatory bazowe	20
2.3.2. Systemy wieloklasyfikatorowe	26
2.3.3. Metody oceny klasyfikatorów	29
2.4. Optymalizacja w diagnostyce	31
2.5. Podsumowanie	32
<b>Rozdział 3. Kontekst</b>	<b>35</b>
3.1. Definicja kontekstu	35
3.2. Kontekst w zadaniach klasyfikacji	39
3.3. Podsumowanie	45
<b>Rozdział 4. Metoda detekcji uszkodzeń z zastosowaniem kontekstu</b>	<b>47</b>
4.1. Akwizycja danych i ekstrakcja cech	47
4.2. Zadanie klasyfikacji	54
4.2.1. Selekcja cech	54
4.2.2. Miary oceny sprawności klasyfikatorów	55

4.3.	Fuzja klasyfikatorów z wykorzystaniem kontekstu .....	56
4.4.	Optymalizacja parametrów klasyfikatora kontekstowego .....	58
4.4.1.	Funkcje kryterialne .....	59
4.4.2.	Optymalizacja liczby atrybutów .....	61
4.4.3.	Dobór parametrów klasyfikatorów .....	63
4.5.	Poszukiwanie rozwiązania optymalnego .....	63
4.6.	Implementacja klasyfikatora w procesie diagnostycznym .....	66
4.7.	Podsumowanie .....	67
<b>Rozdział 5.</b>	<b>Badania weryfikacyjne</b> .....	<b>69</b>
5.1.	Obiekt badań .....	69
5.2.	Opis eksperymentu .....	71
5.3.	Akwizycja i przygotowanie danych .....	75
5.4.	Realizacja procesu diagnostycznego .....	81
5.4.1.	Trenowanie modeli detekcji uszkodzeń .....	82
5.4.2.	Zastosowanie modeli detekcji uszkodzeń .....	83
5.5.	Prezentacja i dyskusja wyników zastosowania opracowanej metody detekcji uszkodzeń .....	83
5.5.1.	Analiza wyników procesu optymalizacji .....	83
5.5.2.	Analiza wyników detekcji uszkodzeń .....	88
5.6.	Podsumowanie .....	106
<b>Rozdział 6.</b>	<b>Podsumowanie i wnioski</b> .....	<b>107</b>
6.1.	Wnioski końcowe .....	108
6.2.	Plan przyszłych badań .....	108
<b>Rozdział A.</b>	<b>Zbiorne wyniki eksperymentu</b> .....	<b>109</b>
<b>Bibliografia</b>	.....	<b>137</b>
<b>Streszczenie</b>	.....	<b>145</b>
<b>Summary</b>	.....	<b>146</b>

# Wykaz ważniejszych oznaczeń

$\Phi$	Komitet klasyfikatorów / klasyfikator kontekstowy
$\phi$	Klasyfikator
$\chi$	Algorytm trenujący klasyfikator
$\mathbf{E}, \mathbf{e}$	Zbiór danych, podzbiór danych
$\mathbf{E}_t, \mathbf{E}_s, \mathbf{E}_v$	Zbiór danych treningowych, testowych, walidacyjnych
$\mathbf{E}', \mathbf{e}'$	Zbiór danych, podzbiór danych z zredukowaną liczbą cech
$z_p, s$	Zmienna procesowa, sygnał diagnostyczny
$sf$	Stan funkcjonalny
$S$	Przestrzeń symptomów
$F$	Zbiór klas / znanych uszkodzeń
$f$	Klasa / uszkodzenie
$\mathbf{CL}, \mathbf{CP}, \mathbf{CK}$	Zbiór cech liczbowych sygnału, cech podstawowych, cech kontekstowych
$\mathbf{cl}, \mathbf{cp}, \mathbf{ck}$	Zbiór wartości cechy liczbowej, podstawowej, kontekstowej
$cl, cp, ck$	Cecha liczbowa sygnału, cecha podstawowa, cecha kontekstowa
$\mathbf{p}$	Wektor parametrów sterujących algorytmem trenowania klasyfikatora
$\mathbf{w}$	Wektor wag
$sp$	Stopień pewności klasyfikatora o decyzji
$kd$	Końcowa decyzja klasyfikatora
$\mathbf{KD}$	Zbiór końcowych decyzji
$\mathbf{O}$	Zbiór rozwiązań optymalnych
$o$	Rozwiązanie optymalne
$r_{td}, r_{fd}$	Wskaźnik prawidłowych, fałszywych alarmów
$TPR$	Wskaźnik wyników prawdziwie pozytywnych ( <i>True Positive Rate</i> )
$TNR$	Wskaźnik wyników prawdziwie negatywnych ( <i>True Negative Rate</i> )
$AVG_{TPR}$	Średnia wartość wskaźników $TPR$
$AVG_{TNR}$	Średnia wartość wskaźników $TNR$
$\mathbf{P}_t$	Początkowy zbiór osobników populacji w pokoleniu $t$
$\mathbf{Q}_t$	Zbiór osobników populacji utworzonej na bazie zbioru $P$ w pokoleniu $t$
$\mathbf{R}_t$	Zbiór wszystkich osobników w populacji $t$ , $P_t \cup Q_t$
$U$	Niezdominowany front rozwiązań optymalnych





# Rozdział 1

## Wstęp

Diagnostyka techniczna odgrywa kluczową rolę w efektywnym zarządzaniu procesami przemysłowymi, usprawniając proces detekcji i naprawy uszkodzeń. Uszkodzenia występujące podczas eksploatacji maszyn i urządzeń biorących udział w realizacji procesów przemysłowych w istotny sposób wpływają na zmniejszenie efektywności procesu oraz redukcję wyniku ekonomicznego. Uszkodzenia, które w stosunkowo długim przedziale czasu nie zostaną wykryte, mogą doprowadzić do znacznego i trwałego zniszczenia maszyn i urządzeń, co w konsekwencji może doprowadzić do długotrwałego wyłączenia z użycia całego ciągu technologicznego. Nieprawidłowo działający proces może mieć również istotny wpływ na bezpieczeństwo pracy. Niewykryte uszkodzenia mogą przyczynić się do zagrożenia zdrowia i życia osób pracujących w bezpośrednim sąsiedztwie środka technicznego lub też doprowadzić do zanieczyszczenia środowiska naturalnego. Z tego powodu w związku z rosnącym poziomem złożoności maszyn i systemów przemysłowych, istnieje potrzeba opracowywania zaawansowanych systemów wnioskowania, które są w stanie efektywnie wspomagać ich diagnostykę (Korbicz *i in.*, 2004; Patton *i in.*, 2000; Caccavale i Villani, 2003).

W diagnostyce procesów przemysłowych wyróżnić można dwie podstawowe grupy metod diagnozowania (Korbicz *i in.*, 2004): podejście bazujące na modelu badanego obiektu oraz metody bazujące na diagnostyce symptomowej. W przypadku podejścia bazującego na modelu obiektu, diagnostyka procesów przemysłowych opiera się na porównaniu zachowania rzeczywistego systemu z jego teoretycznym modelem matematycznym. Dzięki temu podejściu, możliwe jest zidentyfikowanie odchyłeń od normy, które mogą sugerować występowanie uszkodzeń. Wyróżnić można różne metody modelowania procesów przemysłowych, takie jak modele oparte na równaniach różniczkowych, czy modele szeregów czasowych (Isermann, 2005; Jain *i in.*, 2000; Patton *i in.*, 2000). Z drugiej strony, metoda bazująca na diagnostyce symptomowej opiera się na analizie sygnałów diagnostycznych, takich jak np. temperatura, przepływ, ciśnienie czy napięcie, które są generowane przez badany obiekt. W przeciwieństwie do podejścia bazującego na modelu, metoda ta nie wymaga tworzenia modeli matematycznych procesu, co może upraszczać proces diagnostyki. Do tej metody zaliczyć można podejścia opierające się na analizie widmowej, analizie statystycznej czy metodach uczenia maszynowego (Korbicz *i in.*, 2004; Patton *i in.*, 2000).

W diagnostyce technicznej coraz częściej pojawia się również pojęcie kontekstu (Turney, 2002b; Turney, 2002c; Timofiejczuk, 2011; Korbicz *i in.*, 2004). Obszarem związanym z zagadnieniem diagnostyki technicznej, w którym w szczególności można zauważyć wykorzystanie kontekstu są systemy doradcze i ekspertowe (Weiser, 1991; Moczulski, 2005; Cholewa i Pedrycz, 1987). Kontekst jest częścią wiedzy, która pozwala na wybranie pewnego fragmentu wiedzy charakterystycznej dla danego kontekstu. Kontekst stanowi pewnego rodzaju filtr, który pozwala na ograniczenie wiedzy potrzebnej do realizacji zadania (Timofiejczuk, 2011). Jednym z obszarów, który pozwala na integrację kontekstu z symptomową metodą detekcji uszkodzeń, jest zastosowanie algorytmu uczenia maszynowego uwzględniającego kontekst. W tym celu modyfikuje się znane już algorytmy klasyfikacji lub metody łączenia wielu klasyfikatorów (Singh *i in.*, 2003; Nascimento *i in.*, 2018; Turney, 2002b), czego przykładem są tak zwane komitety klasyfikatorów.

Tworzenie komitetów, w których każdy z klasyfikatorów jest rozpatrywany niezależnie, prowadzi do konieczności dostrajania wielu odrębnych parametrów metod uczących. Istnieje wiele metod, które pozwalają na zautomatyzowanie tego procesu, jak np. przeszukiwanie losowe lub przeszukiwanie systematyczne. Jednak realizacja tego procesu może być zadaniem czasochłonnym. Z tego powodu stosuje się też inne podejścia, bazujące na zastosowaniu algorytmów optymalizacji takie jak algorytmy genetyczne czy ewolucyjne.

## 1.1. Problem badawczy i cel rozprawy

Geneza prac opisanych w niniejszej rozprawie jest związana z realizacją projektu szkieletowego systemu wspomagania podejmowania decyzji dla systemów monitorowania procesów, urządzeń i zagrożeń (DISESOR) (Przystałka i Sikora, 2017), a głównym obszarem wykorzystania systemu były procesy, urządzenia i zagrożenia występujące w górnictwie. Jednym z elementów tego projektu było opracowanie modułu szkieletowego systemu eksperckiego, który poprzez zastosowanie wybranych metod uczenia maszynowego pozwalał na zdefiniowanie takiego procesu wnioskowania, którego rezultatem była diagnoza prezentowana użytkownikowi (np. dyspozytorowi) w postaci komunikatu w systemie SCADA. Istotnym elementem systemów ekspertowych oraz doradczych jest mechanizm objaśniający proces uzyskania wyniku wnioskowania (Cholewa i Pedrycz, 1987). Badania przemysłowe i prace rozwojowe przeprowadzone w ramach projektu DISESOR wykazały, że bardzo interesującym kierunkiem badań jest uwzględnienie kontekstu w procesie budowy klasyfikatorów i meta-klasyfikatorów. Przedstawiono kilka koncepcyjnych przykładów rozwiązania zadania budowy systemu detekcji i/lub izolacji uszkodzeń bez uwzględnienia lub z uwzględnieniem informacji o kontekście w odniesieniu do diagnozowania maszyn i urządzeń (Kalisch *i in.*, 2015; Kalisch, 2018; Przystałka *i in.*, 2018). Rozpoczęto również prace badawcze mające na celu zastosowanie takiego podejścia do diagnostyki procesów przemysłowych (Wachla *i in.*, 2015).

Na podstawie doświadczeń wyniesionych z realizacji projektu DISESOR stwierdzono, że opracowane podejścia bazujące na metodach uczenia maszynowego pozwalają w całości lub częściowo automatyzować proces tworzenia modeli klasyfikatorów ale też wy-

kazują pewne wady. Modele klasyfikatorów utworzone dla problemu detekcji uszkodzeń opracowywane w ramach procesów wnioskowania w systemie DISESOR za pomocą algorytmów uczenia nadzorowanego, wykazały brak możliwości lub znacznie utrudniały analizę procesu podejmowania decyzji. Część dostępnych metod klasyfikacji zupełnie wyklucza interpretację modelu ze względu na swój charakter, jak np. klasyfikatory oparte o sieci neuronowe. Inne klasyfikatory, takie jak drzewa decyzyjne, klasyfikatory regułowe, sieci bayesa czy logika rozmyta, dają możliwość interpretacji uzyskanego modelu detekcji i/lub izolacji uszkodzeń. Często ze względu na trudny charakter danych pochodzących z rzeczywistego procesu, analiza modelu tego typu jest znacznie utrudniona. Zauważono również, że niezbędne jest opracowanie podstaw teoretycznych fuzji wyników klasyfikacji z uwzględnieniem kontekstu w taki sposób, aby można było zastosować to podejście w diagnostyce procesów przemysłowych. W celu rozwiązania wymienionych wyżej problemów, określono następujące zadania badawcze:

- Sformułowanie podstaw teoretycznych metody detekcji uszkodzeń z zastosowaniem cech kontekstowych.
- Zdefiniowanie problemu optymalizacji wielokryterialnej z uwzględnieniem kryteriów bazujących na macierzy pomyłek dla zadania detekcji uszkodzeń.
- Opracowanie kryteriów optymalizacji dla zadania strojenia komitetu klasyfikatorów bazującego na kontekście.
- Zaplanowanie i wykonanie czynnych eksperymentów diagnostycznych mających na celu zgromadzenie danych uczących oraz przeprowadzenie walidacji zaproponowanej metody.

Celem rozprawy jest opracowanie metody detekcji uszkodzeń z uwzględnieniem kontekstu (Wachla *i in.*, 2015). W ramach realizacji pracy przygotowano podejścia bazujące na pojedynczych kontekstach będących częścią metody wnioskowania wykorzystującej komitety klasyfikatorów. Założono, że dodatkowe zastosowanie algorytmu optymalizacji, który podczas swojego działania wpływa na parametry uczenia klasyfikatorów oraz na wektor danych wejściowych, pozwoli na określenie optymalnego zbioru parametrów. Na podstawie tych parametrów możliwe jest utworzenie klasyfikatora spełniającego założenie dotyczące minimalizacji wskaźnika fałszywych alarmów, maksymalizacji wskaźnika prawidłowych alarmów i zachowanie jak największej powtarzalności wyników.

## 1.2. Zakres rozprawy

Rozprawa dotyczy zagadnień wnioskowania diagnostycznego, które prowadzone jest dla obiektów technicznych, głównie procesów przemysłowych. Treść rozprawy została zawarta w sześciu rozdziałach. Rozdział pierwszy zawiera genezę i opis problemu badawczego oraz zakres rozprawy.

Rozdział drugi dotyczy podstawowych koncepcji wnioskowania, stosowanych w diagnostyce procesów przemysłowych. Scharakteryzowano w nim pojęcia podstawowe stosowane w rozprawie w szczególności pojęcie kontekstu oraz przedstawiono zagadnienia

diagnostyki symptomowej i wspartej modelem. W rozdziale tym omówiono także wybrane algorytmy klasyfikacyjne, a także sposoby estymacji błędu klasyfikacji i miary oceny zdolności klasyfikacyjnej. Ponadto, w tej części pracy opisano najważniejsze zagadnienia związane z algorytmami optymalizacji, w szczególności optymalizacji wielokryterialnej i sposobu zastosowania algorytmu optymalizacji w zadaniu klasyfikacji.

Rozdział trzeci poświęcony jest w całości zagadnieniom związanym z pojęciem kontekstu. Opisano w nim ogólną definicję kontekstu, a następnie skupiono się na zastosowaniu kontekstu w uczeniu maszynowym, w szczególności uwzględniając metody bazujące na systemach wieloklasyfikatorowych, jak np. komitety klasyfikatorów. Omówiono również w jaki sposób uwzględnia się kontekst w diagnostyce procesów.

Rozdział czwarty zawiera opis metody detekcji stanu procesów przemysłowych z wykorzystaniem kontekstu oraz komitetów klasyfikatorów, opracowanej w ramach realizowanych badań. Opisano w nim szczegółowo kolejne etapy realizacji metody, w tym akwizycję danych i dalsze ich przetwarzanie poprzez np. ekstrakcję cech. W rozdziale tym opisano także w jaki sposób przygotowywane jest zadanie uczenia klasyfikatorów poprzez zastosowanie metody selekcji cech i oceny poprawności klasyfikacji. Przedstawiono sposób działania fuzji klasyfikatorów poprzez opisanie metody głosowania oraz uwzględnienie kontekstu w komitecie klasyfikatorów. W kolejnych częściach rozdziału czwartego omówiono w jaki sposób proces trenowania klasyfikatorów został połączony z wybraną metodą optymalizacji wielokryterialnej, w celu uzyskania wyników działania klasyfikatorów, zgodnych z założeniami zdefiniowanymi jako minimalizacja wskaźnika fałszywych alarmów, maksymalizacja wskaźnika prawdziwych alarmów oraz maksymalizacja powtarzalności uzyskiwanych wyników.

W rozdziale piątym opisano badania weryfikacyjne. Wykorzystano do tego celu stanowisko laboratoryjne, które umożliwiło wygenerowanie danych, które posłużyły do oceny zaproponowanej metody detekcji uszkodzeń z uwzględnieniem kontekstu. Zebrane wyniki zostały przedstawione w formie tabel i wykresów oraz szczegółowo omówione. Podczas analizowania wyników skupiono się na przedstawieniu wpływu parametrów bazowych metody na uzyskiwane rezultaty oraz porównano wyniki otrzymane dla opracowanej metody detekcji uszkodzeń z podstawowymi metodami nieuwzględniającymi kontekstu. Analiza wykazała szereg korzyści wynikających z zastosowania klasyfikatora kontekstowego. Uwzględnienie kontekstu pozwoliło na redukcję poziomu skomplikowania i złożoności klasyfikatorów oraz poprawę analizowanych wskaźników. Pracę zakończono podsumowaniem, przedstawieniem wniosków oraz zarysem kierunków dalszych badań.

## Rozdział 2

# Wybrane zagadnienia diagnostyki procesów przemysłowych

Diagnostyka procesów przemysłowych stanowi istotną gałąź diagnostyki technicznej, która zajmuje się wykrywaniem, rozróżnianiem i identyfikacją zmian stanów obiektów technicznych, którymi są między innymi procesy przemysłowe. Jedną z przyczyn występowania tych zmian są uszkodzenia, awarie oraz inne niepożądane zdarzenia (Korbicz *i in.*, 2004; Isermann, 1997). W rozdziale omówiono podstawowe zagadnienia związane z diagnostyką techniczną w szczególności skupiając się na tych, które są charakterystyczne dla diagnostyki procesów przemysłowych.

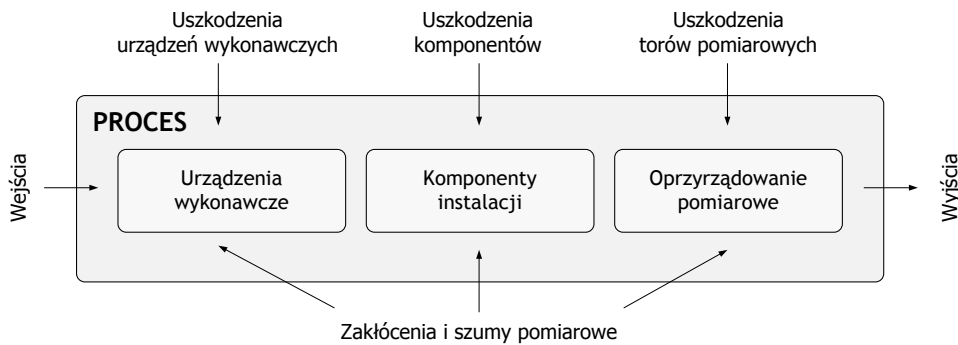
### 2.1. Pojęcia podstawowe

W diagnostyce technicznej, a co za tym idzie również w diagnostyce procesów przemysłowych stosuje się szereg pojęć i definicji, które pozwalają na czytelne opisanie elementów związanych z procesami przemysłowymi, jak i systemami opracowywanymi w celach diagnostycznych (Isermann i Balle, 1997). Terminologia ta została przyjęta przez wiele zespołów badawczych prowadzących zaawansowane badania w dziedzinie diagnostyki technicznej (Blanke *i in.*, 2006; Caccavale i Villani, 2003; Isermann, 2006; Korbicz *i in.*, 2004; Moczulski, 2002; Patton *i in.*, 2000). W kolejnych punktach niniejszego rozdziału omówiono najważniejsze pojęcia, związane z diagnostyką techniczną.

#### 2.1.1. Obiekt diagnozowania

Na rysunku 2.1 przedstawiono ogólny schemat obiektu diagnozowania używany w diagnostyce procesów przemysłowych (Blanke *i in.*, 2006; Isermann, 2005; Korbicz *i in.*, 2004). Głównym elementem schematu, dotyczącym interakcji pomiędzy obiektem, a otoczeniem, jest szereg wejść i wyjść. Poprzez wejścia istnieje możliwość oddziaływania na elementy składowe obiektu z zewnątrz, natomiast za pomocą wyjść obiekt jest w stanie oddziaływać na otoczenie. Oprócz wejścia jawnego na schemacie oznaczono również zbiór wejść niejawnych, które reprezentują zakłócenia i szумы pomiarowe oraz uszkodzenia elementów obiektu. Za pomocą czujników i innych urządzeń pomiarowych, istnieje

możliwość wyekstrahowania tych danych, a po ich interpretacji, informacji, z diagnozowanego obiektu w postaci sygnałów, które mogą posłużyć do wyznaczenia cech sygnałów, a w kolejnych krokach sygnału diagnostycznego. Otoczenie, w którym umieszczony jest obiekt diagnozowania, traktowany jest jako kontekst jego działania co zostało opisane szerzej w dalszej części pracy.



Rys. 2.1: Schemat obiektu diagnozowania (Kościelny, 2001)

### 2.1.2. Sygnał, cecha sygnału i sygnał diagnostyczny

Obserwacja oddziaływań pomiędzy otoczeniem i badanym obiektem jest prowadzona za pośrednictwem sygnałów. Sygnał to przebieg dowolnej wielkości fizycznej będącej nośnikiem informacji. W celu pozyskania tej informacji wyznacza się wartości wybranych cech sygnałów (np. wartość skuteczna), nazywanych zmiennymi procesowymi. Na podstawie zmiennych procesowych mogą być wyznaczane sygnały diagnostyczne, tzn. przebiegi dowolnych wielkości będących nośnikami informacji o stanie technicznym obiektu diagnozowania (Cholewa i Kaźmierczak, 1996). Cecha sygnału może zostać opisana jako uporządkowana trójka:

$$cl = \langle obj, nam(cl), val(cl) \rangle, \quad (2.1)$$

gdzie *obj* jest nazwą rozpatrywanego obiektu, *nam* dotyczy nazwy cechy sygnału a *val* zawiera wartość cechy sygnału. Pierwszy składnik zaproponowanej trójki jest najczęściej pomijany, przez co atrybut cechy sygnału ostatecznie można opisać w postaci uporządkowanej pary (Cholewa i Kaźmierczak, 1996):

$$cl = \langle nam(cl), val(cl) \rangle. \quad (2.2)$$

Wyróżnić można cechy własne jak np. wartość średnia, czy wartość średniokwadratowa, które są wyznaczone dla jednego przebiegu sygnału oraz cechy wzajemne, które są wyznaczone dla większej liczby przebiegów, przykładem takich cech są korelacje i kowariancje. Cechy mogą być także wyznaczane jako cechy jakościowe. Cechy liczbowe wyznaczane są, zależnie od zastosowania, w różnych dziedzinach, jak np. dziedzina czasu, częstotliwości lub amplitudy.

W wyniku działania algorytmu detekcji uszkodzeń, generowane są sygnały diagnostyczne, które mogą być reprezentowane w różny sposób (Korbicz *i in.*, 2004). Jedną z reprezentacji jest pojedynczy binarny sygnał diagnostyczny rozróżniający dwa stany diagnozowanego obiektu, „obiekt działa prawidłowo” lub „wykryto nieprawidłowe działanie obiektu”.

### 2.1.3. Fuzja danych

W literaturze można znaleźć wiele odwołań do fuzji jako podejścia pozwalającego na łączenie danych, które zapisywane są w różnej formie. Jednym z bardziej popularnych podejść jest fuzja obrazów, polegająca na połączeniu dwóch obrazów zaobserwowanych przez dwa urządzenia wizyjne, umieszczone w różnych punktach, ale obserwujących tą samą scenę (Jamrozik *i in.*, 2011). Obrazy te, niezależnie od różnic parametrów urządzeń wizyjnych, zawsze mają część wspólną. Analogicznie do fuzji obrazów, stosuje się fuzję danych, zapisanych zazwyczaj jako wektory lub macierze. W tym przypadku także, podstawą jest określenie części wspólnej zarejestrowanych danych. Oprócz fuzji danych (Esteban *i in.*, 2005) w literaturze opisuje się także fuzję informacji (Nakamura *i in.*, 2007) czy fuzję czujników (Bloch, 1996), które traktuje się jako synonimy. Jedno z podejść zastosowania fuzji danych, polega na podzieleniu fuzji na dwa etapy. W pierwszym etapie, nazywanym dopasowaniem, dane są dopasowywane pod względem reprezentacji. W drugim etapie, nazywanym agregowaniem, dane łączone są w celu uzyskania nowych danych posiadających pożądaną własność. W przypadku fuzji obrazów, stosuje się dopasowanie geometryczne obrazów pochodzących z różnych źródeł, zarejestrowanych w różnym czasie lub z różnego punktu widzenia (Jamrozik *i in.*, 2011).

Innym przykładem fuzji danych, który jest opisany szerzej w dalszej części pracy jest fuzja klasyfikatorów (Woźniak, 2006; Kuncheva, 2004). Fuzja klasyfikatorów polega na dopasowaniu odpowiedzi zespołu klasyfikatorów, pod względem typu i postaci zwracanych wartości, a następnie agregacji ujednoczonych odpowiedzi, w celu otrzymania wyniku klasyfikacji. Oczekuje się, że rezultatem działania fuzji klasyfikatorów jest odpowiedź, charakteryzująca się korzystniejszą wartością miary dokładności klasyfikacji (*ang. accuracy*) lub innych wskaźników dostępnych np. w wyniku analizy macierzy pomyłek tj. precyzja (*ang. precision*), czułość (*ang. recall*) (Kalisch, 2014) itd. Jednym z zagadnień, które jest stosowane równoległe do zadania klasyfikacji jest uwzględnienie kontekstu, który może być stosowany do budowania klasyfikatorów oraz zespołów klasyfikatorów.

### 2.1.4. Kontekst

Pojęcie kontekst (łac. contextus) funkcjonuje od bardzo dawna i związane jest z różnymi podejściami. Zasadniczo definicje kontekstu można podzielić na dwie grupy. Pierwsza grupa dotyczy podejścia kognitywistycznego (Kotelly, 1970; McCarthy, 1987; McCarthy, 1993) a druga inżynierskiego (Akman i Surav, 1996). Podejście kognitywistyczne dotyczy czynników, które należy rozważać wspólnie, jako czynniki współistniejące, powiązane z czymś, a kontekst jest używany do modelowania oddziaływań między obiektami

rozumianymi jako zdarzenia. Definicja kontekstu używana jest przede wszystkim w lingwistyce czy socjologii. W podejściu inżynierskim kontekst jest rozumiany jako określona wiedza, jako pewnego rodzaju filtr, który pozwala na przeprowadzenie procesu wnioskowania w oparciu o pewne fragmenty wiedzy zdefiniowane poprzez kontekst. W zakresie diagnostyki technicznej kontekst może np. stanowić wiedzę na temat otoczenia diagnozowanego obiektu (Turney, 2002a). W sytuacji gdy kontekst jest znany i jawny, można bezpośrednio uwzględnić go w procesie diagnostycznym lub np. zastosować metody odkrywania kontekstu jeżeli nie jest on znany (Timofiejczuk, 2011). Istnieje też podejście bazujące na utworzeniu klasyfikatora, którego zadaniem jest rozpoznanie kontekstu z danych wejściowych co umożliwia wykorzystanie tak uzyskanej informacji o kontekście w dalszym procesie wnioskowania (Kalisch, 2015).

Ze względu na fakt, że kontekst jest istotnym elementem opisanych badań, w dalszej części pracy poświęcono temu zagadnieniu cały rozdział aby dokładniej omówić idee kontekstu oraz to, w jaki sposób kontekst jest rozumiany przez autora pracy, w szczególności uwzględniając sposoby zastosowania kontekstu w detekcji uszkodzeń.

## 2.2. Zadania diagnostyki procesów przemysłowych

Ze względu na wzrost rozmiarów i złożoności obiektów technicznych, w tym procesów przemysłowych, rośnie również ryzyko występowania awarii oraz stanu technicznego, który umożliwia funkcjonowanie obiektu technicznego, ale cechuje się uszkodzeniami utrudniającymi lub uniemożliwiającymi poprawne działanie obiektów. Wynika to z faktu, że zwiększona złożoność diagnozowanych obiektów utrudnia identyfikację stanów niepoprawnych, które mogą prowadzić do awarii. Każdorazowe wystąpienie awarii lub konieczność cyklicznej oceny stanu poszczególnych urządzeń, w celu wykrycia nieprawidłowości, które docelowo mogą doprowadzić do awarii powoduje, że proces przemysłowy często ulega zatrzymaniu, co bezpośrednio wpływa na koszty utrzymania takiego procesu. Ponadto wzrost automatyzacji procesów przemysłowych wraz ze zmniejszeniem się liczby osób obsługi wpływa na zwiększenie liczby alarmów, co może doprowadzić do przeciążenia operatorów, a co za tym idzie wpłynąć na błędne decyzje podejmowane w krytycznych momentach (Kościelny, 2001). W tym celu opracowuje się systemy wspomaganie podejmowania decyzji, których zadaniem jest generowanie dla operatorów odpowiedzi dotyczących aktualnego stanu technicznego, jego genezy oraz charakteru awarii, wyjaśnieniu na podstawie jakich przesłanek system sugeruje określoną decyzję. Aby realizacja działania takiego systemu była możliwa, konieczna jest implementacja odpowiedniej metody, która pozwala na przeprowadzenie procesu wnioskowania na podstawie dostarczonych danych i informacji pochodzących z diagnozowanego procesu przemysłowego (Korbicz *i in.*, 2004). Wyróżnia się dwie podstawowe metody diagnostyki procesów przemysłowych:

- metoda wsparta modelami,
- metoda bazująca na symptomach.



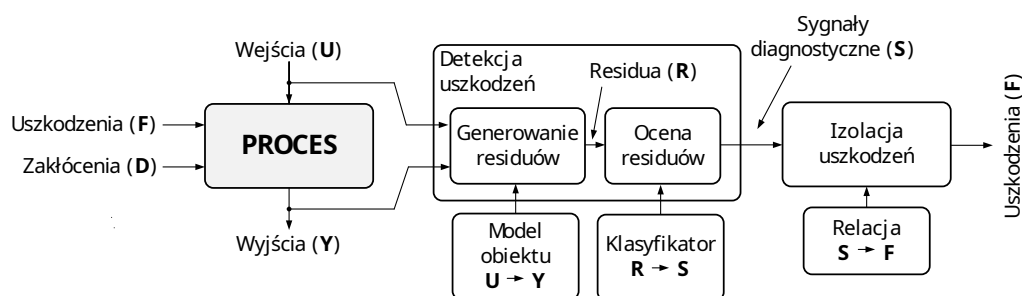
Bez względu na zastosowaną metodę, w zadaniu diagnozowania można wyróżnić trzy podstawowe fazy (Isermann i Balle, 1997):

- detekcja uszkodzenia (*ang. fault detection*) - wykrycie wystąpienia uszkodzenia w obiekcie oraz określenie chwili czasu, w której dokonano detekcji,
- izolacja uszkodzenia (*ang. fault isolation*) - występuje po detekcji uszkodzenia; obejmuje ustalenie rodzaju, miejsca oraz czasu trwania uszkodzenia; może być zastąpiona fazą rozpoznawania stanu lub klasy stanów obiektu,
- identyfikacja uszkodzenia (*ang. fault identification*) - występuje po fazie izolacji uszkodzenia, obejmuje określenie rozmiaru uszkodzenia, charakteru zmienności uszkodzenia w czasie oraz prognozowanie przyszłych stanów.

Wymienione trzy fazy składają się na diagnostykę uszkodzenia (*ang. fault diagnosis*). W kolejnych podrozdziałach rozprawy omówiono dwie podstawowe metody diagnostyki procesów przemysłowych wymienione wyżej.

### 2.2.1. Diagnostyka wsparta modelami

Na rysunku 2.2 przedstawiono schemat blokowy reprezentujący koncepcję procesu diagnozowania z zastosowaniem fazy detekcji i izolacji uszkodzeń. Na etapie detekcji uszkodzeń z wybranych cech sygnałów wejściowych i wyjściowych gromadzonych podczas monitorowania procesu oraz cech sygnałów uzyskiwanych za pomocą modelu obiektu wyznaczane są tzw. residua, na podstawie których wyliczane są sygnały diagnostyczne. Zazwyczaj residua rozumiane są jako sygnały będące różnicą pomiędzy rejestrowanymi wartościami zmiennych procesowych, a sygnałami otrzymywanymi z modelu (Blanke *i in.*, 2006; Korbicz *i in.*, 2004). Znane są jednak podejścia, w których residua określane są jako różnice pomiędzy wartościami nominalnymi współczynników fizycznych modelu wyznaczonego dla stanu zdadności, a wartościami bieżącymi współczynników modelu wyznaczonego w chwili diagnozowania (Isermann, 2005).



Rys. 2.2: Schemat blokowy procesu diagnozowania z fazą detekcji i izolacji uszkodzeń z zastosowaniem redundancji analitycznej lub informacyjnej (Korbicz *i in.*, 2004)

W metodzie bazującej na modelu, oprócz obiektu diagnozowanego konieczne jest również uwzględnienie samego modelu, który może być reprezentowany przez model numeryczny określony na bazie uczenia, czy wiedzy eksperckiej (Korbicz *i in.*, 2004). Do

najpopularniejszych modeli można zaliczyć np. modele analityczne, neuronowe i rozmyte (Przystałka, 2011), inne modele będące wynikiem metod uczenia maszynowego (Moczulski, 2002).

W podejściu bazującym na modelu procesu możliwe jest wyróżnienie trzech podstawowych sposobów generowania residuów (Isermann, 2005; Korbicz *i in.*, 2004). Pierwszy ze sposobów polega na utworzeniu modelu, którego rezultat działania jest zgodny z poprawnie działającym procesem. W celu utworzenia takiego modelu niezbędne jest pozyskanie danych, które zostały zebrane podczas nominalnego działania procesu. Jeżeli w trakcie działania procesu nie występuje żadne uszkodzenie, oczekuje się, że sygnały generowane przez model i sygnały odczytane bezpośrednio z procesu będą do siebie w dużym stopniu zbliżone i nie powinna występować między nimi znacząca różnica (wartości residuów będą bliskie zeru). W chwili gdy w monitorowanym procesie wystąpi uszkodzenie, sygnały pobrane z rzeczywistego procesu powinny wyraźnie zmienić swoją wartość w odniesieniu do wartości sygnałów generowanych przez model, a w konsekwencji powinno spowodować znaczące odchylenie wartości residuów od zera. Można podać wiele przykładów zastosowania tego podejścia w praktyce przemysłowej np. do detekcji wycieków w sieciach wodociągowych (Moczulski *i in.*, 2011), detekcji uszkodzeń układu wykonawczego pracującego w cukrowni (Bartyś *i in.*, 2006) i innych (Odgaard *i in.*, 2009).

Drugim podejściem generowania residuów jest zastosowanie wielu modeli diagnozowanego obiektu, do utworzenia których użyto dane pozyskane w trakcie działania procesu w stanie pełnej zdatności jak i w pozostałych stanach związanych z uszkodzeniami. Wynikiem takiego podejścia jest zbiór residuów generowanych równolegle dla wszystkich modeli. Wartości residuów, których wartości są bliskie zeru wskazują na wystąpienie stanu odpowiadającego danemu modelowi. To podejście wyraźnie ułatwia realizację etapu izolacji uszkodzeń, jednak pozyskanie odpowiedniej ilości danych stanowiących zbiór uczący dla modeli związanych z stanem z uszkodzeniem może stanowić znaczny problem (Korbicz *i in.*, 2004).

Nieco odmiennym sposobem generowania residuów jest zastosowanie identyfikacji online. Ten sposób wymaga budowy statycznego lub dynamicznego modelu (teoretycznego) reprezentującego obiekt oraz jego własności w stanie zdatności takie jak rezystancje, masy, sztywności, itp. W takim przypadku współczynniki określające własności fizyczne obiektu zawarte są w parametrach modelu obiektu. Jeśli na podstawie prowadzonej w czasie rzeczywistym identyfikacji modelu procesu określi się wartości tych współczynników i porówna z wartościami nominalnymi (tzn. dla modelu wyznaczonego dla stanu pełnej zdatności), to uzyskane różnice są różnicami niosącymi informację o uszkodzeniach (Isermann, 1993). To podejście również ułatwia fazę izolacji uszkodzeń.

Bez względu na rodzaj zastosowanej metody generowania residuów niezbędna jest odpowiednia ich ocena, która realizowana jest w części decyzyjnej podejścia pokazanego na schemacie 2.2. Stosuje się wiele różnych technik oceny residuów, z czego najważniejsze to (Chen, 1998):

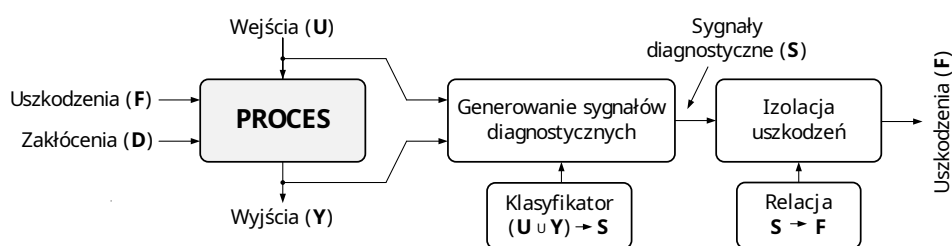
- metody z arbitralnie przyjmowaną wartością progową,
- metody adaptacyjne polegające na wyznaczaniu wartości progowej,

- metody z zastosowaniem oceny rozmytej residuów,
- inne.

Uzyskane w ten sposób sygnały diagnostyczne mogą zostać poddane kolejnym analizom ukierunkowanym na rozróżnienie uszkodzenia. Możliwe jest również podejście, w którym wnioskowanie diagnostyczne prowadzone jest bezpośrednio na residuach, co wymaga założenia tego, że dostępne są środki umożliwiające odwzorowanie przestrzeni wartości residuów (zazwyczaj wstępnie przetworzonych) w przestrzeń wartości sygnatur uszkodzeń lub stanów obiektu (Korbicz *i in.*, 2004).

### 2.2.2. Diagnostyka symptomowa

Diagnostyka symptomowa, zwana także bezpośrednią, stosowana jest w przypadkach nieznamośności lub nadmiernej złożoności modelu obiektu. Diagnozowanie bezpośrednie to postępowanie, w którym każda z trzech podstawowych faz procesu diagnozowania (detekcja, izolacja i identyfikacja uszkodzeń) prowadzona jest na podstawie sygnałów diagnostycznych wygenerowanych wyłącznie w wyniku kontroli i analizy podstawowych i nadmiarowych zmiennych procesowych (Korbicz *i in.*, 2004; Patton *i in.*, 2000). Najczęściej proces ten realizowany jest zgodnie ze schematem przedstawionym na rysunku 2.3. Schemat diagnostyki symptomowej jest zbliżony do schematu opisującego diagnozowanie z zastosowaniem modelu. Jedną z różnic dotyczy sposobu realizowania etapu detekcji uszkodzeń, gdzie w przypadku diagnostyki bezpośredniej nie są generowane residua, a zmienne procesowe są poddawane bezpośredniej analizie. Podobnie jak w przypadku diagnostyki wspartej modelem, wyjściem tego etapu są sygnały diagnostyczne. Zadaniem bloku, w którym następuje izolacja uszkodzeń, jest rozróżnienie uszkodzeń na podstawie analizy cech sygnałów diagnostycznych.



Rys. 2.3: Schemat blokowy procesu diagnozowania bezpośredniego z fazą detekcji i izolacji uszkodzeń (Korbicz *i in.*, 2004)

Niekiedy, zamiast fazy izolacji, stosuje się fazę rozpoznawania stanu obiektu lub klasy stanu. Zadanie takie można rozpatrywać jako problem klasyfikacji, w którym klasy są definiowane przez opisy symptomów świadczących o wystąpieniu danego stanu obiektu. Definicje symptomów mogą być formułowane na podstawie wiedzy specjalistów stosując sformalizowane metody pozyskiwania wiedzy (Moczulski, 2005). Mogą być także wynikiem zastosowania metod identyfikacji na podstawie czynnych i biernych eksperymentów diagnostycznych. W literaturze opisywane są przypadki stosowania podejścia łączącego

fazę detekcji i izolacji uszkodzeń. Wymaga to środków umożliwiających odwzorowanie przestrzeni wartości zmiennych procesowych (wstępnie przetworzonych) w przestrzeń wartości sygnatur uszkodzeń lub stanów obiektu (Cholewa *i in.*, 2008; Korbicz *i in.*, 2004).

Można podać wiele przykładów zastosowania tego podejścia w praktyce przemysłowej np. do detekcji uszkodzeń autonomicznej łodzi podwodnej (Ji *i in.*, 2021), detekcji i izolacji uszkodzeń w rozległym systemie cyber-fizycznym (Alippi *i in.*, 2017) i innych (Sheriff *i in.*, 2017).

## 2.3. Zadanie klasyfikacji stanu technicznego obiektu

W przypadku diagnostyki procesów przemysłowych zadanie klasyfikacji polega na przyporządkowaniu wektora cech sygnałów  $cI$ , odpowiadającego nieznanemu uszkodzeniu  $f$ , do znanej klasy  $f_i$ , reprezentującej uszkodzenie należące do przestrzeni  $F$  znanych uszkodzeń. Przypisanie wejściowego wektora cech do jednej ze znanych klas jest możliwe, dzięki zastosowaniu zbioru uczącego, który zawiera znane wzorce reprezentujące relację pomiędzy elementami wielowymiarowej przestrzeni symptomów  $S$ , a elementami przestrzeni uszkodzeń  $F$  (Cichosz, 2000; Kuncheva, 2004).

### 2.3.1. Klasyfikatory bazowe

W literaturze znanych jest wiele metod uczenia maszynowego pozwalających na budowanie klasyfikatorów. W rozdziale opisano wybrane metody klasyfikacji powszechnie stosowane w diagnostyce procesów przemysłowych (Chen *i in.*, 2004; Asadi Majd *i in.*, 2017; Chen *i in.*, 2016). Przykłady najbardziej znanych podejść, stosowanych w diagnostyce procesów przemysłowych do rozróżniania stanu pełnej zdatności i stanu z uszkodzeniem omówiono poniżej wraz ze wskazaniem najważniejszych pozycji literaturowych.

#### k - Najbliższych sąsiadów

Klasyfikacja bazująca na metodzie k-najbliższych sąsiadów (ang. k-Nearest Neighbour, kNN) to jedna z ważniejszych nieparametrycznych metod klasyfikacji (Krzyśko *i in.*, 2008). Przyporządkowanie klasyfikowanego obiektu do jednej ze znanych klas bazuje na wyznaczeniu podobieństwa pomiędzy tym obiektem, a zbiorem sklasyfikowanych obiektów wchodzących w skład danych trenujących. Podobieństwo w tym przypadku jest definiowane jako odległość pomiędzy klasyfikowanym obiektem, a pozostałymi obiektami w wielowymiarowej przestrzeni cech. Do określenia odległości pomiędzy obiektami stosowane są różne metryki (Deza i Deza, 2009). Do najczęściej używanych należą:

- Metryka Euklidesa:

$$D(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}, \quad (2.3)$$

- Metryka Manhattan:

$$D(x, y) = \sum_{i=1}^n |x_i - y_i|, \quad (2.4)$$

- Metryka Czebyszewa:

$$D(x, y) = \max_{i=1:n} (|x_i - y_i|). \quad (2.5)$$

gdzie  $D(x, y)$  odległość liczoną według określonej metryki pomiędzy punktami  $x$  i  $y$ .

Stopień pewności o poprawności końcowego wyniku klasyfikacji jest zależny od wartości parametru  $k$ . Dla parametru  $k$  równego 1 końcowa decyzja zawsze będzie równa 100% ponieważ brany pod uwagę jest tylko jeden najbliższy sąsiad bez konkurentów. W przypadku gdy wartość parametru  $k$  jest większa od 1 wtedy stopień pewności zależy od wyniku większościowego głosowania pomiędzy wszystkimi branymi pod uwagę najbliższymi sąsiadami. W przypadku remisu, czyli w momencie gdy taka sama liczba najbliższych sąsiadów wskazuje na dwie różne klasy, końcowa decyzja podejmowana jest poprzez losowanie (Verdier i Ferreira, 2011).

### Naiwny klasyfikator bayesowski

Klasyfikacja oparta na zastosowaniu naiwnego klasyfikatora bayesowskiego jest metodą bazującą na twierdzeniu Bayesa. W twierdzeniu tym zakłada się, że najbardziej prawdopodobną klasą, do której należy przypisać nowy obiekt, opisany danymi wartościami  $N$  atrybutów warunkowych  $X_{j,1} = x_{j,1}, X_{j,2} = x_{j,2}, \dots, X_{j,n} = x_{j,N}$  zapisanych w skrócie jako wektor  $\mathbf{x}$ , jest klasa  $d_i$ , która maksymalizuje prawdopodobieństwo warunkowe  $P(d_i | x_{j,1}, x_{j,2}, \dots, x_{j,N})$ . Klasa ta oznaczona jest jako "maximum a posteriori". Klasyfikator bayesowski szczególnie dobrze nadaje się do problemów o wielu wymiarach na wejściu. W przypadku naiwnego klasyfikatora bayesowskiego obliczenia warunkowego prawdopodobieństwa nie są oparte na dokładnym dopasowaniu aktualnej obserwacji do rekordów użytych w uczeniu, tak jak jest to w pełnym klasyfikatorze bayesowskim. W zamian dopasowuje je statystycznie do pełnego zbioru danych. Klasyfikator naiwny przyjmuje uproszczone (naiwne) założenie, że wobec niezależności zmiennych wejściowych prawdopodobieństwo warunkowe wielowymiarowe jest równe iloczynowi prawdopodobieństw jednowymiarowych wyliczonych na podstawie wszystkich zmiennych  $X_i$ . Przy założeniu równego prawdopodobieństwa wystąpienia wszystkich klas  $D_1, D_2, \dots, D_M$  maksymalizacja powyższego wskaźnika prawdopodobieństwa sprowadza się do operacji określonej wzorem:

$$\max P(D_i/\mathbf{x}) \rightarrow \max \prod_{i=1}^N P(X_j/D_i), \quad (2.6)$$

gdzie:  $P(D_i/\mathbf{x})$  oznacza prawdopodobieństwo warunkowe wystąpienia klasy  $D_i$  dla wektora atrybutów wejściowych  $\mathbf{x}$  a  $P(X_j/D_i)$  definiuje prawdopodobieństwo warunkowe, które estymuje się jako liczbę zgodności z badaną klasą odniesioną do liczby wystąpień danej klasy w zbiorze danych uczących.

Klasyfikator naiwny Bayesa można stosować zarówno przy atrybutach wejściowych nominalnych, jak i atrybutach typu numerycznego. W pierwszym przypadku odpowiednie prawdopodobieństwa  $P(X_j/D_i)$  są estymowane na podstawie bazy uczącej jako względna częstotliwość występowania próbek mających wartość prawdy atrybutu  $X_j$  w klasie  $D_i$ . W przypadku zmiennych  $X_j$  o wartościach rzeczywistych, zamiast wyznaczania kolejnych wartości numerycznych prawdopodobieństwa  $P(X_j/D_i)$  w  $i$ -tej klasie, estymuje się wartość  $P(x/D_i)$  poprzez zastosowanie wielowymiarowej funkcji Gaussa.

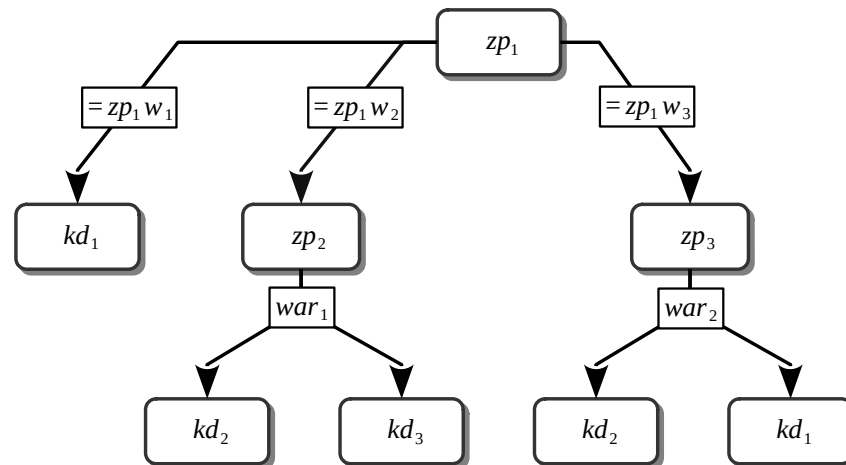
Jednym z przykładów stosowania metod bazujących na twierdzeniu Bayesa jest szkieletowy system doradczy MMNET (Cholewa, 2010). Rozwiązanie to opiera się na sieciach przekonań, zwanych również sieciami bayesowskimi, która jest niezawierającym cykli grafem skierowanym, którego węzły reprezentują zmienne, identyfikujące hipotezy oraz obserwacje, a skierowane gałęzie reprezentują związki pomiędzy tymi zmiennymi. Określenie tych relacji realizowane jest poprzez przypisanie węzłom sieci, tablic prawdopodobieństw warunkowych. Zbiór tablic prawdopodobieństw warunkowych oraz prawdopodobieństw a priori umożliwia wyznaczenie wartości prawdopodobieństwa łącznego, a więc cała sieć reprezentuje w oszczędny sposób łączny rozkład prawdopodobieństwa dla wszystkich zmiennych (Cholewa i Skupnik, 2011; Jensen i Nielsen, 2007). W literaturze można też znaleźć wyniki badań, których celem było zastosowanie naiwnego klasyfikatora bayesowskiego w celu detekcji uszkodzeń w diagnostyce procesów czy maszyn jak np. detekcja uszkodzeń elementów tocznych łożysk (Hou *i in.*, 2017), detekcja uszkodzeń pompy odśrodkowej z zastosowaniem transformacji falkowej (Muralidharan i Sugumaran, 2012) czy detekcja uszkodzeń transformatorów dużej mocy poprzez analizę gazów rozpuszczonych w oleju (Mahamdi *i in.*, 2022).

## Drzewo decyzyjne

Innym popularnym typem klasyfikatora są drzewa decyzyjne (Osowski, 2013; Krzyśko *i in.*, 2008). Drzewem decyzyjnym (klasyfikacyjnym) określa się drzewo reprezentujące proces podziału zbioru obiektów na klasy. W drzewie decyzyjnym wewnętrzne węzły opisują sposób dokonania przyporządkowania obiektów do danych klas (dokonywany w oparciu o wartości cech obiektów). Węzły rozmieszczone na końcach gałęzi odpowiadają klasom, do których obiekty mogą należeć, a gałęzie drzewa to wartości cech, na podstawie których klasyfikacja jest przeprowadzana. Drzewo „rośnie” od góry do dołu, czyli od korzenia do liści. Wyróżnia się dwa podstawowe rodzaje drzew klasyfikacyjnych:

- drzewa binarne, w których z każdego węzła mogą być wyprowadzone jedynie dwie gałęzie,
- drzewa niebinarne, w których z każdego węzła można wyprowadzić więcej niż dwie gałęzie.

Na rysunku 2.4 przedstawiono przykład struktury drzewa decyzyjnego niebinarnego. Węzły decyzyjne ( $zp$ ) w widocznym modelu przypisane są do określonych zmiennych procesowych. W pokazanym przykładzie założono, że zmienna procesowa  $zp_1$  jest zmienną



Rys. 2.4: Stryktura modelu klasyfikatora drzewa decyzyjnego

dyskretną przyjmującą trzy znane wartości ( $zp_1w_1, zp_1w_2, zp_1w_3$ ). W zależności od aktualnie przypisanej wartości do zmiennej procesowej  $zp_1$  aktywowany jest inny fragment drzewa i realizowane są kolejne warunki. Na rysunku 2.4 jedna z wartości zmiennej  $zp_1$  ( $cp_1w_1$ ) kończy realizację zaprezentowanego modelu wnioskowania końcową decyzją  $kd_1$ . W przypadku dwóch pozostałych sprawdzane są kolejne warunki dotyczące dwóch ciągłych zmiennych procesowych  $zp_1$  i  $zp_2$ . Dla każdej zmiennej zdefiniowany jest warunek ( $war_1, war_2$ ) określający wartość progową decyzji. Zakończeniem każdej gałęzi w drzewie jest końcowa decyzja  $kd$ .

Dane przetwarzane za pomocą algorytmu klasyfikatora są analizowane wzdłuż kolejnych gałęzi przez poszczególne węzły. W węźle dokonywany jest podział na podgrupy elementów, które są danymi wejściowymi tego węzła. Podział ten jest dokonywany zgodnie z zapisanym w węźle warunkiem. Analiza trwa do momentu, osiągnięcia przez element liścia odpowiadającego jednej z klas. Drzewa klasyfikacyjne są źródłem reprezentacji wiedzy, która jest bardzo łatwa do interpretacji przez człowieka. Zbiór wszystkich ścieżek może być przekształcony do zbioru reguł (najczęściej przy pomocy koniunkcji warunków elementarnych), klasyfikujących przykłady w sposób identyczny jak drzewo, na podstawie którego wyznaczono te reguły. Algorytmy stosowane do konstrukcji drzew klasyfikacyjnych należą do grupy algorytmów zachłanych. Drzewo decyzyjne jest generowane w sposób rekurencyjny, np. techniką „z góry na dół”, w sposób „dziel i rządź”. Najczęściej stosowanymi algorytmami, są: ID3, C4.5, CART oraz CHAID. Różni je przede wszystkim przyjęte kryterium podziału, definiujące sposobu tworzenia nowych węzłów w drzewie (Osowski, 2013). Jednym z powszechnie stosowanych kryteriów jest współczynnik Ginniego (ang. Gini index) definiowany dla zbioru danych  $E$  w postaci:

$$I(E) = \sum_{k=1}^M r_k^2, \quad (2.7)$$

gdzie  $M$  oznacza liczbę klas, a  $r_k$  proporcję wystąpień  $k$ -tej klasy w zbiorze  $E$ . Innym

wskaźnikiem rozkładu klasy decyzyjnej jest miara entropijna, definiowana dla pojedynczego zbioru  $E$  wzorem:

$$Ent(E) = - \sum_{k=1}^M r_k \log_2 r_k. \quad (2.8)$$

Jeżeli analizowany zbiór danych zawiera jedynie dane należące do jednej klasy, wówczas obie miary, zarówno Giniego, jak i entropijna są równe zero.

Kryterium podziału powinno minimalizować możliwość wystąpienia błędnej klasyfikacji, jednak należy też mieć na uwadze zagadnienie optymalnej wielkości drzewa klasyfikacyjnego (Krzyśko *i in.*, 2008). Utworzenie drzewa decyzyjnego, w którym wszystkie wektory z próby uczącej są doskonale klasyfikowane przez drzewo na poziomie 100% skuteczności klasyfikacji (każdy węzeł będący liściem drzewa jest związany tylko z wektorami danych należącymi do tej samej klasy) powoduje to, że tak utworzony klasyfikator może charakteryzować się znacznie mniejszą skutecznością klasyfikacji nowych przypadków, co negatywnie wpłynie na skuteczność klasyfikacji wytrenowanego wzorca. Wybór optymalnej wielkości drzewa klasyfikacyjnego jest równoważny z podaniem reguły decydującej o tym, czy dany węzeł ma podlegać podziałowi, czy ma być liściem drzewa. Jednym z rozwiązań jest wprowadzenie tzw. reguły stopu. Jest to najprostszą regułą tego typu, która przerywa proces dalszego dzielenia węzła  $t$ , jeżeli liczba przyporządkowanych mu obserwacji ze zbioru uczącego jest odpowiednio mała:

$$n(t) < n_{min}. \quad (2.9)$$

gdzie  $n_{min}$  jest ustaloną wartością progową. Inną powszechnie stosowaną metodą optymalizacji wielkości drzewa jest jego przycinanie. W pierwszym kroku tworzone jest drzewo maksymalne, którego gałęzie następnie są selektywnie redukowane poprzez usuwanie podziałów, które nie mają istotnego znaczenia dla poprawności klasyfikacji. Jedną z metod jest rekurencyjne usuwanie kolejnych węzłów przy każdorazowym badaniu błędu klasyfikacji i zaakceptowanie takiej struktury drzewa, która daje maksymalny akceptowalny błąd klasyfikacji jednocześnie przy minimalnej złożoności struktury drzewa.

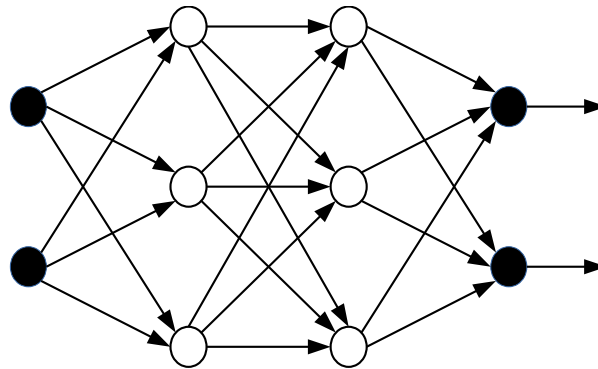
Każda z wymienionych wcześniej metod automatycznego budowania drzew decyzyjnych znajduje swoje zastosowanie w systemach detekcji uszkodzeń. W pracy (Benkercha i Moulahoum, 2018) opisano zastosowanie klasyfikatora drzewa decyzyjnego opartego o algorytm C4.5. W publikacji (Lipinski *i in.*, 2020) do detekcji uszkodzeń przekładni planetarnej użyto między innymi klasyfikatora drzewa decyzyjnego opartego o algorytm CART stosując zarówno kryterium podziału bazujące na współczynniku Ginniego jak i entropii. Innym przykładem zastosowania drzew decyzyjnych w diagnostyce technicznej jest detekcja uszkodzeń linii przesyłowych wysokiego napięcia (Samantaray *i in.*, 2011).

### Sieć perceptronu wielowarstwowego

Jedną z popularnych metod klasyfikacji, jest zastosowanie sieci neuronowych. Najprostszym rodzajem sieci są jednokierunkowe sieci MLP (ang. *MultiLayer Perceptron*). Ten



rodzaj sieci został szczegółowo omówiony w literaturze (Osowski, 2006; Osowski, 2013) i powszechnie stosowany w diagnostyce przemysłowych. Jedną z metod zastosowania sztucznych sieci neuronowych w diagnostyce procesów jest diagnostyka oparta na modelu badanego obiektu i wyznaczaniu residuów pomiędzy rzeczywistymi sygnałami i sygnałami generowanymi przez model (Korbicz *i in.*, 1994). Jednak sztuczne sieci neuronowe są również z powodzeniem wykorzystywane jako algorytmy nadzorowanego uczenia maszynowego w zadaniach klasyfikacji (Han *i in.*, 2012). Na rysunku 2.5 przedstawiono przykład struktury perceptronu wielowarstwowego z dwoma warstwami ukrytymi.



Rys. 2.5: Struktura perceptronu wielowarstwowego

Neurony ułożone w warstwach połączone są tylko między kolejnymi warstwami, natomiast sygnał przekazywany jest w jednym kierunku od wejścia do wyjścia. Poszczególne neurony sieci realizują odwzorowanie nieliniowe, które w przypadku  $i$ -tego neuronu  $k$ -tej warstwy można opisać wyrażeniem:

$$y_i^{(k)} = f(u_i^{(k)}) = f\left(\sum_j w_{ij} y_j^{(k-1)}\right), \quad (2.10)$$

gdzie  $\mathbf{w}$  jest wektorem wag zawierającym wagi  $w_{ij}$  dla każdego z wejść neuronu. Funkcja aktywacji  $f(u)$  w warstwie ukrytej sieci perceptronowej jest stosowana np. funkcja sigmoidalna. Stosuje się dwa rodzaje funkcji sigmoidalnej (Krzyśko *i in.*, 2008):

- unipolarna -  $f(u) = \frac{1}{1+e^{-\beta u}}$ ,
- bipolarna -  $f(u) = \tanh(\beta u) = \frac{1-e^{-\beta u}}{1+e^{-\beta u}}$ .

gdzie  $\beta$  jest współczynnikiem liczbowym (zwykle o wartości równej 1).

Najważniejszym etapem działania sieci neuronowej jest jej uczenie (trenowanie). Można wyróżnić dwie główne metody uczenia: uczenie nadzorowane i nienadzorowane. W przypadku uczenia nadzorowanego konieczne jest posiadanie danych trenujących, dla których istnieje zbiór prawidłowych odpowiedzi, które powinien wygenerować model sieci neuronowej. Proces uczenia polega na porównywaniu odpowiedzi sieci neuronowej z zakładanymi wartościami. Różnica jest traktowana jako błąd, który służy do dostrojenia przyjętych wag w poszczególnych neuronach. Jedną z metod uczenia jest metoda najszybszego spadku bazująca na wstecznej propagacji błędów (Osowski, 2006). Iteracyjne

działanie metody pozwala w kolejnych krokach minimalizować błąd pomiędzy odpowiedzią modelu, a wartościami docelowymi aż do momentu, w którym różnica nie będzie wykraczać poza dozwolony (akceptowalny) zakres.

W przypadku uczenia nienadzorowanego nie podaje się prawidłowej odpowiedzi modelu. Przykładem jest metoda uczenia bazująca na regule Hebba, w której algorytm uczący można zaliczyć do uczenia korelacyjnego, w którym siła połączenia między neuronowego wzrasta przy istnieniu korelacji między sygnałami postsynaptycznymi i presynaptycznymi neuronu (Krzyśko *i in.*, 2008).

W diagnostyce technicznej powszechnie stosuje się sztuczne sieci neuronowe. Jedną z form ich wykorzystania jest opracowanie modeli diagnozowanych obiektów w celu realizacji zadania diagnostyki wspartej modelem. Sztuczna sieć neuronowa również bardzo dobrze sprawdza się jako klasyfikator stosowany w podejściu bazującym na diagnostyce symptomowej (Jamil *i in.*, 2015). Znane są również podejścia, w których sztuczna sieć neuronowa jest stosowana zarówno jako model generujący wartości sygnałów symulowanego obiektu jak i jako klasyfikator, który na podstawie obliczanych residuów generuje sygnał diagnostyczny (Korbicz, 1997).

### 2.3.2. Systemy wieloklasyfikatorowe

Od momentu rozpoczęcia prac nad algorytmami uczenia maszynowego opracowano wiele różnych metod opierających się na odmiennych koncepcjach i podejściach. W każdej z tych metod można wyszczególnić pewne zalety ich działania jak i wady. Jednym z elementów, który może mieć kluczowe znaczenie dla skuteczności działania danej metody jest zbiór danych trenujących. Wykorzystanie klasyfikatorów różnych typów może przynieść odmienne rezultaty dla tego samego zbioru danych. Wybór optymalnej metody najczęściej jest bardzo trudny. Dlatego znane są w literaturze koncepcje polegające na budowaniu komitetów składających się z co najmniej dwóch klasyfikatorów. W literaturze dostępne są wyniki badań udowadniające, że podejście bazujące na tworzeniu systemów wieloklasyfikatorowych pozwala na uzyskanie lepszych wyników niż w przypadku pojedynczego klasyfikatora (Kuncheva, 2004).

Zasadniczym elementem systemów wieloklasyfikatorowych jest sposób wspólnego rozpatrywania klasyfikatorów. Jednym z podejść jest fuzja klasyfikatorów (Woźniak, 2006) polegająca na stosowaniu jako ich elementów bazowych tzw. klasyfikatorów słabych, czyli takich, dla których wynik klasyfikacji jest niewiele lepszy od decyzji losowej. Innym przypadkiem jest stosowanie klasyfikatorów niestabilnych, dla których niewielka zmiana w zbiorze danych może w znacznym stopniu wpłynąć na skuteczność klasyfikacji. Do klasyfikatorów niestabilnych można zaliczyć np. drzewa decyzyjne czy sieci neuronowe. Natomiast do klasyfikatorów mocnych można zakwalifikować np. naiwny klasyfikator bayesowski. Zastosowanie w tych metodach klasyfikatorów mocnych, które uzyskują wysoki wynik skuteczności klasyfikacji, może skutkować wygenerowaniem wielu bardzo podobnych klasyfikatorów, a końcowy wynik ich działania może być zbliżony do wyniku działania pojedynczego klasyfikatora o najwyższej skuteczności (Woźniak, 2006).

### Fuzja wyników klasyfikacji

Jednym z podejść dotyczących łączenia wielu klasyfikatorów jest fuzja polegająca na użyciu szeregu klasyfikatorów wchodzących w skład komitetu  $\Phi$ , z których końcowy rezultat zostaje wyodrębniony na podstawie głosowania. Ta metoda pozwala na zastosowanie zespołu klasyfikatorów zupełnie różniących się od siebie pod względem takich fundamentalnych aspektów jak ogólna metoda klasyfikacji (np.  $k$  najbliższych sąsiadów, naiwny klasyfikator bayesowski, sztuczna sieć neuronowa, itp.), algorytmy o podobnej idei działania jednak różniące się implementacją i sposobem interpretacji modelu (np. drzewa decyzyjne i klasyfikatory bazujące na regułach) czy klasyfikatory dokładnie tego samego typu ale wykorzystujące różne zbiory parametrów działania. Wyróżnić można różne metody głosowania, jak jednogłośnie, ważone czy większościowe. Głosowanie jednogłośnie wymaga zgodnej decyzji spośród wszystkich klasyfikatorów wchodzących w skład komitetu, co może w znacznym stopniu utrudnić klasyfikację, zwłaszcza w przypadku większej liczby dostępnych klas. Głosowanie większościowe polega na wybraniu klasy o największej liczbie głosów spośród wszystkich biorących udział w głosowaniu klasyfikatorów. W przypadku  $k$  klasyfikatorów  $\phi_1, \phi_2, \dots, \phi_k$  dwuklasowych (gdzie  $k$  jest nieparzyste) i założeniu, że klasy przyjmują wartości  $-1$  lub  $1$ , końcowa decyzja jest podejmowana za pomocą następującego wzoru:

$$kd = \operatorname{sgn}\left(\sum_{i=1}^k \phi_i\right) \quad (2.11)$$

W przypadku głosowania ważonego każdy z klasyfikatorów ma przypisaną wagę, gdzie waga klasyfikatorów o większej skuteczności klasyfikacji jest wyższa w stosunku do klasyfikatorów o niższej skuteczności klasyfikacji. Wagi są najczęściej stosowane w metodach fuzji klasyfikatorów takich jak Bagging i Adaboost, które zostały opisane w dalszej części rozdziału, gdzie przypisywane są w sposób automatyczny w zależności od skuteczności klasyfikacji poszczególnych klasyfikatorów. W przypadku większej liczby klas wynikowych, rezultat jest uzyskiwany na podstawie losowania spośród konfliktowych klas. Reguła decyzyjna w przypadku  $k$  klasyfikatorów  $\phi_1, \phi_2, \dots, \phi_k$  podejmujących decyzję o numerze należącym do zbioru  $-l, l$  oraz ich wag  $w_1, w_2, \dots, w_i$  może być zapisana następująco:

$$kd = \operatorname{sgn}\left(\sum_{i=1}^k w_i \phi_i\right) \quad (2.12)$$

Kolejnym podejściem w fuzji klasyfikatorów są metody ingerujące w zbiór danych trenujących. W tej koncepcji uznano, że przyczyną zbyt niskiej skuteczności klasyfikacji nie jest rodzaj wybranego klasyfikatora, a dane znajdujące się w zbiorze uczącym, których rozkład w przestrzeni cech utrudnia poprawną klasyfikację. Podstawową metodą jest tu metoda Bagging. Metoda ta polega na utworzeniu szeregu klasyfikatorów opartych na tym samym sposobie działania, ale dla każdego z nich przygotowany jest inny zestaw danych o rozmiarze równym liczebności zbioru danych źródłowych. Dane dla każdego z klasyfikatorów są losowane spośród danych wejściowych, przy czym każda

z wartości w całym procesie losowania ma takie samo prawdopodobieństwo wylosowania. Skutkiem takiego działania jest możliwość wielokrotnego powtarzania się tych samych danych w pojedynczym zbiorze trenującym. Na podstawie tej metody opracowane zostały kolejne algorytmy, określane mianem Boostingu, do których zaliczany jest algorytm AdaBoost (Yao *i in.*, 2012). Ta metoda fuzji klasyfikatorów dopasowuje się do błędów popełnianych przez poprzednie klasyfikatory w komitecie. Każda z wartości w zbiorze trenującym ma przypisaną wagę, która określa istotność danej próbki. Po nauczaniu pierwszego klasyfikatora na losowym zbiorze danych weryfikowana jest skuteczność klasyfikacji na tym samym zbiorze. Dane trenujące, dla których wynik działania klasyfikatora był błędny, w następnej iteracji mają wyższą wagę, co przekłada się na większe prawdopodobieństwo ich wylosowania. Dzięki temu kolejne klasyfikatory są trenowane na zbiorze wypełnionym w większym stopniu danymi, które są trudne w analizie i interpretacji. Dodatkowym elementem działania algorytmu jest nadanie wag dla poszczególnych klasyfikatorów, które są uwzględnione w procesie końcowego wnioskowania (głosowanie wazone).

### Metaklasyfikatory

Metaklasyfikacja polega na równoległym zastosowaniu wielu, często różnych, klasyfikatorów (również na podstawie innych danych wejściowych) i następnie na połączeniu wyników wnioskowania początkowych klasyfikatorów w końcowym klasyfikatorze. Głównym celem takiego podejścia jest uzyskanie lepszej dokładności klasyfikacji w porównaniu do pojedynczych klasyfikatorów wejściowych. Przykładem zastosowania takiego podejścia jest budowa komitetu klasyfikatorów binarnych, których zadaniem jest rozpoznawanie pojedynczej konkretnej klasy spośród puli wszystkich klas, których liczba jest większa niż 2. Takie podejście można przedstawić zależnością:

$$kd_i^k = \begin{cases} y_i = f \\ y_i \neq f \end{cases} \quad i = 1, 2, \dots, N. \quad (2.13)$$

gdzie  $kd_i^k$  jest  $i$ -tą decyzją klasyfikatora binarnego wytrenowanego do detekcji klasy  $f$ , która może być równa  $f$  lub różna od  $f$ . Wyniki tych klasyfikatorów są podawane na wejście kolejnego, niebinarnego klasyfikatora, którego zadaniem jest podjęcie końcowej decyzji na podstawie wyników działania bazowych klasyfikatorów. W literaturze takie podejście opisywane jako komitet  $K$ -klasyfikatorów (Jankowski, 2014) (gdzie  $K$  oznacza liczbę klasyfikatorów równą liczbie rozpatrywanych klas). Inny wariant tej metody występujący pod nazwą komitet  $K^2$ -klasyfikatorów bazuje na koncepcji tworzenia komitetu klasyfikatorów binarnych rozróżniających jedynie dwie konkretne klasy. Oznaczenie  $K^2$  może sugerować, że liczba klasyfikatorów w komitecie rośnie kwadratowo w stosunku do liczby klas, w rzeczywistości po usunięciu powtarzających się kombinacji par klas uzyskamy komitet zawierający dokładnie  $\binom{K}{2}$  klasyfikatorów.

W pracy (Kalisch, 2014) autor przedstawił wyniki analizy dotyczącej porównania sku-

teczności detekcji i izolacji uszkodzeń, przy pomocy algorytmów uczenia maszynowego i różnych metod budowania komitetów klasyfikatorów, w tym z zastosowaniem metaklasifikatora, na przykładzie danych symulacyjnych wygenerowanych za pomocą modelu numerycznego zaworu elektro-pneumatycznego opracowanego w ramach projektu DAMA-DICS (Bartyś *i in.*, 2006). Inne podejścia omówione w tym rozdziale dotyczące zagadnień fuzji danych i klasyfikatorów również są powszechnie wykorzystywane w symptomowej diagnostyce procesów. Przykładem mogą być publikacje (Kapucu i Cubukcu, 2021; Yang i Ismail, 2022; Dhibi *i in.*, 2021), w których autorzy zastosowali metody łączenia klasyfikatorów w celu detekcji uszkodzeń w instalacjach fotowoltaicznych.

### 2.3.3. Metody oceny klasyfikatorów

W celu estymacji skuteczności klasyfikacji danego klasyfikatora konieczne jest przeprowadzenie jego walidacji. Ograniczenia wynikające z liczebności danych dostępnych w procesie trenowania wymagało opracowania metod, które już przy niewielkim zbiorze danych pozwalały na ocenę klasyfikatora. Wymieniono powszechnie stosowane metody wraz z z krótkim opisem (Jain *i in.*, 2000; Kuncheva, 2004).

- Metoda resubstytucji - Metoda ta polega na wytrenowaniu klasyfikatora i późniejszym przetestowaniu go za pomocą tego samego zbioru danych. Podejście to pozwala na przeprowadzenie weryfikacji dla zbioru danych zawierających bardzo niewielką liczbę przykładów, jednak istnieje duże zagrożenie, że otrzymana estymata będzie zawyżona w stosunku danych rzeczywistych.
- Metoda wydzielenia - W celu realizacji tego podejścia konieczne jest losowe podzielenie zbioru danych na dwa niezależne podzbiory, gdzie pierwszy jest wykorzystany do trenowania a drugi do testowania klasyfikatora. Oba podzbiory powinny mieć podobną liczebność, co jednak na znaczną redukcję liczby przykładów trenujących, wynik estymacji skuteczności klasyfikacji może być zaniżony. Metodę tę można powtórzyć kilkakrotnie, aby uniezależnić wyniki estymacji od podziału zbioru przykładów.
- Metoda walidacji krzyżowej - Metoda polega na podzielenie całego zbioru danych  $K$  podzbiorów, gdzie  $K$  określa liczbę powtórzeń operacji trenowania i testowania klasyfikatora.  $K - 1$  podzbiorów jest traktowana jako zbiór treningowych a ostatni niewykorzystany podzbiór jako zbiór testowy. Proces jest powtarzany  $K$  razy, gdzie za każdym razem inny podzbiór jest testowym a pozostałe pełnią rolę treningowych. Końcowa estymata jest uśrednieniem uzyskanych wyników. Błąd estymacji tej metody jest stosunkowo niski (tym niższy, im mniejsze jest  $K$ ).
- Metoda  $N$ -krotnej walidacji krzyżowej (*ang. Leave one out*) - Metoda jest szczególnym przypadkiem metody walidacji krzyżowej, w której parametr  $K$  jest równy  $N$ , gdzie  $N$  określa liczebność zbioru danych. W trakcie działania tej metody, zbiór treningowy składa się z  $N - 1$  próbek a ostatnia nie wykorzystana próbka danych jest próbką testującą. Metoda ta jest zalecana w sytuacji, gdy dostępny

zbiór danych charakteryzuje się niską liczebnością próbek. Wariancja skuteczności klasyfikacji tej metody jest wyższa niż przy metodzie walidacji krzyżowej.

Poprzez zastosowanie jednej z dostępnych metod estymacji błędu klasyfikatora, jak np. walidacja krzyżowa, możliwe jest uzyskanie macierzy pomyłek, w której zawarte są liczby poprawnie i błędnie sklasyfikowanych stanów. Na podstawie macierzy pomyłek możliwe jest wyznaczenie szeregu miar oceniających skuteczność klasyfikatora.

Tab. 2.1: Macierz pomyłek dla przykładu klasyfikacji binarnej

Stan rzeczywisty	Stan przewidywany	
	Klasa 1	Klasa 2
Klasa 1	Prawdziwie pozytywny (TP)	Fałszywie negatywny (FN)
Klasa 2	Fałszywie pozytywny (FP)	Prawdziwie negatywny (TN)

Tak zdefiniowana macierz pomyłek pozwala na zastosowanie szeregu miar statystycznych, poniżej przedstawiono listę najczęściej wykorzystywanych miar.

- Dokładność (Accuracy):

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.14)$$

- Czułość (True positive rate):

$$TPR = \frac{TP}{TP + FN} \quad (2.15)$$

- Swoistość (True negative rate):

$$TNR = \frac{TN}{FP + TN} \quad (2.16)$$

- Wskaźnik fałszywie pozytywnych (False positive rate)

$$FPR = \frac{FP}{FP + TN} \quad (2.17)$$

- Wskaźnik fałszywie negatywnych (False negative rate)

$$FNR = \frac{FN}{FP + FN} \quad (2.18)$$

Miary te mają swoje przełożenie na zagadnienia diagnostyki np. medycznej ale też diagnostyki technicznej w tym procesów, co pozwala na bezpośrednie ich wykorzystanie do oceny skuteczności rozpoznawania uszkodzeń w diagnostyce procesów technicznych. W tym celu stosuje się np. wskaźniki fałszywych i prawdziwych alarmów (Bartyś *i in.*, 2006), które szerzej zostały omówione w rozdziale 4.

Wymienione miary mogą być również elementem bardziej rozbudowanego systemu diagnostycznego składającego się z wielu klasyfikatorów pracujących w komitecie lub mogą być też wykorzystane do zbudowania mechanizmu automatycznego dostrajania parametrów klasyfikatorów w procesie metauczenia z wykorzystaniem algorytmów optymalizacji, co zostało opisane w dalszej części pracy.

## 2.4. Optymalizacja w diagnostyce

Jednym z problemów związanych ze stosowaniem algorytmów uczenia maszynowego jest zastosowanie odpowiedniej metody klasyfikacji i dostosowanie jej parametrów. Każda z powszechnie dostępnych metod charakteryzuje się określonym zbiorem parametrów mających bezpośredni wpływ na proces ich trenowania oraz na uzyskiwane wyniki. Dobór parametrów jest też zależny od charakterystyki danych trenujących, a w kolejnym etapie nowych danych, na podstawie których gotowy wytrenowany model dokonuje klasyfikacji. Parametry mogą zostać dobrane na podstawie tzw. wiedzy eksperckiej, gdzie osoba posiadająca wiedzę na temat działania algorytmu dobiera parametry bazując na swojej wiedzy i doświadczeniu. Istnieje możliwość automatyzacji procesu doboru parametrów poprzez zastosowanie jednej z dostępnych metod jak np. metoda systematycznego przeszukiwania, która jednak ze względu na swój charakter jest metodą niezwykle kosztowną i czasochłonną. Innym podejściem może być zastosowanie jednej z dostępnych metod optymalizacji.

Problem optymalizacji jest znany od bardzo dawna i jest widoczny w problemach poszukiwania najkrótszej drogi, maksymalnego pola (Stachurski i Wierzbicki, 1999), itp. Wiele rozwiązań dotyczących podejmowanych problemów wymaga zastosowania optymalizacji, w której uwzględnia się wiele kryteriów. Optymalizację wielokryterialną (której rezultatem jest zbiór wartości optymalnych) najprościej przedstawić w odniesieniu do optymalizacji jednokryterialnej, w której wynikiem zastosowania funkcji celu jest jedna wartość (Marler i Arora, 2004; Kalisch, 2018). W tym celu zastosować można metodę sumy ważonej, w której wyniki poszczególnych funkcji celu problemu wielokryterialnego mnożone są razy indywidualnie przydzielone wagi, a następnie sumowane do pojedynczej wartości. Jednak tak skonstruowany problem ogranicza się do jednego rozwiązania, które zostało osiągnięte poprzez znalezienie minimum funkcji celu. Istnieje jednak ryzyko, że znalezione minimum jest minimum lokalnym, a w innym miejscu szukanego obszaru znajduje się minimum globalne, które nie zostało znalezione przez algorytm. Przykładem takiego zadania jest np. kupno samochodu, gdzie pod uwagę bierze się takie kryteria jak cena, wyposażenie, moc silnika, zużycie paliwa, itp. W przedstawionym przykładzie z reguły nie będzie możliwe znalezienie pojedynczego rozwiązania optymalnego, dla którego wszystkie kryteria będą optymalne. Dzieje się tak ponieważ w większości przypadków kryteria nie są ze sobą zgodne (minimalizacja jednego z nich może powodować wzrost innych), sytuacją taką charakteryzuje się jako dominowanie jednego kryterium nad innym, co można zapisać symbolicznie w następujący sposób:

$$x \succ y \Leftrightarrow \exists i f_i(x) \leq f_i(y) \quad (2.19)$$

Rozwiązanie  $x$  dominuje nad rozwiązaniem  $y$  wtedy i tylko wtedy, gdy wartość funkcji celu dla  $x$  nie jest większa niż dla rozwiązania  $y$  w przypadku minimalizacji funkcji celu. W wyniku tego założenia wprowadzono podział możliwych rozwiązań zadania optymalizacji wielokryterialnej na zdominowane i niezdominowane.

Rozwiązanie jest niezdominowane (paretooptymalne) wówczas, gdy nie jest możliwe

znalezienie rozwiązania lepszego z uwagi na co najmniej jedno kryterium bez pogorszenia wyniku z uwagi na pozostałe. Jeżeli dla zadania minimalizacji stosuje się  $k$  funkcji kryterialnych:

$$C(x) = (c_1(x), c_2(x), \dots, c_i(x)) \quad (2.20)$$

Jeżeli wśród wszystkich znalezionych rozwiązań możliwe jest wskazanie pojedynczego rozwiązania, które jest nie gorsze od pozostałych rozwiązań dla wszystkich funkcji kryterialnych, wtedy takie rozwiązanie jest rozwiązaniem zdominowanym, a zależność ta jest matematycznie opisywana w następujący sposób:

$$c_i(x) \geq c_i(y) \quad (2.21)$$

dla  $i = 1, \dots, k$ .

Problem wielokryterialny można sprowadzić do problemu jednokryterialnego np. poprzez zastosowanie metody ważonych kryteriów. W takim przypadku, kryterium zastępcze jest definiowane jako suma wszystkich funkcji celu, gdzie dodatkowo każdej funkcji celu przypisywana jest waga określająca jej istotność w całym równaniu:

$$C_x = \sum_{i=1}^k w_i c_i(x) \quad (2.22)$$

gdzie  $k$  określa liczbę funkcji celu,  $x$  to wektor rozwiązań a  $w$  stanowią wagi, których suma jest równa 1:

$$\mathbf{w} \in [0, 1] \sum_{i=1}^k w_i = 1 \quad (2.23)$$

W wyniku takiego przekształcenia uzyskuje się pojedynczą funkcję celu  $C(x)$ , którą optymalizuje się przy użyciu metod typowych dla zadania jednokryterialnego. Graficznie rozwiązanie można przedstawić jako punkt przecięcia obszaru rozwiązań dopuszczalnych z hiperprostą, zależną od wartości ważności kryteriów  $w_i$ . Problematyczny w tej metodzie jest wybór wartości wag kryteriów, co w oczywisty sposób może prowadzić do różnych rozwiązań. Zaletą tej metody jest prosta implementacja.

Zadanie optymalizacji jest powszechnie stosowane w zagadnieniach diagnostyki procesów przemysłowych. Autor niniejszej rozprawy stosował w swoich badaniach to podejście w celu znalezienia optymalnych wartości parametrów klasyfikatorów procesu wnioskowania (Kalisch *i in.*, 2020).

## 2.5. Podsumowanie

Diagnostyka procesów przemysłowych, jako jedna z gałęzi diagnostyki technicznej, zajmuje się detekcją i rozpoznawaniem zmian stanu procesów przemysłowych. Niniejszy rozdział zawiera przegląd podstawowych pojęć i koncepcji związanych w diagnostyką procesów, a istotnych z punktu widzenia badań realizowanych w rozprawie. W rozdziale



przedstawiono pojęcia podstawowe związane z tematem pracy takie jak obiekt diagnozowania, sygnał, cecha sygnału, sygnał diagnostyczny i fuzja danych. Ponieważ zagadnieniu kontekstu poświęcono cały kolejny rozdział pracy, w tym rozdziale pojęcie to zostało jedynie zainicjowanie. W rozdziale dokonano krótkiej charakterystyki dwóch podstawowych koncepcji diagnostyki procesów przemysłowych, a mianowicie diagnostyki wspartej modelami oraz diagnostyki symptomowej. Opisano najpopularniejsze klasyfikatory stosowane w zagadnieniach diagnostyki, między innymi klasyfikator k-najbliższych sąsiadów, naiwny klasyfikator bayesowski, drzewo decyzyjne oraz klasyfikatory wykorzystujące sieci neuronowe. Przedstawiono również sposoby estymacji błędu klasyfikacji oraz miary opisujące skuteczność klasyfikacji. W ostatnim punkcie rozdziału omówiono pojęcie optymalizacji zaczynając od genezy tego zagadnienia, a kończąc na algorytmach genetycznych, które są elementem metody opisywanej w dalszej części pracy.



## Rozdział 3

# Kontekst

W rozdziale opisano jeden z najistotniejszych elementów pracy, czyli kontekst. Na początku skupiono się na genezie i definicji pojęcia kontekstu oraz jego historii. Zaznaczono również wybrane zastosowania kontekstu także w obszarach związanych z diagnostyką czy uczeniem maszynowym. Drugą część rozdziału poświęcono w całości zastosowaniu kontekstu w zagadnieniach związanych z diagnostyką procesów oraz omówiono wybrane podejścia wykorzystujące kontekst w procesie budowania schematów wnioskowania używanych np. w celu detekcji uszkodzeń.

### 3.1. Definicja kontekstu

Zgodnie z definicją w słowniku języka polskiego PWN, pojęcie kontekst przede wszystkim związane jest z naukami lingwistycznymi ponieważ spośród czterech dostępnych definicji, trzy z nich wiążą się z dostarczeniem odbiorcy dodatkowych informacji pozwalających na precyzyjne zrozumienie treści w postaci tekstu, utworów literackich, dzieł naukowych, itp (Drabik *i in.*, 2021). Takie rozumienie kontekstu wydaje się też być naturalną interpretacją tego pojęcia, ponieważ w życiu codziennym kontekst stanowi bardzo ważny element poprawnej komunikacji pomiędzy ludźmi. Brak lub błędny kontekst może w znaczący sposób wpłynąć na niepoprawny odbiór prezentowanych treści i prowadzić do błędnych konkluzji (Adair *i in.*, 2015). Przeprowadzone badania pokazują (Afzaal *i in.*, 2020), że jednoznaczna i uniwersalna definicja kontekstu jest trudna do określenia. Jednakże inna definicja pojęcia kontekstu zawarta w słowniku języka polskiego PWN (Drabik *i in.*, 2021) brzmi „zespół czynników współistniejących, powiązanych z czymś”, co daje znacznie większą przestrzeń do interpretacji i pozwala na przypisanie kontekstu również w innych obszarach naukowych takich jak psychologia, filozofia jak i również w naukach technicznych związanych np. z zagadnieniami sztucznej inteligencji (McCarthy, 1996; McCarthy i Buvac, 1998; Akman i Surav, 1996; Brézillon, 1999). Johna McCarthy w bardzo dużym stopniu przyczynił się do popularyzacji podejścia kontekstowego w zagadnieniach sztucznej inteligencji, co zostało docenione przez inne zespoły badawcze, które dostrzegły korzyści z takiego podejścia i zaczęły wprowadzać kontekst do swoich badań. W sztucznej inteligencji można wyszczególnić szereg definicji kontekstu dotyczących zagadnień

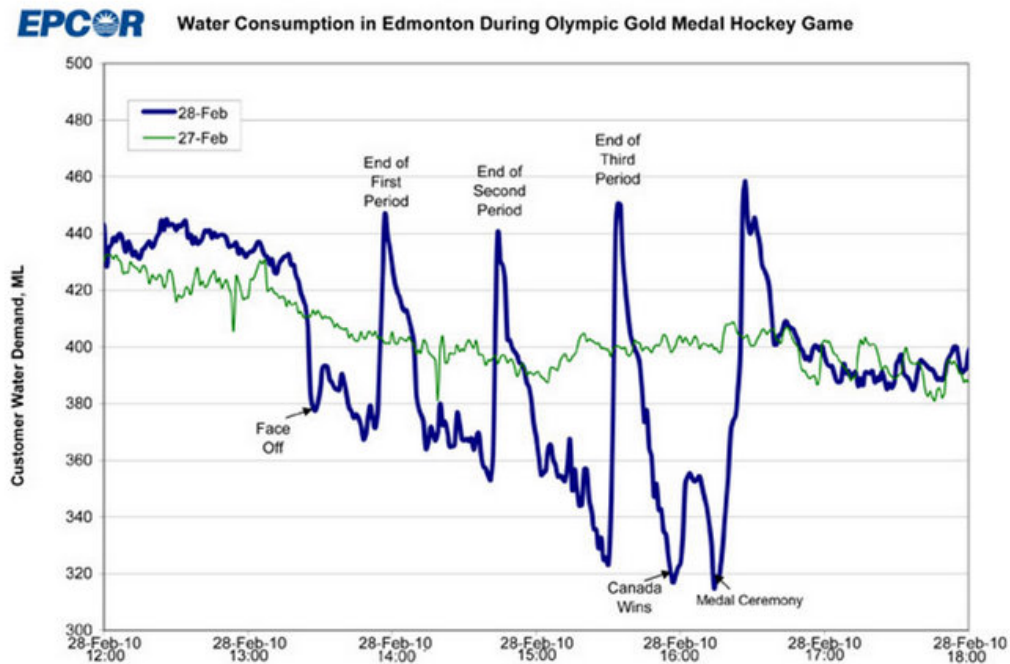
drążenia danych (Singh *i in.*, 2003), uczenia maszynowego (Nascimento *i in.*, 2018), w szczególności analiza dokumentów tekstowych (Takeuchi *i in.*, 2008), strukturyzacji reguł w systemach wspomagania decyzji, sposobu prezentowania informacji użytkownikowi, czy w celu grupowania wiedzy i sposobu użycia jej w procesie wnioskowania (Timofiejczuk, 2011).

Rozszerzająca się świadomość istnienia kontekstu również w naukach technicznych pozwoliła na implementację kontekstu w kolejnych systemach i aplikacjach tworząc systemy kontekstowe (*ang. context-aware systems*) (Weiser, 1991; Vahdat-Nejad *i in.*, 2016; Alegre *i in.*, 2016; Baldauf *i in.*, 2007). Korzyścią płynącą z podejścia kontekstowego w systemach, jest możliwość dostosowania jego działania na podstawie informacji dodatkowych (np. z otoczenia), a nie będących wynikiem poleceń płynących od użytkownika. Przykładem takiego podejścia mogą być aplikacje (np. mobilne), które automatycznie dostosowują wygląd swojego interfejsu graficznego bazując na szeregu informacji płynących z licznych sensorów. Serwisy internetowe, które podpowiadają treści zgodne z wcześniej oglądanymi przez użytkownika materiałami. Rozwiązania należące do grupy systemów określanych jako inteligentny dom (*ang. Smart Home*) potrafiące reagować na położenie i czynności wykonywane przez użytkowników.

Inną grupą systemów informatycznych, w których od samego początku rozważano uwzględnienie kontekstu są systemy doradcze znane również jako systemy ekspertowe (Moczulski, 2005; Cholewa i Pedrycz, 1987). Zadaniem takiego systemu jest wspomaganie człowieka w podejmowaniu decyzji w zakresie działań technicznych.

Ciekawym przykładem występowania kontekstu w praktyce można zaobserwować na wykresie 3.1 zaprezentowanym przez przedsiębiorstwo wodociągowe w Edmonton w Kanadzie.

Na wykresie pokazano porównanie poziomu zużycia wody w Edmonton między 27 (linia zielona), a 28 lutego (linia niebieska) w godzinach od 12:00 do 18:00. Można zauważyć wyraźną zmianę pomiędzy tymi wykresami, które wskazują na istotną różnicę w sposobie gospodarowania wodą przez mieszkańców. Powodem widocznych anomalii był rozgrywany 28 lutego mecz o złoty medal w hokeju pomiędzy Kanadą i Stanami Zjednoczonymi. Mecz cieszył się wśród kanadyjczyków ogromnym zainteresowaniem, co można zaobserwować na przedstawionym wykresie. W trakcie trwania gry zużycie wody wyraźnie spadało aż do chwili zakończenia jednej z tercji meczu i rozpoczęcia przerwy. W trakcie przerwy zużycie wody gwałtownie wzrastało aby ponownie wyraźnie spaść w momencie rozpoczęcia kolejnej tercji. Zależność ta trwa aż do zakończenia ceremonii rozdania medali, po której ponownie zużycie wody wzrastało. Przykład ten wyraźnie pokazuje jak duży wpływ na funkcjonowanie monitorowanego obiektu (w tym przypadku sieci wodociągowej) mogą mieć niektóre wydarzenia. Zakładając, że sieć wodociągowa jest monitorowana przez system, którego celem jest detekcja anomalii, które mogą świadczyć o wystąpieniu wycieków lub innych problemów, takie gwałtowne zmiany w zużyciu wody mogą zostać zinterpretowane przez system doradczy jako awarie. Generowanie dużej liczby fałszywych alarmów może negatywnie wpłynąć na funkcjonalność systemu poprzez częste wprowadzanie w błąd operatora. W opisanej wyżej sytuacji, takie wy-



Rys. 3.1: Wykres zużycia wody w Edmonton w trakcie finałowego meczu hokeja (Kiernan, n.d.)

darzenie może być traktowane jako kontekst działania sieci wodociągowej. Dodanie do systemu monitorującego informacji kontekstowej w postaci szczególnych wydarzeń występujących w przyszłości, może pomóc w opracowaniu skuteczniejszego systemu, który będzie przygotowany na nietypowe wartości zmiennych procesowych.

Jednym z obszarów, dla których tworzy się systemy doradcze jest diagnostyka techniczna, której elementem jest diagnostyka procesów. Wśród wielu publikacji dotyczących systemów doradczych można znaleźć nawiązania do kontekstu (Korbicz *i in.*, 2004). Jednym z często występujących typów systemów doradczych są regułowe systemy ekspertowe (Niederliński, 2000), w których wiedza diagnostyczna służąca do utworzenia systemu doradczego jest zapisywana w postaci tzw. reguł prostych, które zapisuje się w postaci:

$$\text{jeżeli } A \text{ to } B \quad (3.1)$$

gdzie  $A$  jest warunkiem, a  $B$  konkluzją. Na tej podstawie buduje się proste zdania logiczne i reguły złożone połączone funktorami **i**, **lub**. Reguły takie posiadają kompletną, często bardzo rozbudowaną część warunkową, charakteryzującą się dużą liczbą warunków cząstkowych. Reguły złożone są stosunkowo proste w interpretacji jednak są zasadniczo trudne w formułowaniu, weryfikacji i uzupełnianiu. Dodatkową wadą jest to, że taki zbiór reguł charakteryzuje się dużą nadmiarowością informacji, ponieważ identyczne warunki cząstkowe są powtarzane w dużej liczbie reguł. W celu zniwelowania istotności tych wad stosuje się stwierdzenia pomocnicze mające postać wniosków pośrednich. Reguły umożliwiające wyznaczenie takich wniosków nie posiadają nadmiernie rozbudowanej części

warunkowej. Zaletą takiego postępowania jest możliwość prostej weryfikacji zbioru reguł i ograniczenie ich redundancji. Przykładem prostego zastosowania warunków pośrednich reprezentowanych przez kontekst w systemach doradczych jest uporządkowanie kolejności wykonywanych reguł.

W przypadku systemu doradczego, którego zadaniem jest również interakcja z użytkownikiem i zadawanie pytań, zmiana kolejności warunków może mieć wpływ na kolejność zadawanych pytań i ich zakres. Zastosowanie w tym przypadku kontekstu  $C$  umożliwia wydzielenie części wiedzy zapisanej w formie reguł. Zastosowanie kontekstu w regule złożonej można przedstawić w następującej formie:

$$\text{jeżeli } C_1 \text{ i } A_3 \text{ to } B \quad (3.2)$$

gdzie:

$$C_1 = A_1 \text{ i } A_2 \quad (3.3)$$

Wprowadzenie pojęcia kontekstu pozwala na wyraźną strukturyzację i uporządkowanie zbioru reguł.

Na potrzeby niniejszej pracy kontekst rozumiany jest jako część wiedzy, która nie jest używana wprost do znalezienia rozwiązania, ale wpływa na wybór wiedzy bezpośrednio związanej z rozpatrywanym problemem (kontekst sproceduralizowany). Kontekst używany jest do reprezentacji procesu wnioskowania w przestrzeni o ograniczonej liczbie stanów, w których problem lub obiekt może się znaleźć. Kontekst może być związany np. z otoczeniem lub środowiskiem, w którym działa obiekt lub z zależnościami pomiędzy własnościami obiektu.

Kontekst jest częścią wiedzy, która pozwala na wybranie pewnego fragmentu wiedzy połączonej z danym kontekstem. Kontekst stanowi pewnego rodzaju filtr, który pozwala na ograniczenie wiedzy potrzebnej do realizacji zadania (Timofiejczuk, 2011). W pracy przyjęto, że podstawą reprezentacji kontekstu jest cecha kontekstowa (Turney, 2002a). Biorąc to pod uwagę, opis kontekstu może zostać przedstawiony za pomocą zbioru danych  $\mathbf{E} = [cl_1, cl_2, \dots, cl_N, kd]$ , który składa się z zbioru cech liczbowych  $\mathbf{CL}$  oraz klasy określającej docelową końcową decyzję klasyfikatora  $\mathbf{KD}$ . Parametry  $x$  i  $y$  reprezentują w dalszej części tego opisu wartości stałe przypisane do cech liczbowych  $\mathbf{CL}$  w zbiorze danych  $\mathbf{E}$ , a parametr  $z$  zawiera wartości dla zbioru  $\mathbf{KD}$ . Spośród wszystkich dostępnych cech, na podstawie których realizowane jest wnioskowanie, wyodrębnić można cztery podstawowe grupy rozpatrywane i stosowane w pracy (Turney, 2002c).

- Cecha podstawowa ( $cp_{i,k}$ ) stanowi cechę, która niesie ze sobą informację na temat przynależności określonej próbki danych do jednej z klas ze zbioru  $F$ , który definiuje zbiór wszystkich możliwych klas (dla problemu klasyfikacji) lub znanych stanów obiektu diagnozowania, np. F0 – stan pełnej zdatności, FX – stan z uszkodzeniem. Można ją wyrazić za pomocą prawdopodobieństwa warunkowego w postaci wzoru:

$$p(kd_k = z_k \mid cp_{i,k} = x_{i,k}) \neq p(kd_k = z_k) \quad (3.4)$$

gdzie  $i$  jest indeksem cechy podstawowej z przedziału  $1 \leq i \leq N$  a  $k$  jest indeksem wiersza danych z przedziału  $1 \leq k \leq K$ .

- Cecha kontekstowa ( $ck_{j,k}$ ) nie jest cechą podstawową, dlatego nie niesie ze sobą bezpośrednio informacji o przynależności próbek do klas, ale może zostać wykorzystana w celu poprawy ogólnej sprawności klasyfikacji. Może zostać opisana za pomocą wzoru:

$$p(kd_k = z_k \mid ck_{j,k} = y_{j,k}) = p(kd_k = z_k) \quad (3.5)$$

gdzie  $j$  jest indeksem cechy kontekstowej z przedziału  $1 \leq j \leq M$  a  $k$  jest indeksem wiersza danych z przedziału  $1 \leq k \leq K$ .

- Cecha kontekstowo zależna jest cechą podstawową  $cp_i$ , która jest zależna od cechy kontekstowej  $ck_j$  oraz spełnia nierówność:

$$p(kd_k = z_k \mid cp_{i,k} = x_{i,k}, ck_{j,k} = y_{j,k}) \neq p(kd_k = z_k \mid cp_{i,k} = x_{i,k}) \quad (3.6)$$

gdzie  $i$  jest indeksem cechy podstawowej z przedziału  $1 \leq i \leq N$ ,  $j$  jest indeksem cechy kontekstowej z przedziału  $1 \leq j \leq M$  a  $k$  jest indeksem wiersza danych z przedziału  $1 \leq k \leq K$ .

- Cecha nieistotna, która nie należy do żadnej z wyżej wymienionych grup i nie jest użyteczna z punktu widzenia zadania klasyfikacji.

Kontekst może być znany lub ukryty, w przypadku tego drugiego istnieją metody pozwalające na odkrycie kontekstu (Timofiejczuk, 2011). Kontekst w takim przypadku jest identyfikowany na podstawie zbioru przypadków, będących wynikiem grupowania. W podejściu bazującym na identyfikacji ukrytego kontekstu należy również wziąć pod uwagę sytuację, gdzie w trakcie wnioskowania w oparciu o dane bieżące, zostanie odkryty kontekst, który nie występuje w bazie zidentyfikowanych kontekstów. W takim przypadku należy uzupełnić bazę kontekstów, co jest realizowane podczas procesu wnioskowania. Celem zastosowania kontekstu w systemie diagnostycznym jest np. poprawienie wskaźników działania systemu poprzez zmniejszenie liczby fałszywych alarmów lub ograniczenie złożoności obliczeniowej algorytmów. Ponieważ w systemach diagnostycznych powszechnie stosuje się metody uczenia maszynowego (klasyfikatory), jedną z metod zastosowania jest uwzględnienie kontekstu w procesie trenowania klasyfikatorów.

## 3.2. Kontekst w zadaniach klasyfikacji

Powszechnie znanym problemem związanym z uczeniem maszynowym jest dryf konceptu (*ang. Concept Drift*). Problem jest szczególnie widoczny w przypadku realizacji zadania klasyfikacji z wykorzystaniem danych strumieniowych (Kuncheva, 2008). Podstawowym etapem budowania klasyfikatora jest jego wytrenowanie, w tym celu konieczne jest zebranie odpowiednich danych trenujących, które pozwolą na utworzenie klasyfikatora przygotowanego do rozpoznawania określonych stanów. Dalsze wykorzystanie klasyfikatora

w celu realizacji zadania klasyfikacji dla nowych danych zakłada, że nowe dane nie będą statystycznie znacząco odbiegać od danych trenujących, co może być reprezentowane np. za pomocą zbliżonych do siebie rozkładów normalnych wyznaczonych na bazie danych trenujących i danych docelowych. Terminem *Concept Drift* określa się sytuację, w której charakterystyka nowych danych podawanych na wejście klasyfikatora zaczyna odbiegać od danych, na których klasyfikator został wytrenowany. Proces ten charakteryzuje się dosyć niewielką dynamiką i nie jest łatwo obserwowalny, jednak w długofalowym procesie ma bardzo istotny wpływ na skuteczność działania klasyfikatora pogarszając jego sprawność, czego rezultatem jest np. zwiększenie liczby fałszywych alarmów i/lub zmniejszenie liczby poprawnych alarmów. Proces ten (*ang. Concept Drift*), z czasem w literaturze zaczął być utożsamiany z określeniem kontekstu, ze względu na możliwe przyczyny występowania takiego dryfu, które mogą wynikać np. z oddziaływania konkretnych obszarów otoczenia, w którym znajduje się obiekt, z którego pobierane są dane wejściowe do klasyfikatora (Widmer i Kubat, 1994; Widmer, 1997).

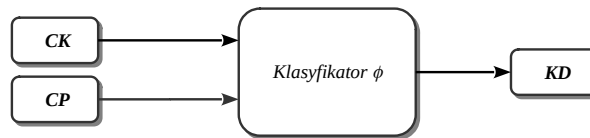
Innym zagadnieniem łączącym w sobie uczenie maszynowe oraz kontekst, jest rozpoznanie kontekstu w danych np. za pomocą klasyfikatora. W podejściu tym, zakłada się, że kontekst w którym realizowany jest proces jest znany, jednak informacje na jego temat nie są bezpośrednio dostępne. Aby uzyskać dostęp do kontekstu, możliwe jest zastosowanie metod uczenia maszynowego, które pozwolą na zidentyfikowanie kontekstu, w którym aktualnie znajduje się proces. Zadanie zebrania odpowiedniej ilości danych wymaganych do wytrenowania klasyfikatora nie powinno stanowić problemu, ponieważ zakłada się, że proces zawsze działa w jednym ze znanych kontekstów. W pracy (Kalisch, 2015) autor przedstawił przykład detekcji kontekstu pracy kombajnu górniczego, gdzie rozpatrywano sześć charakterystycznych stanów pracy kombajnu, które w wyniku analizy zebranych danych oraz logicznej identyfikacji możliwych kombinacji, zostały sprowadzone do sześciu możliwych wariantów kontekstu, które były ekstrahowane ze zmiennych procesowych za pomocą klasyfikatora.

Jednym z autorów skupiających się w swojej pracy na kontekście oraz uczeniu maszynowym z wykorzystaniem komitetów klasyfikatorów jest Peter Turney (Turney, 2002a; Turney, 2002b; Turney, 2002c). W swoich pracach przedstawia koncepcje realizacji klasyfikacji, również z uwzględnieniem kontekstu opierając się również na przykładach związanych z diagnostyką techniczną.

Jedną z podstawowych operacji, które dotyczą uwzględnienia kontekstu, jest przekształcenie danych wejściowych do klasyfikatora w taki sposób, aby dane wrażliwe na kontekst (cechy kontekstowo zależne) były rozpatrywane przez klasyfikator jako cechy podstawowe. Podejście to, pozwala na uwzględnienie kontekstu na bardzo wczesnym etapie realizacji procesu klasyfikacji bez konieczności ingerowania w sam klasyfikator czy budowanie bardziej zaawansowanych struktur klasyfikatorów opisanych w dalszej części pracy. Jedną z metod, którą można zastosować w celu realizacji tego zadania jest normalizacja danych, przy czym normalizacja realizowana jest osobno dla każdego z rozpatrywanych kontekstów.

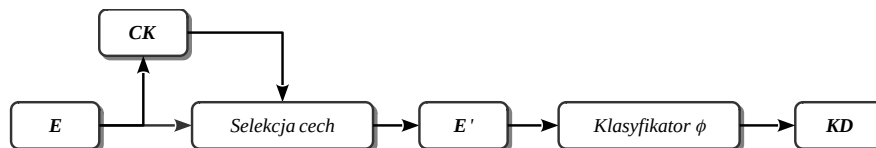
Kolejną metodą niewymagającą budowania komitetów klasyfikatorów jest traktowa-





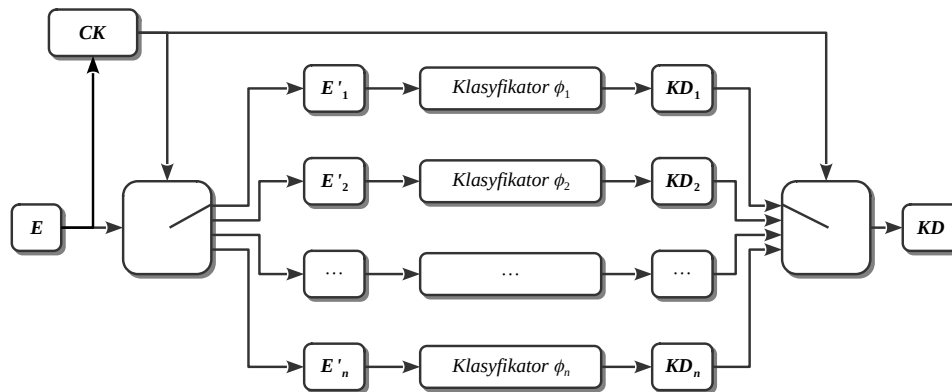
Rys. 3.2: Schemat kontekstowej ekspansji

nie cech kontekstowych **CK**, jako cechy podstawowe **CP** podawanej na wejście klasyfikatora razem z pozostałymi danymi, co pokazano na rysunku 3.2. Metoda ta, w literaturze jest określana jako kontekstowa ekspansja danych (*ang. Contextual Expansion*) (Turney, 2002a). W takim podejściu to od klasyfikatora zależy w jak dużym stopniu cecha kontekstowa będzie uwzględniona w procesie wnioskowania. Jeżeli na etapie przygotowania danych, przed podaniem ich na wejście klasyfikatora, zastosowana zostanie metoda nadzorowanej selekcji cech, ze zbioru danych mogą zostać usunięte atrybuty zawierające cechy kontekstowe. Wynika to z faktu, że cechy kontekstowe z założenia nie zawierają bezpośrednio informacji o stanie technicznym urządzenia lub procesu. Również zastosowanie niektórych algorytmów uczenia klasyfikatorów bazujących na regułach (jak np. drzewo decyzyjne) może przynieść podobny rezultat jak w przypadku użycia metody nadzorowanej selekcji cech.



Rys. 3.3: Schemat metody kontekstowo ważonych klasyfikatorów

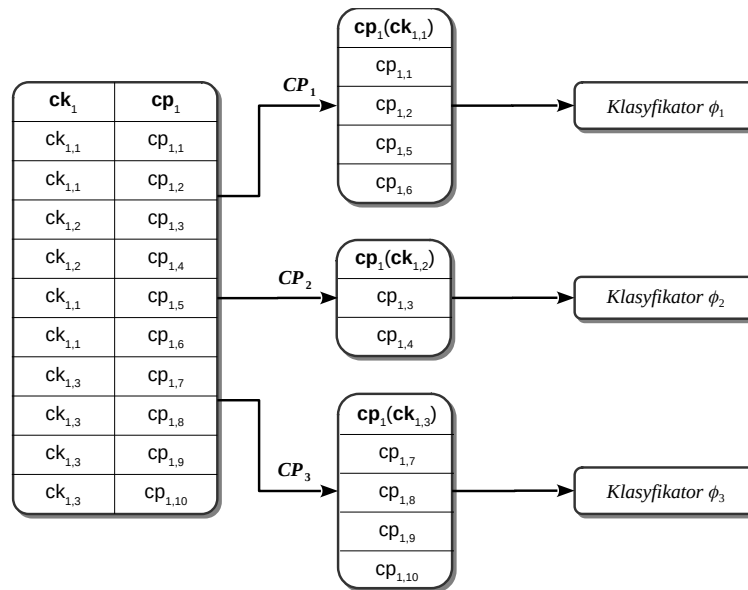
Innym podejściem polegającym na uwzględnieniu kontekstu w procesie przygotowywania danych jest selekcja najistotniejszych atrybutów z punktu widzenia kontekstu (3.3). W tym przypadku stosuje się dodatkowy klasyfikator (np. drzewo decyzyjne), który w procesie trenowania klasyfikatora końcowego służy do określenia, które cechy podstawowe z całego zbioru **E** niosą ze sobą najistotniejsze informacje w powiązaniu z cechami kontekstowymi **CK**. Informacja ta służy w dalszej części procesu do określenia wag dla każdego z sygnałów w postaci zmodyfikowanego zbioru **E'**. Liczba wag dla każdego z sygnałów odpowiada liczbie kontekstów. Wagi są uwzględniane w końcowym procesie klasyfikacji zarówno w trakcie trenowania, testowania jak i właściwego działania klasyfikatora lub klasyfikatorów końcowych.



Rys. 3.4: Schemat metody kontekstowego wyboru klasyfikatorów

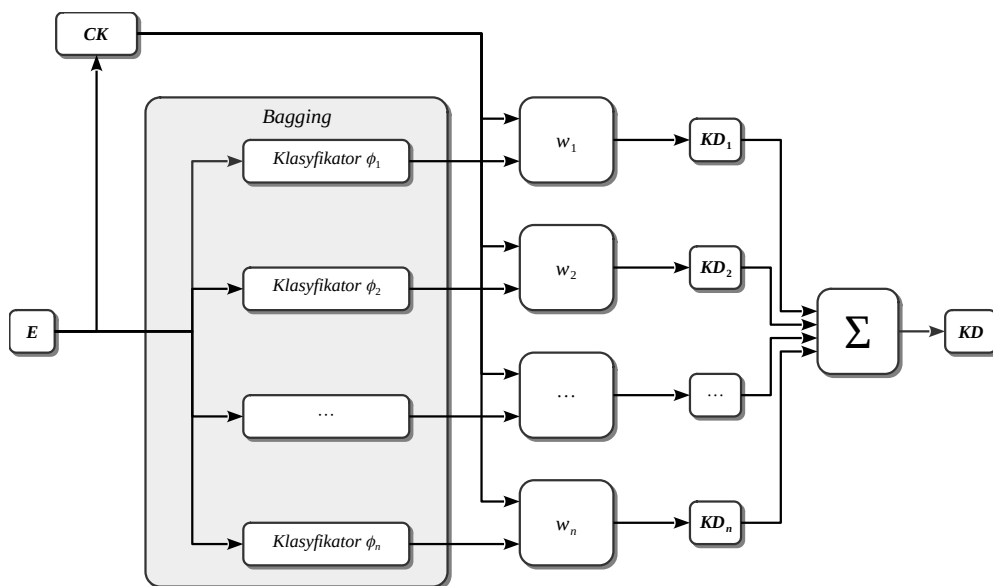
Jednym z najczęstszych sposobów uwzględnienia kontekstu w metodach uczenia maszynowego, jest budowanie struktur wieloklasyfikatorowych, w których kontekst ma wpływ na sposób realizacji procesu wnioskowania w komitecie. Przykładem takiego podejścia jest metoda określana jako kontekstowy wybór klasyfikatora (*ang. Contextual classifier selection*, 3.4) (Kalisch *i in.*, 2020; Kalisch, 2018). W podejściu uwzględniającym klasyfikator, opisanym w niniejszym punkcie rozprawy, każdy z klasyfikatorów w komitecie jest związany z innym wariantem lub wariantami cech kontekstowych  $CK$ . Liczba klasyfikatorów nie może być większa niż liczba wariantów kontekstów. Minimalna liczba klasyfikatorów w komitecie nie jest określona, jednak zastosowanie tylko jednego klasyfikatora spowoduje, że cecha kontekstowa nie będzie mieć żadnego wpływu na jego działanie. Taki klasyfikator zawsze będzie trenowany na pełnym zbiorze danych i będzie również realizował proces klasyfikacji dla wszystkich danych wejściowych na etapie stosowania modelu w diagnostyce procesu. Głównym elementem metody jest podział danych (zarówno trenujących, testowych, weryfikacyjnych i wynikających z działania procesu) ze względu na przynależność do wariantu kontekstu. Takie podejście pozwala na przygotowanie odrębnych grup danych trenujących dla każdego z klasyfikatorów zastosowanych w komitecie. Dzięki temu możliwe jest zastosowanie zbioru klasyfikatorów tego samego typu (np. drzewa decyzyjne) lub zupełnie różnych klasyfikatorów dla poszczególnych wariantów kontekstu. Tak jak to pokazano na schemacie 3.4, to wartość cechy kontekstowej decyduje o tym, do którego klasyfikatora zostanie wysłany przypisany do niej wektor. Wynikiem działania każdego klasyfikatora bazowego jest końcowa decyzja klasyfikatora  $KD$ , jednak decyzja jest wynikiem działania tylko jednego klasyfikatora  $\phi_i$ , związanego z daną wartością cechy kontekstowej.

Na rysunku 3.5 przedstawiono sposób działania algorytmu podziału danych trenujących ze względu na wartość cechy kontekstowej  $ck_1$ . W prezentowanym przykładzie początkowy zbiór danych składa się z pojedynczej cechy kontekstowej  $ck_1$  oraz pojedynczej cechy podstawowej  $cp_1$ . Jak można zaobserwować na schemacie, mechanizm ten tworzy odseparowane względem siebie podzbiory danych  $CP_i$ , które w kolejnym kroku służą do wytrenowania klasyfikatorów  $\phi$ . Podzbiory danych utworzone dla określonych



Rys. 3.5: Schemat podziału danych względem cechy kontekstowej

wartości cechy kontekstowej zostały na rysunku oznaczone etykietą  $CP_i$ , które zawierają pojedynczą cechę podstawową  $cp_1(ck_i)$ , co oznacza podzbiór zawierający wartości cechy podstawowej  $cp_1$  wyselekcjonowane tylko dla wartości cechy kontekstowej  $ck_i$ .

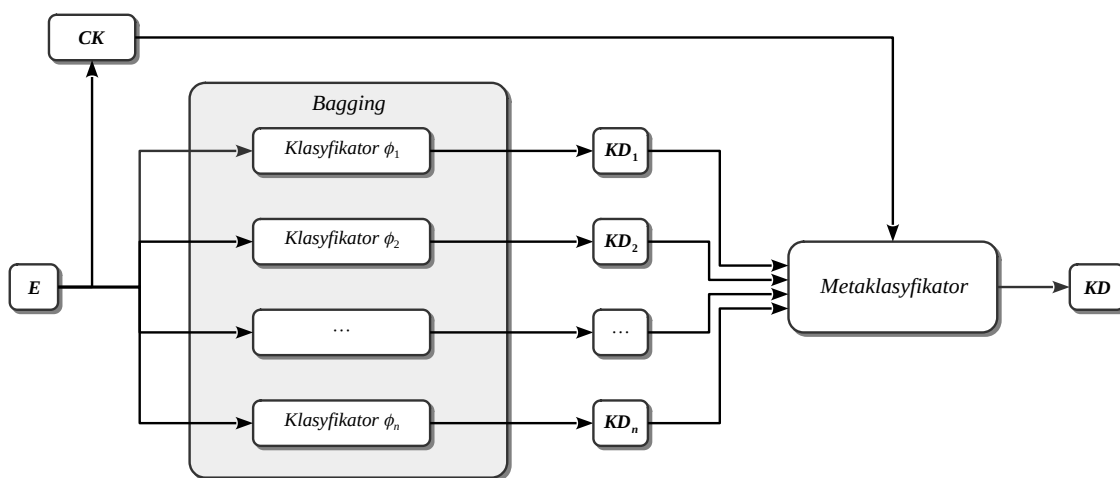


Rys. 3.6: Schemat metody śpiących ekspertów

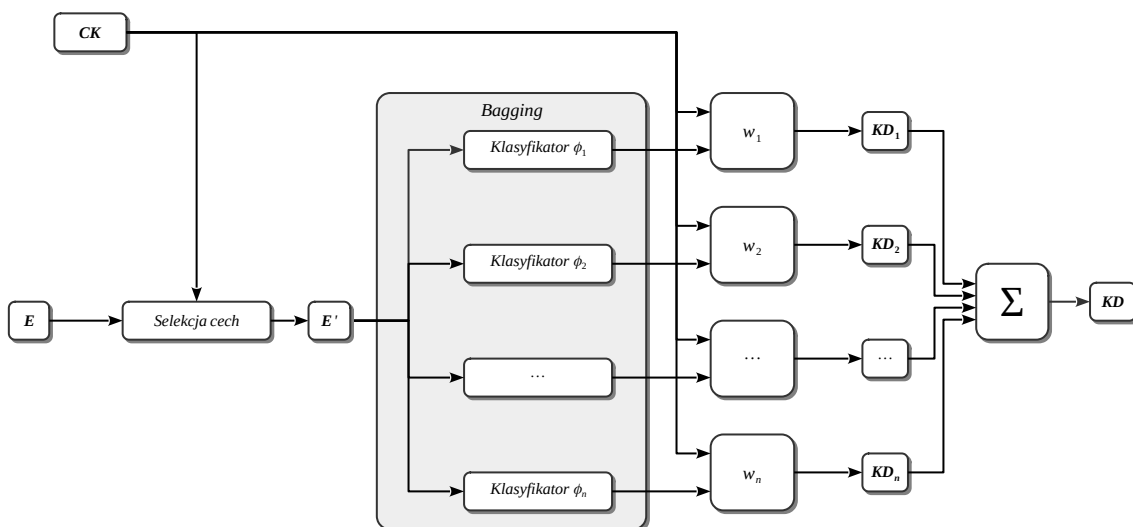
Kolejną metodą bazującą na komitecie klasyfikatorów jest podejście określane w literaturze jak *Śpiący eksperci* (ang. *Sleeping experts*). W przypadku tej metody końcowy wynik klasyfikacji opiera się na metodzie ważonego głosowania większościowego, gdzie wagi  $w_i$  zmieniają się w zależności od wartości cech kontekstowych  $CK$ , co pokazano na rysunku 3.6. W metodzie tej można zastosować komitet dowolnych klasyfikatorów bazujących na różnych metodach klasyfikacji lub zastosować komitet klasyfikatorów, ba-

zujących na tej samej metodzie klasyfikacji natomiast zróżnicowanie wprowadzić poprzez zastosowanie jednej z metod dywersyfikujących zbiory danych trenujących jak np. Bagging, co zostało opisane w rozdziale 2.3.2.

Jedną z metod fuzji klasyfikatorów jest zastosowanie metaklasifikatora, który na podstawie wyników działania klasyfikatorów bazowych dokonuje wygenerowania końcowej decyzji. Uwzględnienie kontekstu w tej metodzie jest realizowane za pomocą metaklasifikatora, w którym cecha kontekstowa jest traktowana jako dodatkowa cecha podstawowa (3.7). Od klasyfikatora zależy w jak dużym stopniu będzie ona uwzględniona w procesie wnioskowania. Istnieje możliwość zastosowania cechy kontekstowej w procesie losowania wejściowych zmiennych procesowych klasyfikatorów bazowych, w celu ukierunkowania ich działania na określony kontekst.



Rys. 3.7: Kontekstowy wybór klasyfikatora



Rys. 3.8: Schemat metody łączącej kontekstowo wagi klasyfikatory i śpiących ekspertów

Realizując proces wnioskowania z uwzględnieniem kontekstu istnieje możliwość wykorzystanie cechy kontekstowej na wielu etapach tego procesu. Poprzez połączenie dwóch opisanych wcześniej metod, można utworzyć nowe podejście, które w pewnych zastosowaniach może pozwolić uzyskać bardziej zadowalające rezultaty. Na rysunku 3.8 zaprezentowano schemat, który w pierwszej kolejności bazuje na metodzie przypisującej do atrybutów kontekstowo zależne wagi, które w dalszym etapie pozwalają na ograniczenie liczby cech na wejściu klasyfikatorów poprzez selekcję zależną od tych wag. W drugim kroku wyniki działania klasyfikatorów należących do komitetów poddawane są głosowaniu z uwzględnieniem wag, które również zostały określone na podstawie cechy kontekstowej.

### 3.3. Podsumowanie

Kontekst jest szerokim pojęciem, które jest powszechnie używany razem z zagadnieniami należącymi do różnych obszarów nauki i życia codziennego. Kontekst jest naturalnym elementem lingwistyki wykorzystywanym przez ludzi, znacząco wpływającym na jakość komunikacji. W niniejszym rozdziale opisano podstawowe zagadnienia związane z kontekstem oraz omówiono historię rozwoju definicji i interpretacji kontekstu w obszarach związanych z diagnostyką techniczną i systemami doradczymi, a przede wszystkim dotyczącym uczenia maszynowego. Opisano wybrane metody zastosowania kontekstu do przygotowania danych wykorzystywanych w dalszym etapie do trenowania, testowania, weryfikacji pojedynczego klasyfikatora. Następnie omówiono wybrane podejścia bazujące na komitatach klasyfikatorów, których integralnym elementem jest cecha kontekstowa mająca wpływ na działanie klasyfikatora. W dalszej części pracy przedstawiono kompletne podejście do detekcji uszkodzeń z zastosowaniem kontekstu, w której wykorzystano jedną z metod opisanych w rozdziale 3.2 wraz z elementami opisanymi w rozdziale 2.



## Rozdział 4

# Metoda detekcji uszkodzeń z zastosowaniem kontekstu

W niniejszym rozdziale opisano poszczególne elementy metody detekcji uszkodzeń z zastosowaniem kontekstu. Metodę opisano od etapu akwizycji i przygotowania danych, poprzez wszystkie etapy przygotowania modeli klasyfikatorów z zastosowaniem algorytmów optymalizacji i z uwzględnieniem kontekstu, aż do ich zastosowania jako całościowe rozwiązanie.

### 4.1. Akwizycja danych i ekstrakcja cech

W celu przygotowania modeli diagnostycznych oraz ich dalszego zastosowania, konieczna jest akwizycja danych rejestrowanych podczas działania obiektu. Dane te mają postać zmiennych procesowych. Ich wartości mogą być rejestrowane podczas działania obiektów bezpośrednio poprzez zastosowanie sensorów monitorujących poszczególne elementy diagnozowanego procesu. W ten sposób do systemu diagnostycznego są wysłane dane opisujące np. określone wartości fizyczne, istotne z punktu widzenia poprawnie działającego procesu, takie jak temperatura, natężenie prądu, prędkość obrotowa, itp. Sposób akwizycji i udostępnienia danych w systemie diagnostycznym zależy od technologii (producent i typ sterownika PLC, oprogramowanie komputera, itp.) zastosowanej w danym przypadku. Jedną z możliwości jest zastosowanie sterownika PLC z zaimplementowanym serwerem OPC oraz oprogramowania pozwalającego na pobieranie danych w czasie rzeczywistym za pomocą klienta OPC. Dane te zawierają informacje o aktualnej wartości wybranych zmiennych procesowych oraz stempel czasowy wykonanego pomiaru.

Przygotowanie modeli klasyfikatorów opisanych w niniejszej rozprawie, jest realizowane automatycznie poprzez zastosowanie odpowiednich do typu klasyfikatora algorytmów trenujących. W tym celu konieczne jest wygenerowanie danych trenujących, czyli zbioru danych zawierających wartości zmiennych procesowych wraz z przypisaną do nich klasą (stanem). Przygotowanie danych trenujących wymaga zarejestrowania przebiegu zmiennych procesowych w trakcie realizacji zadań procesu przemysłowego, czyli następujących po sobie operacji. W przypadku systemu diagnostycznego bazującego na mo-

delach klasyfikatorów, które powstają w wyniku uczenia nadzorowanego, konieczne jest wygenerowanie danych trenujących związanych ze wszystkimi rozpatrywanymi stanami. Akwizycja danych będących odzwierciedleniem poprawnie działającego systemu nie stanowi problemu, ponieważ proces najefektywniej realizuje swoje zadania w stanie pełnej zdatności. Problematiczne jest uzyskanie danych odzwierciedlających stany uszkodzenia, szczególnie w przypadku obiektów technicznych, które są obiektami krytycznymi, to znaczy takimi, których awarie powodują poważne konsekwencje. Każde uszkodzenie występujące w trakcie realizacji rzeczywistego procesu wpływa negatywnie na rezultaty jego działania, co powoduje zatrzymanie takiego przebiegu. W takim przypadku najczęściej stosowanym rozwiązaniem jest wygenerowanie danych za pomocą stanowiska laboratoryjnego (Skupnik, 2009; Timofiejczuk, 2008) lub modelu matematycznego (Odgaard *i in.*, 2009; Bartyś *i in.*, 2006; Wachla, 2018; Cenacewicz i Katunin, 2016). Ograniczona liczba danych jest problemem, z którym można sobie poradzić również poprzez wybór metody trenowania i weryfikacji klasyfikatorów, co zostało opisane w dalszej części pracy.

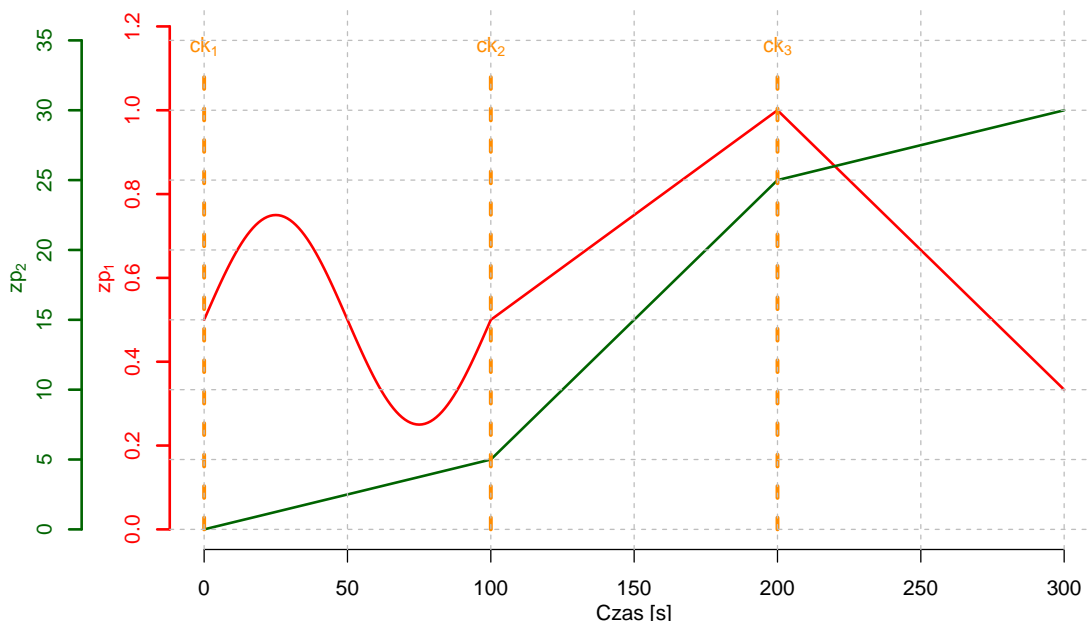
Obiekty techniczne, których dotyczy niniejsza rozprawa to instalacje, w których realizowane są procesy przemysłowe. Przerwanie przebiegu procesów lub celowe generowanie stanów odpowiadających uszkodzeniom, czy awariom, jest niemożliwe. Katedra Podstaw Konstrukcji Maszyn Politechniki Śląskiej dysponuje laboratorium wyposażonym w instalacje pozwalające na symulowanie rzeczywistego działania procesów przemysłowych, a także stanów technicznych, które odpowiadają uszkodzeniom. Dane zarejestrowane bezpośrednio podczas działania stanowiska, na którym realizowany jest proces reprezentują przebiegi czasowe wybranych zmiennych procesowych. Tego typu dane zawierają najbardziej szczegółowe informacje o poprawności przebiegu badanego procesu, jednak mogą okazać się nieodpowiednie z punktu widzenia algorytmów trenujących klasyfikatory stosowane w diagnostyce niebazującej na modelu. Z tego powodu w diagnostyce maszyn i procesów powszechnie stosuje się metody agregacji danych, które pozwalają na przekształcenie istniejących i utworzenie nowych zbiorów danych (Fidali, n.d.). W tabeli 4.1 przedstawiono listę zawierającą często stosowane cechy liczbowe obliczane w dziedzinie czasu (Fidali, n.d.; Patel *i in.*, 2013).

Na rysunku 4.1 pokazano przykładowy przebieg wartości zmiennych procesowych  $zp_1$ ,  $zp_2$  w czasie  $t$  [s]. Wykres reprezentuje dane wygenerowane za pomocą funkcji matematycznych. W wygenerowanych danych został również uwzględniony kontekst, który może być reprezentowany przez cechę lub cechy kontekstowe. Dane zostały podzielone na trzy przedziały, które są przypisane do konkretnej wartości cechy kontekstowej. Cecha kontekstowa zawierająca dodatkową informację o warunkach działania procesu odpowiada skończonej liczbie wartości, co powoduje, że jest ona rozpatrywana jako zmienna procesowa dyskretna. Na wykresie zaznaczono jedynie początek określonego przedziału czasu poprzez wyznaczenie pionowej pomarańczowej przerywanej linii. Oznaczenie to należy rozumieć w ten sposób, że wszystkie wartości zmiennych procesowych  $zp_1$  i  $zp_2$  od 0 do 100 sekundy są przypisane do wartości cechy kontekstowej równej 1 ( $ck_1$ ), między 101 a 200 sekundą do  $ck_2$ , a wartości zmiennych procesowych między 201, a 300 sekundą do  $ck_3$ . Na wykresie można też zauważyć, że każdy przedział związany jest z konkretną



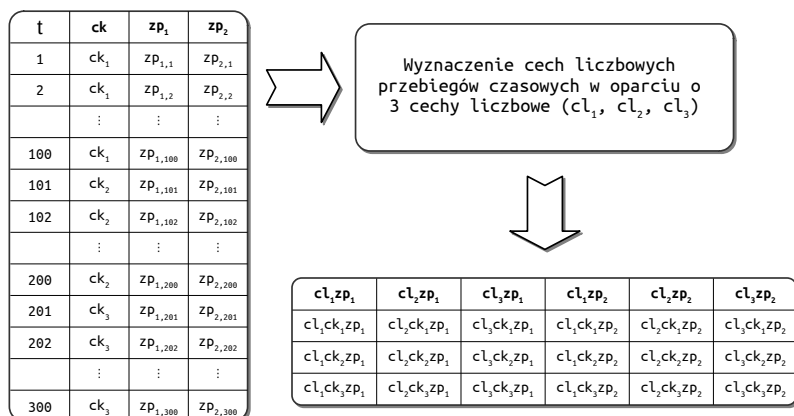
Tab. 4.1: Lista zawierająca często stosowane cechy liczbowe obliczane w dziedzinie czasu

Nazwa cechy	Definicja cechy
Wartość średnia	$x_{avg} = \frac{1}{n} \sum_{i=1}^n x_i$
Mediana	$x_{med} = x_{\frac{n}{2}}, / x_{\frac{n+1}{2}}$
Wartość minimalna	$x_{min} = \min(x)$
Wartość maksymalna	$x_{max} = \max(x)$
Odchylenie standardowe	$x_{std} = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$
Wartość skuteczna	$x_{rms} = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i)^2}$
Bezwymiarowy współczynnik kształtu	$x_{shf} = \frac{x_{rms}}{\bar{x}}$
Kurtoza	$x_{krt} = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^4$
Energia	$x_{en} = \frac{1}{n} \sum_{i=1}^n  x_i ^2$
Skośność	$x_{skw} = \frac{\sum_{i=1}^n (x_i - \bar{x})^3}{(n-1) \cdot \sigma^3}$
Entropia	$x_{ent} = - \sum_{i=1}^n p_i \log p_i$
Rozkład sygnału 1	$x_{sd1} = \frac{\sqrt{\frac{\sum_{i=1}^N (x_i)^2}{N}}}{\frac{1}{N} (\sum_{i=1}^N  x_i )}$
Rozkład sygnału 2	$x_{sd2} = \frac{\max(x_i)}{\frac{1}{N} (\sum_{i=1}^N  x_i )}$
Wartość szczytowa	$x_{peak} = \max( x )$
Bezwymiarowy współczynnik szczytu	$x_{crf} = \frac{x_{peak}}{x_{rms}}$
Bezwymiarowy współczynnik impulsowości	$x_{imp} = \frac{x_{peak}}{x_{avg}}$
Wartość międzyszczytowa	$x_{p2p} = x_{max} - x_{min}$



Rys. 4.1: Przykładowy przebieg wartości zmiennych procesowych  $zp_1$ ,  $zp_2$  w czasie  $t$  podzielone względem wartości cechy kontekstowej  $ck_1, ck_2, ck_3$

funkcją matematyczną, co symuluje różne tryby działania badanego procesu.



Rys. 4.2: Proces przekształcania przebiegów czasowych na zbiór cech liczbowych zależnych od kontekstu

Na rysunku 4.2 przedstawiono schemat przekształcania przebiegów czasowych pogrupowanych za pomocą cechy kontekstowej w zbiór cech liczbowych obliczonych w ramach cechy kontekstowej. Schemat został przygotowany w taki sposób, aby odpowiadał danym pokazanym na wykresie 4.1. Dane źródłowe zostały przedstawione na rysunku jako tabela składająca się z cechy kontekstowej ( $ck$ ) i dwóch zmiennych procesowych ( $zp_1$  i  $zp_2$ ). Można zauważyć, że każda wartość zmiennej procesowej ( $zp_{i,j}$ ) zarejestrowana w trakcie działania badanego procesu jest przypisana do wartości cechy kontekstowej ( $ck_k$ ). W wyniku tej analizy, można wyodrębnić grupy danych zawierające takie wartości zmiennych procesowych, które są przypisane do tych samych wartości cechy konteksto-

wej, co odpowiada przedziałom wyodrębnionym na wykresie 4.1. Operacja wyznaczenia cech liczbowych przebiegów czasowych zaznaczona jako osobny obiekt na schemacie przekształca wartości zmiennych procesowych w ramach tak powstałych grup do pojedynczych wartości będących obliczonymi cechami liczbowymi *SL*. Wielkość zbioru cech liczbowych zależy od liczby funkcji matematycznych zastosowanych w procesie. W tym przykładzie zastosowano trzy cechy liczbowe:

- $cl_1$  - wartość minimalna,
- $cl_2$  - wartość maksymalna,
- $cl_3$  - wartość średnia.

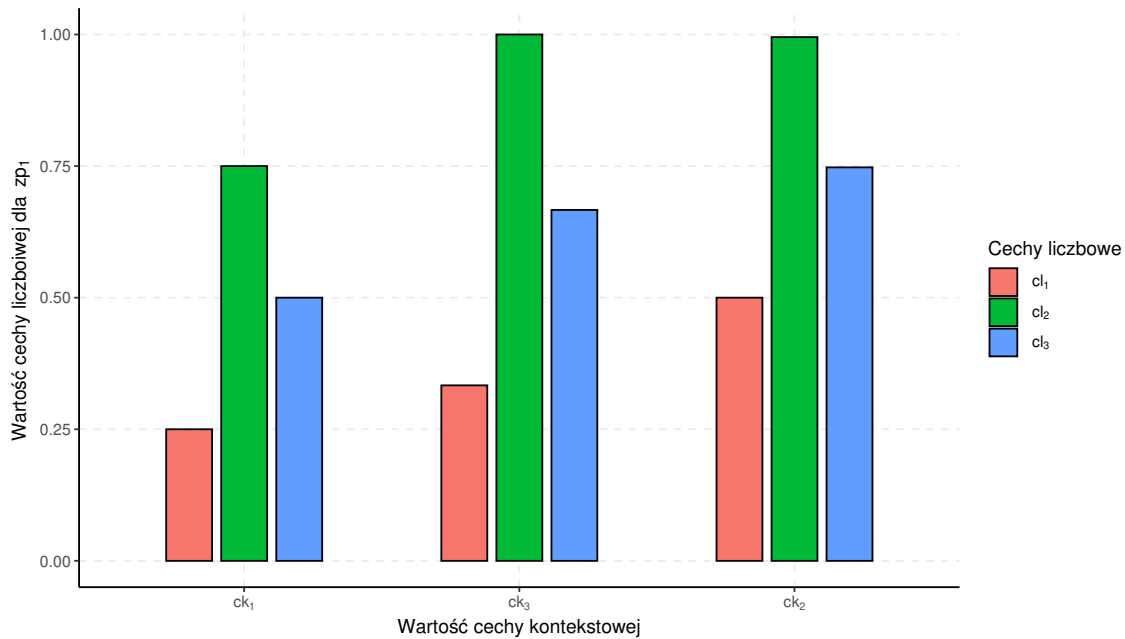
W tabeli 4.2 pokazano wartości liczbowe wyżej wymienionych cech obliczone dla każdej z dwóch zmiennych procesowych, których przebiegi zostały zaprezentowane na rysunku 4.1.

Tab. 4.2: Wartości liczbowe cech wyznaczonych dla poszczególnych zmiennych procesowych

<i>ck</i>	$cl_1zp_1$	$cl_2zp_1$	$cl_3zp_1$	$cl_1zp_2$	$cl_2zp_2$	$cl_3zp_2$
$ck_1$	0.25	0.75	0.50	0.0	5.0	2.5
$ck_2$	0.50	1.00	0.75	5.0	25.0	15.0
$ck_3$	0.33	1.0	0.66	25.0	30.0	27.5

W wyniku tego procesu redukowana jest liczba wierszy tabeli danych, jednak zwiększana jest liczba kolumn, co zostało przedstawione w tabeli 4.2. Zostały tam wymienione wszystkie wartości cech liczbowych obliczone na podstawie danych źródłowych. Można zaobserwować, że 300 wierszy danych znajdujących się w zbiorze źródłowym zostało zredukowanych do jedynie 3 wierszy, a dwie kolumny, będące zmiennymi procesowymi zostały rozszerzone do 6 kolumn, co stanowi wszystkie kombinacje zmiennych procesowych i cech liczbowych (w tym przypadku  $2 \cdot 3$ ).

Na rysunku 4.3 przedstawiono graficzną reprezentację końcowego zbioru danych dla zmiennej procesowej 1 ( $zp_1$ ), który powstał po przekształceniu początkowych przebiegów za pomocą kontekstu i cech liczbowych. Tak przygotowany zbiór danych stanowi źródło do kolejnego kroku metody, którym jest rozpoznawanie stanu, realizowane poprzez zadanie klasyfikacji. W przytoczonym przykładzie wyjaśniono proces przekształcania danych jedynie dla pojedynczego eksperymentu przeprowadzonego na podstawie funkcji matematycznych, co jednak w pełni obrazuje sposób i poziom redukcji danych i w konsekwencji wpływu na skuteczność działania procesu trenowania klasyfikatorów. Niewielka liczba wierszy w danych może okazać się niewystarczająca do powstania satysfakcjonującego modelu klasyfikatora, dlatego trzeba zadbać o odpowiednią liczebność danych w zbiorze trenującym, np. poprzez wielokrotne powtarzanie procesu (eksperymentu). Inną koncepcją pozwalającą na rozszerzenie zbioru trenującego jest augmentacja. Augmentacja umożliwia generowanie nowych przykładów trenujących poprzez dodawanie zniekształceń do już dostępnych zbiorów danych. W przypadku zmiennych procesowych lub cech



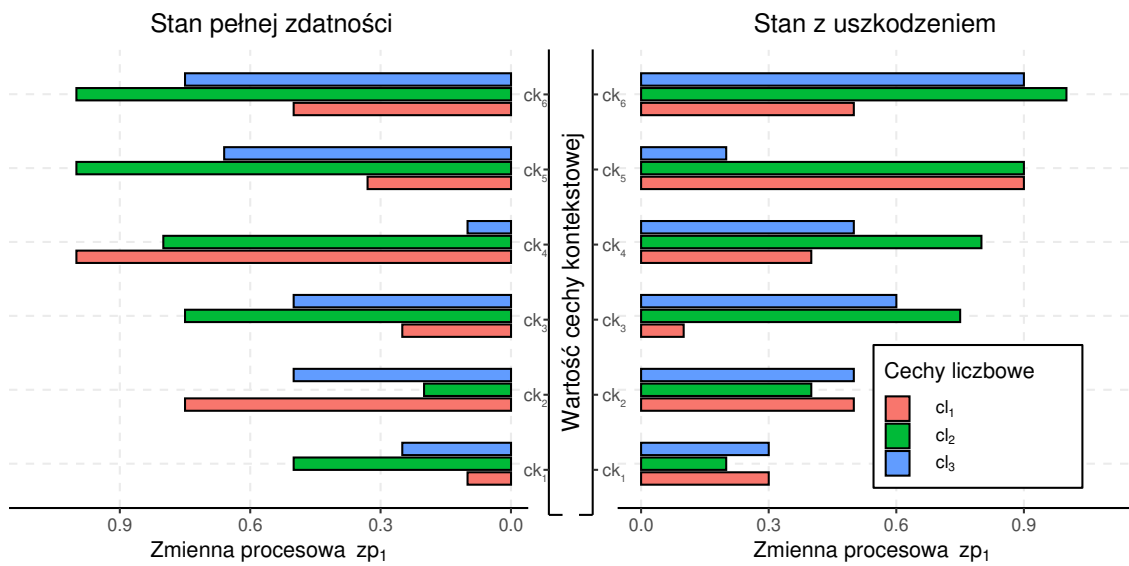
Rys. 4.3: Wynik przekształcenia zmiennej procesowej  $zp_1$  na zbiór wartości cech liczbowych  $cl_1, cl_2, cl_3$

sygnałów wyznaczonych w dziedzinie czasu, można wprowadzić do sygnału szum, który wpłynie na jego rozkład.

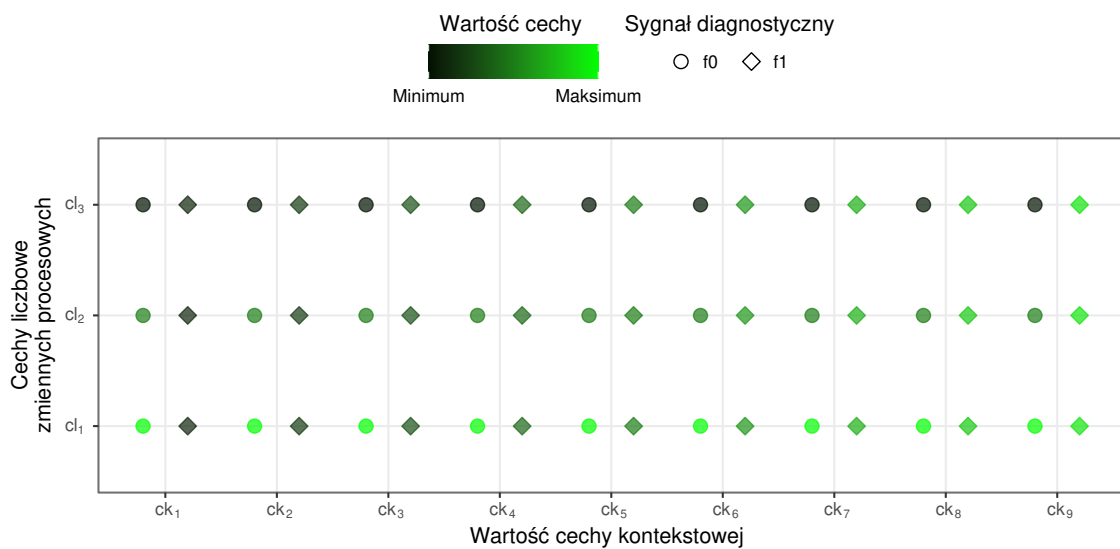
Na wyżej przedstawionym przykładzie pokazano jeden z możliwych sposobów wizualizacji danych składających się z wartości cech liczbowych obliczonych dla zmiennych procesowych pogrupowanych względem wartości cechy kontekstowej. Wykres słupkowy widoczny na rysunku 4.3 pozwala na wystarczająco czytelne porównanie poszczególnych wartości dla niewielkiej ich liczby. W przypadku większej liczby wartości czytelność takiego wykresu znacząco spadnie, dodatkowo uwzględniając stan procesu należałoby porównać wartości cech uzyskane dla stanu pełnej zdatności i z uszkodzeniem. W tym celu w dalszej części pracy zaproponowano rozwinięcie wykresu widocznego na rysunku 4.3 w taki sposób, aby w czytelny sposób zestawić ze sobą wartości cech dla obu rozpatrywanych stanów. Wyniki zaprezentowane na kolejnych wykresach mają charakter poglądowy i nie są w żaden sposób związane z poprzednimi wykresami i tabelami.

Na rysunku 4.4 przedstawiono przykład takiego wykresu, gdzie z lewej strony znajdują się wartości obliczone dla danych zebranych w czasie poprawnie działającego procesu, a po prawej stronie dane wartości dla stanu z uszkodzeniem. Na potrzeby prezentacji rozszerzono liczebność wartości cechy kontekstowej z 3 do 6 i jak można zauważyć na wykresie, interpretacja graficznej reprezentacji wartości jest prosta.

Jednak wykres pokazany na rysunku 4.4 obarczony jest pewnymi ograniczeniami. Na pojedynczym wykresie możliwe jest wyświetlenie danych dla pojedynczej zmiennej procesowej. Wykres tego typu wyraźnie odseparowuje od siebie wyniki będące rezultatem obliczeń dla kolejnych wariantów cechy kontekstowej bez względu na ich liczebność, jednak duża liczba cech znacząco wpływa na rozmiar wykresu i może negatywnie wpływać na separację wartości cech w ramach tego samego wariantu cechy kontekstowej. W wy-



Rys. 4.4: Sposób wizualizacji wartości obliczonych cech dla wybranej zmiennej procesowej z uwzględnieniem podziału na dane dotyczące stanu bez uszkodzenia i stanu z uszkodzeniem

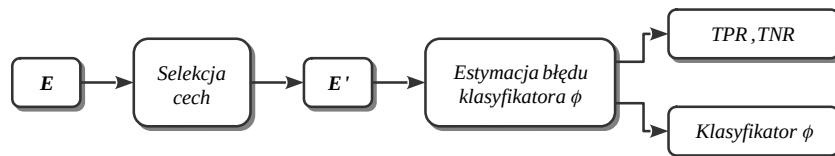


Rys. 4.5: Wizualizacja wartości cech wyliczonych w oparciu o wszystkie zmienne procesowe z rozróżnieniem na dane wskazujące stan pełnej zdatności i stan z uszkodzeniem

niku tych trudności zdecydowano się na opracowanie innej formy wizualizacji wyników, pozwalającej na pokazanie pełnego zestawu wyliczonych wartości cech dla wszystkich wartości cechy kontekstowej i zmiennych procesowych. Na rysunku 4.5 przedstawiono przykład polegający na porównaniu wartości konkretnej cechy liczbowej obliczonej dla zmiennej procesowej oraz dla cechy kontekstowej.

## 4.2. Zadanie klasyfikacji

Realizacja zadania klasyfikacji została przedstawiona na rysunku 4.6 i zaczyna się od utworzenia klasyfikatora  $\phi$ . W tym celu zastosowano metodę uczenia nadzorowanego przy zastosowaniu wybranych algorytmów klasyfikacji. Pierwszym krokiem realizacji tego zadania było utworzenie zbioru danych  $E$ , w którym każdy wektor cech został przypisany do określonego stanu. Tak przygotowane dane mogą zostać wykorzystane do wytrenowania klasyfikatora, lub mogą zostać poddane procesowi selekcji cech, który redukuje zbiór cech w danych do podzbioru  $E'$ . Wynikiem działania tego procesu są opisane wcześniej miary  $TPR$  i  $TNR$ . Aby uzyskać wiarygodną estymatę tych wskaźników, należy zastosować jedną z metod jak np. walidację krzyżową.



Rys. 4.6: Proces uczenia pojedynczego klasyfikatora

### 4.2.1. Selekcja cech

Przed wprowadzeniem danych trenujących do algorytmu uczenia klasyfikatora, dane poddano odpowiedniemu przygotowaniu. Jednym z kroków tej operacji jest selekcja cech w celu usunięcia nadmiarowej ich liczby (Jović *i in.*, 2015; Chandrashekar i Sahin, 2014). Duża liczba cech może znacząco wpłynąć na czas trwania procesu uczenia klasyfikatora jak i na jego rezultat. Ponadto nadmiarowa liczba cech, która w rzeczywistości nie wnosi żadnych użytecznych informacji może pogorszyć skuteczność działania niektórych typów klasyfikatorów, jak np. naiwny klasyfikator bayesowski, który bazuje na wartościach statystycznych i nie charakteryzuje się wbudowanym mechanizmem selekcji atrybutów tak jak np. w algorytmach drzewa decyzyjnego. Rozpatrywane są dwie grupy metod selekcji cech: z nauczycielem i bez nauczyciela. W rozprawie wykorzystano metodę selekcji cech z nauczycielem, gdzie konieczne jest posiadanie zbioru treningowego, w którym wektory danych trenujących są przypisane do znanych klas. Jedną z podstawowych metod jest zastosowanie wybranych miar wskaźników korelacji takich jak wskaźnik korelacji Pearsona, korelacji Spearmana, wskaźnik chi-kwadrat lub informacji wzajemnej. Możliwa jest do zastosowania każda metoda, pozwalająca ocenić poziom zależności pomiędzy cechami sygnałów a zmienną zależną reprezentującą rozpoznawaną klasę. Analizując wartości

poszczególnych wskaźników korelacji obliczonych dla każdej z cech, jesteśmy w stanie określić, które cechy są potencjalnie istotne lub nieistotne. Celowo użyto stwierdzenia potencjalnie, ponieważ faktyczna zależność klas w stosunku do danej cechy może nie spełniać założeń danej metody jak np. liniowość czy monotoniczność co w rezultacie może dawać niepoprawne rezultaty w kontekście istotności danej cechy.

Głównym celem zastosowania algorytmu selekcji jest wskazanie tych atrybutów, które w największym stopniu pozwalają na separację danych ze względu na ich przynależność do klas (stanów technicznych). Selekcja atrybutów jest jednym z pierwszych etapów realizacji procesu uczenia klasyfikatorów. Odpowiednio wyselekcjonowane atrybuty mogą w znacznym stopniu przyspieszyć zadanie trenowania klasyfikatorów ze względu na redukcję ilości danych przeznaczonych do trenowania klasyfikatora jednocześnie oczekuje się poprawy sprawności klasyfikacji poprzez wybranie atrybutów najbardziej relewantnych. W podrozdziale 4.2.2 wyjaśniono dokładniej metodę selekcji cech w połączeniu z algorytmem optymalizacji.

#### 4.2.2. Miary oceny sprawności klasyfikatorów

W rozdziale 2.3.3 opisano najpopularniejsze metody weryfikacji klasyfikatorów pod względem ich skuteczności rozpoznawania klas. Podstawowa forma macierzy pomyłek opisana w wyżej wymienionym rozdziale jest macierzą binarną i służy do zestawienia wyników działania klasyfikatora rozpoznającego dwie klasy. Badanie opisane w pracy dotyczy zadania detekcji uszkodzeń. Z tego powodu w dalszej części pracy postanowiono zmodyfikować schemat macierzy pomyłek tak, aby zamiast na klasy bezpośrednio wskazywała na stan bez uszkodzenia i z uszkodzeniem, co przedstawiono w tabeli 4.3.

Tab. 4.3: Macierz pomyłek dla zadania detekcji uszkodzeń

Stan rzeczywisty	Stan przewidywany	
	Klasa stanu z uszkodzeniem (FX)	Klasa stanu bez uszkodzenia (F0)
Klasa stanu z uszkodzeniem (FX)	Prawdziwie pozytywny (TP)	Fałszywie negatywny (FN)
Klasa stanu bez uszkodzenia (F0)	Fałszywie pozytywny (FP)	Prawdziwie negatywny (TN)

Miary oceny skuteczności działania klasyfikatora wymienione w rozdziale 2.3.3 są powszechnie wykorzystywane w diagnostyce procesów w celu oceny poszczególnych wariantów skuteczności lub błędów detekcji (Korbicz *i in.*, 2004; Wojtusik, 2006). Przy odpowiednim zdefiniowaniu klas w danych, miara  $TPR$  (2.15) odpowiada wskaźnikowi detekcji prawdziwych alarmów  $r_{td}$ , a miara  $TNR$  (2.16) odpowiada wskaźnikowi detekcji stanu bez awarii, natomiast miara  $FPR$  (2.15) pozwala wyznaczyć częstość wystąpienia fałszywych alarmów  $r_{fd}$ . Wskaźniki prawdziwych i fałszywych alarmów w zagadnieniach diagnostyki procesów wyraża się wzorami:

$$r_{fd} = \frac{\sum t_{fd}^i}{t_{from} - t_{on}} \quad (4.1)$$

$$r_{td} = \frac{\sum t_{td}^i}{t_{hor} - t_{from}} \quad (4.2)$$

gdzie  $t_{fd}^i$  jest  $i$ -tym wystąpieniem alarmu w przedziale czasu od  $t_{on}$  do  $t_{from}$ , a  $t_{td}^i$  jest  $i$ -tym wystąpieniem alarmu w przedziale czasu od  $t_{from}$  do  $t_{hor}$  (Bartyś *i in.*, 2006). Domyślnym celem działania algorytmu uczenia klasyfikatora jest dążenie do jak najwyższej jego sprawności poprzez maksymalizację wskaźnika dokładności (2.14) uzyskiwanego dla danych trenujących. W dalszej części pracy opisano metodę bazującą na algorytmie optymalizacji parametrów klasyfikatorów, którego celem jest uwzględnienie miar  $TPR$  i  $TNR$ , w celu ukierunkowania klasyfikatorów na rozpoznanie konkretnych stanów.

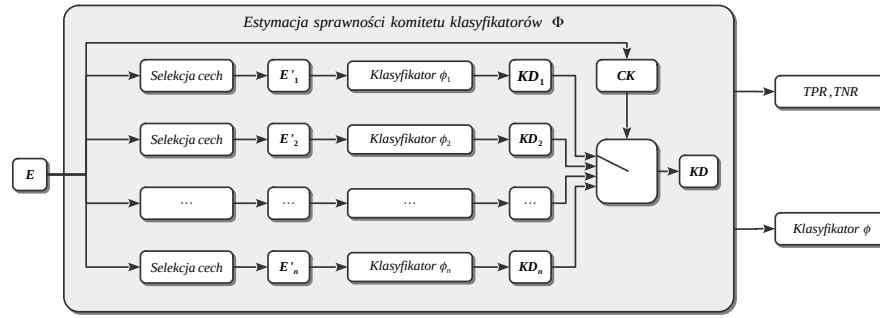
### 4.3. Fuzja klasyfikatorów z wykorzystaniem kontekstu

Jedną z podstawowych form fuzji klasyfikatorów jest fuzja na poziomie ich odpowiedzi w komitecie klasyfikatorów. Takie podejście pozwala na zastosowanie większej liczby klasyfikatorów. Na podstawie każdego z klasyfikatorów podejmowana jest niezależna decyzja, a następnie stosowana jest metoda selekcji ostatecznej odpowiedzi. W literaturze wyróżnić można szereg zaproponowanych metod pozwalających na realizację zadania klasyfikacji, bazującego na komitecie klasyfikatorów (Woźniak, 2006), jedną z nich jest metoda głosowania opisana w dalszej części tego rozdziału.

Wykorzystanie kontekstu w fuzji klasyfikatorów polega na zbudowaniu komitetu klasyfikatorów  $\Phi$ , składającego się z klasyfikatorów bazowych  $\phi_i$  zamiast pojedynczego klasyfikatora  $\phi$ . Cecha kontekstowa  $ck$  jest rozpatrywana niezależnie od cech podstawowych  $cp$  i pełni funkcję przełącznika na wyjściu komitetu. Podejście to zakłada, że cecha kontekstowa zawiera skończoną liczbę wartości dyskretnych, związanych z określonymi kontekstami działania procesu. Na rysunku 4.7 przedstawiono wariant klasyfikatora kontekstowego (komitetu klasyfikatorów uwzględniających kontekst), w którym cecha kontekstowa  $ck$  określa, z którego klasyfikatora brana pod uwagę jest decyzja  $kd$  i podawana na wyjście całego komitetu. Powiązanie pomiędzy wartością cechy kontekstowej a decyzją klasyfikatora wymaga analizy pod kątem sprawności poszczególnych klasyfikatorów w połączeniu z kontekstem.

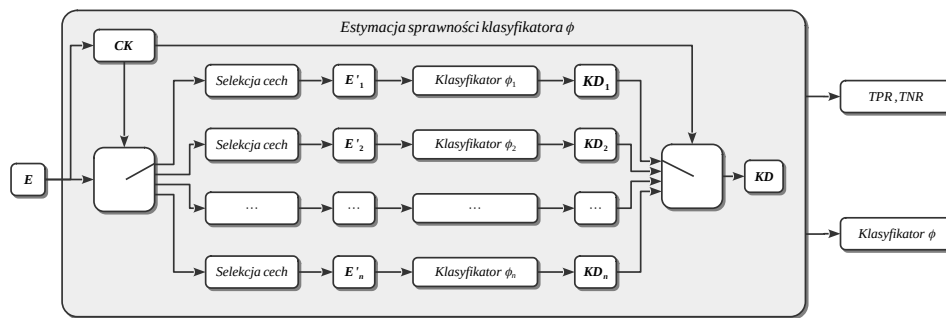
Podejście przedstawione na rysunku 4.7 opiera się na założeniu, że klasyfikatory wchodzące w skład komitetu  $\Phi$  są trenowane na pełnym zbiorze danych podawanym na wejście komitetu. Jedynym elementem dywersyfikującym dane pomiędzy klasyfikatorami jest metoda selekcji cech, która została zaimplementowana dla każdego klasyfikatora osobno. Istnieje też możliwość zastosowania w komitecie klasyfikatorów różnego typu i z różnymi parametrami trenowania. Na rysunku 4.8 rozszerzono tę koncepcję, poprzez





Rys. 4.7: Proces uczenia komitetu klasyfikatorów rozszerzony o selekcję cech

uwzględnienie cechy kontekstowej  $ck$  na początku klasyfikatora  $\Phi$ , przed etapem trenowania klasyfikatorów bazowych  $\phi_i$ . W ten sposób klasyfikatory w komitecie są trenowane na podzbiorach danych nie tylko zróżnicowanych pod względem zawartych w nich cech, ale również pod względem zestawu próbek uczących, ponieważ zastosowany na początku przełącznik, pozwala przekierowywać dane do odpowiedniego klasyfikatora w zależności od wartości kontekstu.



Rys. 4.8: Proces uczenia komitetu klasyfikatorów bazującego na kontekście

Schemat przedstawiony na rysunku 4.8 wymaga odpowiednio przygotowanych danych trenujących, których zestaw został przedstawiony poniżej:

$$\mathbf{E} = \begin{bmatrix} cp_{1,1} & cp_{2,1} & \cdots & cp_{N,1} & ck_{1,1} & ck_{2,1} & \cdots & ck_{M,1} & kd_1 \\ cp_{1,2} & cp_{2,2} & \cdots & cp_{N,2} & ck_{1,2} & ck_{2,2} & \cdots & ck_{M,2} & kd_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ cp_{1,K} & cp_{2,K} & \cdots & cp_{N,K} & ck_{1,K} & ck_{2,K} & \cdots & ck_{M,K} & kd_K \end{bmatrix}, \quad (4.3)$$

gdzie  $cp_{n,k}$  jest  $k$ -tą wartością  $n$ -tej cechy podstawowej,  $N$  jest liczbą cech podstawowych,  $ck_{m,k}$  jest  $k$ -tą wartością  $m$ -tej cechy kontekstowej,  $M$  jest liczbą cech kontekstowych,  $kd$  jest zmienną określającą końcową decyzję na potrzeby procesu trenowania,  $K$  jest liczbą wierszy danych. Dane wejściowe  $\mathbf{E}$ , klasyfikatora kontekstowego  $\Phi$ , są dzielone na podzbiory  $\mathbf{E}_i$  zależne od wartości cechy kontekstowej  $ck$  a następnie poddawane selekcji cech uzyskując w rezultacie podzbiór  $\mathbf{E}'_i$ . Dane wyjściowe klasyfikatorów są uzyskiwane poprzez zastosowanie następującej relacji:

$$[kd] = [kd_i] = \phi_i(\mathbf{E}'_i), \quad (4.4)$$

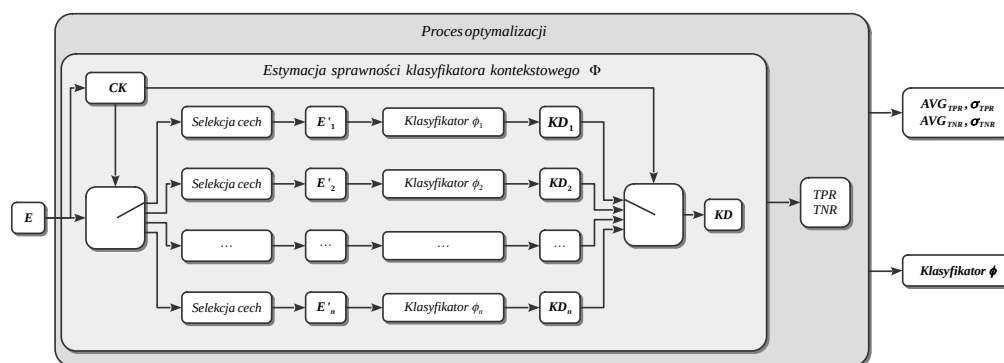
gdzie  $kd_i$  reprezentuje decyzję  $i$ -tego klasyfikatora, indeks  $i \in \{1, 2, \dots, n\}$  jest związany z wariantem (wartością) cechy kontekstowej  $ck$  i jest wykorzystany na etapie selekcji klasyfikatora dla określonego podzbioru danych wejściowych. Klasyfikatory bazowe są budowane za pomocą odpowiednich algorytmów trenujących:

$$\chi_i : \{\mathbf{p}_i, \mathbf{E}'_i\} \rightarrow \phi_i, \quad (4.5)$$

gdzie  $\chi_i$  reprezentuje funkcję uczącą, która jest niezbędna do utworzenia  $i$ -tego klasyfikatora,  $\mathbf{p}_i$  jest zbiorem parametrów wpływających na sposób działania  $i$ -tej funkcji uczącej,  $\mathbf{E}'_i$  jest podzbiorem danych uczących, które zostały wyszczególnione z całego zbioru danych  $\mathbf{E}$  za pomocą cechy kontekstowej.

#### 4.4. Optymalizacja parametrów klasyfikatora kontekstowego

Kolejnym rozwinięciem zaprezentowanego wcześniej schematu jest dodanie procesu optymalizacji, którego zadaniem jest dostrojenie parametrów klasyfikatorów i procesu uczenia w taki sposób, aby cały komitet w jak najwyższym stopniu zbliżał się do rozwiązania optymalnego, tj. maksymalizacji wskaźników mierzących skuteczność detekcji stanu pełnej zdatności i stanu z uszkodzeniem oraz maksymalizację powtarzalności uzyskiwanych przez klasyfikator wyników. Schemat uwzględniający algorytm optymalizacji został przedstawiony na rys. 4.9.



Rys. 4.9: Proces uczenia komitetu klasyfikatorów bazującego na kontekście z uwzględnieniem optymalizacji

Skuteczność zaproponowanej metody detekcji uszkodzeń bazującej na kontekście w znacznym stopniu zależy od struktury ( $\mathcal{M}$ ) i parametrów ( $\mathbf{p}$ ) klasyfikatorów wchodzących w skład komitetu pokazanego na rysunku 4.8. Elementy te mają bardzo duży wpływ na proces detekcji uszkodzeń i zostać dobrane prawidłowo. Głównym celem zastosowania

algorytmu optymalizacji jest znalezienie optymalnej struktury klasyfikatora oraz optymalnych wartości parametrów algorytmu uczącego. W zbiorze parametrów uczących  $\mathbf{p}$  znajdują się parametry algorytmów treningowych klasyfikatorów oraz parametry określające, które cechy danych treningowych są podawane na wejście klasyfikatora. Zbiór tych parametrów można zapisać za pomocą wyrażenia  $p_1, p_2, \dots, p_N, p_{N+1}, p_{N+2}, \dots, p_{N+M}$ , gdzie  $N$  określa liczbę parametrów algorytmu trenującego klasyfikatory, a  $M$  definiuje liczbę parametrów dotyczących selekcji cech. Problem ten jest rozpatrywany jako zadanie optymalizacji wielokryterialnej, które może zostać opisane za pomocą następującego równania:

$$\text{Minimalizacja } \mathbf{C}(\mathcal{M}, \mathbf{p}) = \min \left[ c_1(\mathcal{M}, \mathbf{p}) \quad c_2(\mathcal{M}, \mathbf{p}) \quad \dots \quad c_k(\mathcal{M}, \mathbf{p}) \right], \quad (4.6)$$

w przestrzeni  $\Omega(\mathcal{M}, \mathbf{p}, \mathbf{C})$ ,

gdzie  $c_i$  jest  $i$ -tą funkcją kryterialną, która reprezentuje miarę sprawności detekcji uszkodzeń (należy podkreślić, że funkcje kryterialne reprezentują cele, które nie są ze sobą w konflikcie),  $\Omega$  określa przestrzeń przeszukiwania określoną za pomocą ograniczeń i granic, które powinny zostać określone na podstawie zakresów parametrów i możliwości metod uczących, danych trenujących i testujących, itp. W celu zmniejszenia złożoności problemu możliwe jest wpłynięcie na strukturę schematu komitetu klasyfikatorów ( $\mathcal{M}$ ) poprzez zmianę wartości niektórych parametrów  $\mathbf{p}$  dostępnych w algorytmie uczącym. Ostateczna liczba optymalizowanych parametrów zależy od zastosowanego algorytmu klasyfikatora oraz ogólnej struktury całego schematu algorytmu trenującego.

#### 4.4.1. Funkcje kryterialne

Funkcja celu  $\mathbf{C}$  może zostać opracowana w oparciu o różne założenia i kryteria. W przypadku detekcji uszkodzeń bazującej na klasyfikatorach, jednym z rezultatów uzyskiwanych podczas trenowania i testowania jest macierz pomyłek (Tab. 2.1), która pozwala na wyznaczenie szeregu miar i wskaźników oceniających skuteczność działania klasyfikatora lub całego komitetu w odniesieniu do określonych wymagań. Dostępne miary mogą zostać użyte bezpośrednio w konstrukcji funkcji celu, co przekłada się na wpływ wyników działania klasyfikatora na wartość zwracaną przez funkcję celu.

Ponieważ zadanie zostało ograniczone do problemu detekcji uszkodzeń, w procesie optymalizacji zastosowane zostały dwie funkcje kryterialne  $c_1$  i  $c_2$ , związane kolejno ze stanem z uszkodzeniem i stanem bez uszkodzenia. Dzięki takiemu podejściu, każdy ze stanów może być traktowany indywidualnie i pełnić istotną rolę w procesie poszukiwania rozwiązań optymalnych. Zastosowano dwa podejścia do opracowania funkcji celu. W obu przypadkach głównymi rozpatrywanymi miarami obliczanymi z wyników uzyskanych w postaci macierzy pomyłek są czułość (TPR) i swoistość (TNR), czyli odpowiednio wskaźnik poprawnie zidentyfikowanych stanów z uszkodzeniem i w przypadku TNR bez uszkodzenia. Proces trenowania i testowania klasyfikatorów w ramach pojedynczego osobnika w algorytmie optymalizacji jest realizowany wielokrotnie w celu uzyskania

zbioru wymienionych wcześniej rozwiązań (TPR, TNR), które pozwalają na obliczenie ich wartości średnich oraz odchyłeń standardowych. Te dwa główne elementy stanowią najistotniejszą część zaproponowanych funkcji kryterialnych. Pierwsze z podejść bazuje na parametrze  $t_\alpha$ , który determinuje stopień istotności odchylenia standardowego w funkcji. Postać funkcji kryterialnych została zaprezentowana w postaci następujących równań:

$$c_1 = c_{TPR} = \frac{1}{N} \sum_{n=1}^N (1 - TPR_i) + t_\alpha \sigma_{TPR} \quad (4.7)$$

$$c_2 = c_{TNR} = \frac{1}{N} \sum_{n=1}^N (1 - TNR_i) + t_\alpha \sigma_{TNR} \quad (4.8)$$

gdzie  $N$  jest liczbą wywołań zadania trenowania i testowania klasyfikatorów w ramach pojedynczego wywołania funkcji celu,  $TPR_i / TNR_i$  określają skuteczność detekcji stanu z uszkodzeniem i bez uszkodzenia w  $i$ -tej iteracji zadania trenowania i testowania klasyfikatorów,  $\sigma_{TPR} / \sigma_{TNR}$  opisują odchylenie standardowe uzyskanych wyników ( $TPR$ ,  $TNR$ ) w zakresie wszystkich iteracji w ramach pojedynczego wywołania funkcji celu a  $t_\alpha$  definiuje istotność odchylenia standardowego w funkcji. Wyższa wartość parametru  $t_\alpha$  spowoduje większy wpływ wartości odchylenia standardowego na wynik działania funkcji celu. W trakcie realizacji zadania trenowania klasyfikatorów dąży się do uzyskania modelu klasyfikatora, którego wyniki w postaci miar  $TPR$  i  $TNR$  są maksymalizowane. Funkcja celu natomiast jest przez algorytm optymalizacji minimalizowana przez co konieczne jest dopełnienie wartości tych miar do 1 poprzez działanie  $1 - TPR$  i  $1 - TNR$ .

Drugie podejście jest oparte na wprowadzeniu wartości wagowej, która wpływa w odwrotny sposób na dwa główne elementy funkcji celu, czyli wartość średnią skuteczności detekcji danej klasy, jak i odchylenia standardowego. Domyślnie wartość parametru  $w$  steruje istotnością wartości średniej, a dopełnienie parametru  $w$  udziałem odchylenia standardowego, co zostało przedstawione poprzez poniższe równania:

$$c_1 = c_{TPR} = w \left( \frac{1}{N} \sum_{n=1}^N (1 - TPR_i) \right) + (1 - w) \sigma_{TPR} \quad (4.9)$$

$$c_2 = c_{TNR} = w \left( \frac{1}{N} \sum_{n=1}^N (1 - TNR_i) \right) + (1 - w) \sigma_{TNR} \quad (4.10)$$

gdzie  $N$  jest liczbą wywołań zadania trenowania i testowania klasyfikatorów w ramach pojedynczego wywołania funkcji celu,  $TPR_i / TNR_i$  określają skuteczność detekcji stanu z uszkodzeniem i bez uszkodzenia w  $i$ -tej iteracji zadania trenowania i testowania klasyfikatorów,  $\sigma_{TPR} / \sigma_{TNR}$  opisują odchylenie standardowe uzyskanych wyników ( $TPR$ ,  $TNR$ ) w zakresie wszystkich iteracji w ramach pojedynczego wywołania funkcji celu a  $w$  definiuje udział poszczególnych składowych równania i przyjmuje wartości z zakresu  $\langle 0, 1 \rangle$ .

Można wyróżnić trzy charakterystyczne warianty funkcji kryterialnych, w zależności od przyjętej wartości parametru  $w$ :

- $w = 1$ : jedynie wartości średnie miar  $TPR$  i  $TNR$  są brane pod uwagę w funkcjach kryterialnych, odchylenie standardowe jest ignorowane,
- $w = 0$ : wartość średnia jest ignorowana, a na ostateczny wynik funkcji kryterialnych ma wpływ jedynie odchylenie standardowe,
- $w = 0.5$ : wartość średnia jak i odchylenie standardowe biorą w równym stopniu udział w procesie obliczania wartości funkcji kryterialnych.

Wszystkie pozostałe wartości w proporcjonalnym stopniu modyfikują udział obu składowych funkcji w jej końcowym wyniku.

#### 4.4.2. Optymalizacja liczby atrybutów

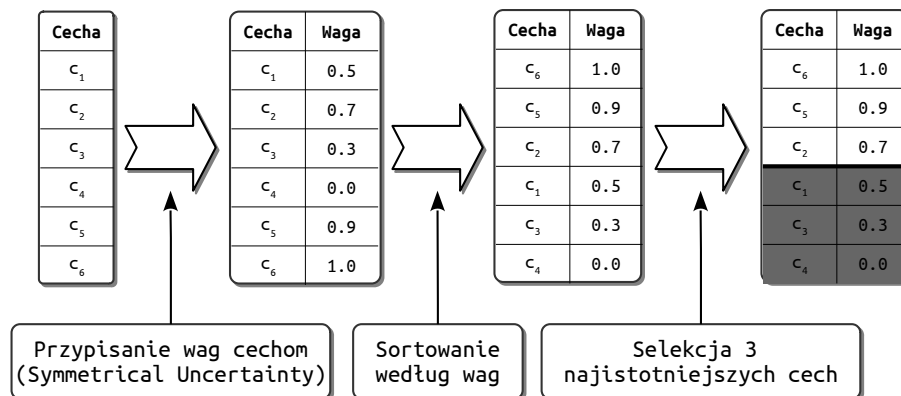
Oprócz parametrów klasyfikatorów optymalizacji podlega również liczba cech podstawowych podawanych na wejście algorytmów trenujących. Oczekuje się, że ograniczenie liczby cech wpłynie pozytywnie na skuteczność detekcji uszkodzeń za pomocą zastosowanego komitetu klasyfikatorów. Selekcja cech może być zastosowana na różnych etapach opisanego wcześniej procesu trenowania klasyfikatorów. Jednym z tych etapów jest początek procesu, w którym dostępny jest pełny zbiór cech podawany w kolejnych etapach na wejście klasyfikatorów. Jednak późniejszy podział początkowego zbioru danych trenujących na podzbiory powiązane z odrębnymi wartościami kontekstu nie gwarantuje, że ten sam podzbiór wyznaczony na podstawie pełnego zbioru będzie optymalny dla wszystkich klasyfikatorów wchodzących w skład komitetu. Z tego powodu postanowiono zastosować niezależne algorytmy selekcji cech dla każdego z klasyfikatorów w Komitecie. Dzięki takiemu podejściu możliwe jest dokonanie selekcji na każdym z podzbiorów danych osobno.

Drugim zadaniem związanym z selekcją jest zastosowanie odpowiedniego algorytmu selekcji. W literaturze jest dostępnych wiele podejść i algorytmów przeznaczonych do realizacji tego zadania, co zostało opisane w rozdziale 4.2.1. Jedną z podstawowych metod bazujących na systematycznym przeszukiwaniu, która pozwala na wskazanie zbioru cech, które maksymalizowałyby skuteczność klasyfikacji dla określonego zbioru danych trenujących. Jednak czas realizacji takiego procesu związany z każdorazowym uruchomieniem zadania trenowania klasyfikatora i jego przetestowaniem nie jest akceptowalny. Liczba wszystkich kombinacji cech (bez stanu, w którym wszystkie cechy są pomijane) jest równa  $2^x - 1$ , gdzie  $x$  jest równe liczbie wszystkich cech wchodzących w skład zbioru danych wejściowych, co w przypadku 52 cech oznacza w przybliżeniu  $45 \times 10^{14}$  wywołań algorytmu uczenia klasyfikatorów.

W opisywanej metodzie stosowany jest algorytm optymalizacji wielokryterialnej, który oprócz dostrajania wartości parametrów algorytmu trenującego, może być wykorzystany do wyznaczenia optymalnego zbioru cech w danych uczących. Jedną z możliwości jest utworzenie binarnego wektora o długości równej liczbie cech, którego wartości określają dołączenie lub wyłączenie wybranej cechy do zbioru danych trenujących. Jest to metoda zbliżona do metody systematycznego przeszukiwania jednak pozwala na ograniczenie

przestrzeni przeszukiwania tylko do tych obszarów, które znajdują się w okolicy minimów funkcji celu.

Istnieją również metody, które pozwalają na ocenę poszczególnych cech dzięki zastosowaniu określonych metod bazujących np. na współczynniku korelacji pomiędzy wartością cechy, a etykietą klasy, co zostało szerzej opisane w rozdziale 4.2.1. Na potrzeby realizacji badania opisanego w niniejszej rozprawie zaproponowano rozwiązanie polegające na połączeniu metody oceniającej istotność cechy w stosunku do klasy z metodą optymalizacji. W tym celu dla każdej podgrupy danych przypisanej do określonego kontekstu, a co za tym idzie też klasyfikatora, zastosowano algorytm bazujący na współczynnikach korelacji, a dokładniej, na symetrycznej niepewności (*ang. symmetrical uncertainty*) (Ziemia, 2012; Senthamarai Kannan i Ramaraj, 2010). Wynikiem działania algorytmu są znormalizowane wagi (z zakresu od 0 do 1) przypisane do każdej z cech, gdzie 1 oznacza najsilniejszą korelację pomiędzy cechą a klasą. Wartość 0 najniższą korelację spośród wszystkich rozpatrywanych cech. Wagi pozwalają na uszeregowanie cech od najbardziej do najmniej istotnych. Kolejnym elementem tej metody jest dodanie do przestrzeni optymalizowanych wartości algorytmu genetycznego wartości reprezentującej liczbę cech z posortowanej listy, zaczynając od najistotniejszej cechy, które następnie są brane pod uwagę w dalszym procesie trenowania klasyfikatora. Zarówno zadanie przypisywania wag istotności cech jak i określenie liczby branych pod uwagę cech jest realizowane osobno dla każdego z klasyfikatorów w komitecie. Realizacja tego zadania została pokazana na rysunku 4.10.



Rys. 4.10: Schemat procesu selekcji atrybutów z zastosowaniem algorytmu korelacyjnej oceny istotności cech

W ten sposób zainicjować można wektor parametrów (wchodzących w skład chromosomu) dostrajanych przez algorytm optymalizacji, gdzie każdy pojedynczy parametr jest liczbą najistotniejszych (z punktu widzenia korelacji) cech wyznaczonych na podstawie podzbioru danych związanych z określonym kontekstem. Liczba tych parametrów w wektorze jest równa liczbie rozpatrywanych wartości kontekstu. Chromosom jest strukturą danych, która reprezentuje konkretny zestaw decyzji lub parametrów, które definiują dane rozwiązanie problemu. Każda cecha lub decyzja reprezentowana przez chromosom jest nazywana genem. Konkretna reprezentacja chromosomu zależy od problemu, który

jest rozwiązywany. Dla problemu doboru parametrów treningowych klasyfikatorów oraz liczby cech, w genach zawarto wartości poszczególnych parametrów oraz liczbę najistotniejszych cech stanowiących wejście algorytmu trenującego dany klasyfikator. Budowa chromosomu prezentuje się następująco:

$$ch = [x_1, \dots, x_n] \quad (4.11)$$

gdzie  $x$  określa liczbę cech branych pod uwagę, a  $n$  liczbę wariantów kontekstu. W kolejnym kroku należy uzupełnić wektor optymalizowanych parametrów o parametry stanowiące podstawę działania określonych algorytmów trenowania klasyfikatorów.

#### 4.4.3. Dobór parametrów klasyfikatorów

Większość klasyfikatorów, a dokładniej algorytmów trenowania klasyfikatorów, posiada zbiór parametrów, które pozwalają na sterowanie procesem trenowania klasyfikatora. W przypadku metody  $k$ -najbliższych sąsiadów głównym parametrem jest zmienna  $k$ , która określa liczbę najbliższych sąsiadów w celu wyznaczenia podobieństwa. Dla drzewa decyzyjnego dostępne są parametry, które definiują np. warunki tworzenia nowych gałęzi czy definiujące proces przycinania drzewa. Każda metoda klasyfikacji, która w swoich założeniach może znacznie różnić się od innych metod, charakteryzuje się indywidualnym zbiorem parametrów. Dlatego wymagane jest, aby w trakcie implementacji procesu optymalizacji, zbiór optymalizowanych parametrów był każdorazowo dostosowany do konkretnej metody klasyfikacji, dotyczy to zarówno liczby parametrów, ich charakteru (np. wartości rzeczywiste, całkowite, binarne, słownikowe, itp.) jak i ich ograniczeń. Bazując na wyżej przedstawionym modelu chromosomu zawierającym jedynie parametry dotyczące liczby cech 4.11, należy dodać kolejne związane z parametrami klasyfikatorów:

$$ch = [x_1, \dots, x_n, y_{1,1}, \dots, y_{m,n}] \quad (4.12)$$

gdzie  $x$  ponownie jest liczbą najistotniejszych cech biorących udział w procesie trenowania klasyfikatora,  $y$  wartością określonego parametru algorytmu trenowania klasyfikatora,  $n$  jest liczbą wariantów kontekstu, a  $m$  liczbą parametrów klasyfikatora biorących udział w procesie trenowania.

### 4.5. Poszukiwanie rozwiązania optymalnego

Optymalizacja z zastosowaniem algorytmów genetycznych wywodzi się ze zjawisk zaobserwowanych w przyrodzie, bazujących na doborze naturalnym i dziedziczeniu. Ogólna idea polega na obserwacji, że w naturze największe szanse na przetrwanie mają osobniki najlepiej przystosowane do środowiska i otoczenia natomiast osobniki przystosowane w najmniejszym stopniu są eliminowane. Jednocześnie osobniki najlepiej przystosowane mają też największą szansę na przekazanie informacji genetycznej osobnikom funkcjonującym w kolejnym pokoleniu (Rutkowski, 2005). W tym procesie zachodzą także operacje,

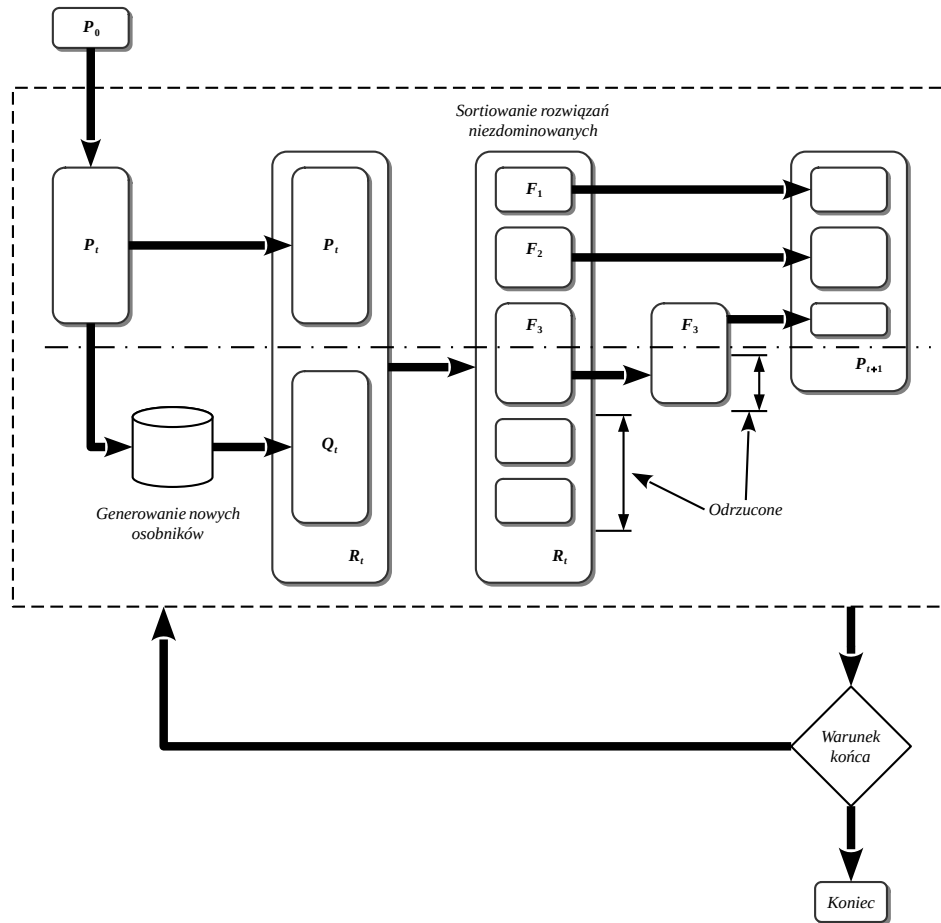
które również mają swój udział w metodach genetycznych i ewolucyjnych procesu optymalizacji. Operacje te funkcjonują w obszarze algorytmów optymalizacji jako operatory. Do najważniejszych operatorów zaliczyć należy:

- Operator selekcji - Selekcja pozwala określić metodę doboru osobników, wykorzystywanych do utworzenia kolejnego pokolenia osobników. Głównym kryterium wyboru osobników jest wartość funkcji celu. Do najpopularniejszych metod należy np. metoda ruletki, gdzie tak jak w przypadku gry w ruletkę, poszczególne obszary koła ruletki są przypisane do określonych osobników. Jednak udział osobnika w tym kole jest proporcjonalny do jego wartości funkcji celu przy czym wszystkie osobniki biorą udział w losowaniu, nawet te, które uzyskały najślabszy wynik.
- Operator krzyżowania - Idea działania genetycznych algorytmów optymalizacji polega na generowaniu kolejnych pokoleń, których osobniki są rezultatem działań na pokoleniu poprzednim. W tym celu stosuje się operator krzyżowania. Pierwszym etapem tego procesu jest wybór par chromosomów z populacji rodzicielskiej. Jest to tymczasowa populacja złożona z chromosomów wybranych metodą selekcji i przeznaczonych do dalszego przetwarzania za pomocą operatorów krzyżowania i mutacji w celu utworzenia nowej populacji potomków. Na tym etapie chromosomy z populacji rodzicielskiej kojarzą w pary. Dokonuje się tego w sposób losowy, zgodnie z prawdopodobieństwem krzyżowania  $p_c$ .
- Operator mutacji - Chromosom jest nośnikiem informacji na temat osobnika, gdzie informacja ta jest zapisana w postaci ciągu bitów. Mutacja polega na zmianie wartości wybranego bitu w chromosomie na przeciwny. To jak duży wpływ ma operator mutacji na proces tworzenia nowego pokolenia osobników zależy od prawdopodobieństwa mutacji  $p_m$ . Prawdopodobieństwo zaistnienia mutacji jest zwykle bardzo małe i od niego zależy, czy dany gen w chromosomie podlega mutacji, czy też nie. Wysokie prawdopodobieństwo mutacji może zbliżyć działanie algorytmu optymalizacji do działania metody losowego przeszukiwania przestrzeni rozwiązań.

Do wyznaczenia rozwiązania optymalnego zastosowano metodę optymalizacji wielokryterialnej NSGA-II (ang. *non-dominated sorting genetic algorithm*) (Deb i in., 2002). Idea działania tego algorytmu polega na podzieleniu osobników w podpopulacje, ze względu na nadane im rangi. Proces działania tej metody został przedstawiony na rysunku 4.11.

W pierwszym kroku, przed uruchomieniem właściwej części algorytmu tworzony jest zbiór osobników początkowych ( $\mathbf{P}_0$ ), odbywa się to najczęściej poprzez zainicjowanie wartości wszystkich zmiennych wchodzących w skład chromosomu wartościami losowymi. Na początku realizacji procesu szukania rozwiązań optymalnych pokolenia  $t$ , na podstawie początkowego zbioru osobników  $\mathbf{P}_t$  o liczebności  $N$  tworzony jest zbiór osobników potomnych  $\mathbf{Q}_t$  (również o liczebności  $N$ ) poprzez stosowanie metod krzyżowania i mutacji, w wyniku czego otrzymujemy zbiór wszystkich rozwiązań  $\mathbf{R}_t$  danego pokolenia  $t$ , o całkowitej liczebności  $2N$ . Kolejnym krokiem jest znalezienie wśród osobników zbioru  $\mathbf{R}_t$  osobników niezdominowanych, których przypisuje się do pierwszego frontu rozwiązań



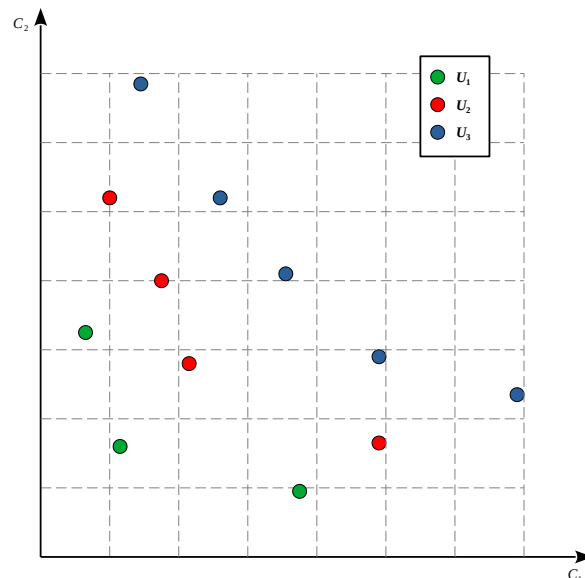


Rys. 4.11: Schemat działania algorytmu optymalizacji wielokryterialnej NSGA-II (Deb *et al.*, 2002)

$U_1$ . Następnie grupę osobników niezdominowanych należących do frontu  $U_1$  ignoruje się tymczasowo, a resztę populacji poddaje się temu samemu mechanizmowi i wyznacza się drugą warstwę rozwiązań niezdominowanych. Nowy zbiór osobników niezdominowanych przypisuje się do kolejnego frontu  $U_2$ . Ten proces jest powtarzany dopóki cała populacja nie zostanie sklasyfikowana. Na rysunku 4.12 przedstawiono graficzną reprezentację wygenerowanych podpopulacji rozwiązań niezdominowanych dla kolejnych frontów.

Tak uzyskane grupy zgodnie ze schematem pokazanym na rysunku 4.11 są wykorzystywane do utworzenia kolejnego pokolenia osobników. W przypadku gdy spełniony zostanie warunek zatrzymania działania algorytmu (np. uzyskanie wymaganej liczby pokoleń), algorytm zwróci zbiór rozwiązań optymalnych, należących do grupy wyników niezdominowanych, leżących na głównym froncie pareto. Jeżeli warunek zatrzymania procesu optymalizacji nie został jeszcze osiągnięty, wybrana część rozwiązań jest uwzględniana w procesie tworzenia kolejnego pokolenia osobników i ponownego uruchomienia wyżej opisanego procesu.

Wynikiem działania procesu optymalizacji jest zbiór rozwiązań paretooptymalnych. Każde z tych rozwiązań spełnia kryteria funkcji kryterialnych i pozwala na zbudowanie

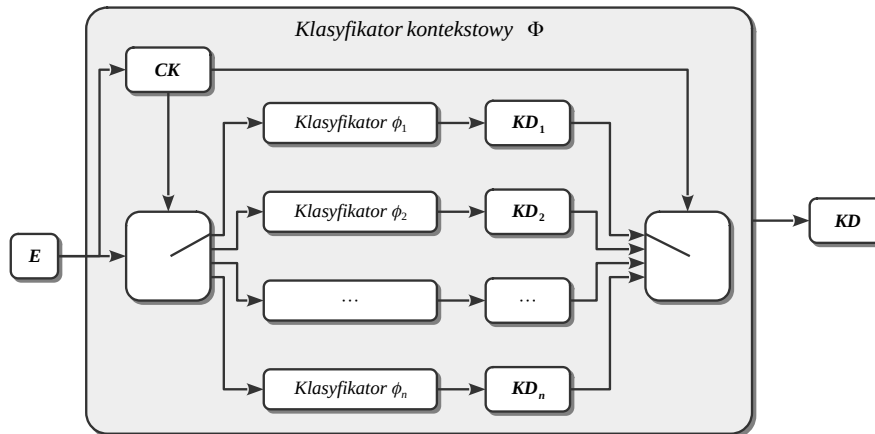


Rys. 4.12: Graficzna reprezentacja podziału rozwiązań na niezdominowane podpopulacje klasyfikatora kontekstowego w oparciu o optymalny zbiór parametrów  $\mathbf{p}$ . Implementacja oraz walidacja wybranego rozwiązania optymalnego została opisana w dalszej części doktoratu.

## 4.6. Implementacja klasyfikatora w procesie diagnostycznym

W wyniku analizy wszystkich otrzymanych rozwiązań optymalnych, należy wybrać rozwiązanie, które posłuży w kolejnym kroku do utworzenia docelowego klasyfikatora kontekstowego. Do każdego rozwiązania przypisana jest lista optymalnych wartości strojonych parametrów  $\mathbf{p}$ . Należy utworzyć proces wnioskowania, w którym należy umieścić klasyfikator kontekstowy. Związane jest to z tym, że parametry klasyfikatorów bazowych wchodzących w skład klasyfikatora kontekstowego, należy ustawić na wartości otrzymane z wybranego rozwiązania optymalnego. W celu przeprowadzenia końcowej walidacji działania wybranego rozwiązania, należy przygotować dwa niezależne zbiory danych, zbiór danych treningowych  $\mathbf{E}_{tr}$  i zbiór danych testowych  $\mathbf{E}_{ts}$ . Klasyfikator kontekstowy  $\Phi$  wytrenowany za pomocą zbioru danych treningowych  $\mathbf{E}_{tr}$  ma postać schematu pokazanego na rysunku 4.13. Klasyfikator taki jest gotowy do realizacji procesu wnioskowania za pomocą nowych danych.

Wynikiem zastosowania takiego klasyfikatora jest decyzja  $kd$  określająca, czy wektor danych wejściowych dotyczy stanu pełnej zdatności czy stanu z uszkodzeniem monitorowanego procesu. Przygotowane wcześniej dane walidacyjne  $\mathbf{E}_{ts}$ , mogą posłużyć do oceny, czy klasyfikator kontekstowy zbudowany w oparciu o wartości parametrów z wybranego rozwiązania optymalnego zwraca satysfakcjonujące wskaźniki wartości  $TPR$  i  $TNR$ . W celu weryfikacji powtarzalności wyników uzyskiwanych przez klasyfikator, należy przygotować większą liczbę par zbiorów danych treningowych  $\mathbf{E}_{tr}$  i testowych



Rys. 4.13: Schemat klasyfikatora kontekstowego gotowego do klasyfikacji nowych danych.

Etts lub skorzystać z jednej z dostępnych metod walidacji jak np. walidacja krzyżowa. Po zakończeniu procesu walidacji, utworzony klasyfikator jest gotowy do implementacji w docelowym środowisku monitorującym badany proces.

## 4.7. Podsumowanie

W celu wykorzystania metody diagnozowania z zastosowaniem kontekstu, konieczna jest realizacja szeregu operacji, przygotowujących dane w celu poprawnej implementacji zaproponowanej metody. W powyższym rozdziale opisano wszystkie niezbędne elementy zaczynając od akwizycji danych i ekstrakcji cech. Jak opisano dane mogą pochodzić z systemu monitorującego sygnały będące wynikiem pracy rzeczywistego obiektu, jak i z modelu matematycznego symulującego działanie takiego obiektu. Kluczowym elementem jest zastosowanie odpowiedniej metody klasyfikacji. Aby skrócić czas trwania procesu trenowania klasyfikatora oraz poprawić skuteczność klasyfikacji algorytmu stosuje się metodę selekcji atrybutów. Następnie wymieniono i opisano najczęściej stosowane miary oceny sprawności klasyfikatora, możliwe do wyznaczenia na podstawie wartości wymienionych w macierzy pomyłek. Wybrana metoda uwzględniająca kontekst w procesie detekcji uszkodzeń wymaga zastosowania komitetu klasyfikatorów. W kolejnych częściach rozdziału przedstawiono proces rozbudowy elementarnego schematu bazującego na pojedynczym klasyfikatorze poprzez dodanie kolejnych klasyfikatorów tworząc komitet oraz elementów związanych z kontekstem, aż do końcowego schematu. Dobór parametrów algorytmu trenującego klasyfikatory bazowe będące częścią klasyfikatora kontekstowego oraz określenie zbioru cech wchodzących w zbiór danych trenujących jest realizowany pod kontrolą algorytmu optymalizacji wielokryterialnej. Zagadnienie optymalizacji zostało dokładnie poruszone w końcowej części rozdziału. W ostatnim punkcie rozdziału opisano proces implementacji klasyfikatora w oparciu o parametry uzyskane w wyniku optymalizacji oraz omówiono problem jego walidacji oraz wdrożenia.



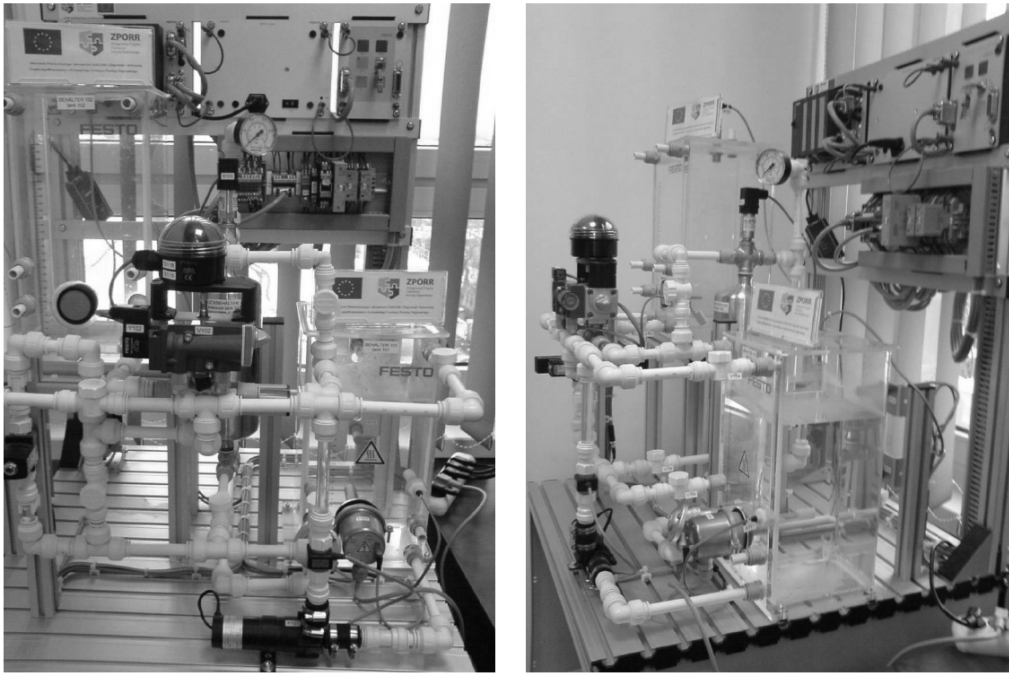
## Rozdział 5

# Badania weryfikacyjne

Badania weryfikacyjne zostały zrealizowane z zastosowaniem stanowiska laboratoryjnego FESTO S7 EduTrainer Compact Siemens S7-300 CPU313C, które pozwala na symulowanie procesów przemysłowych, głównie takich, które realizowane są w instalacjach, w których przesyłane są różne media, na przykład ciecze. Stanowisko jest instalacją składającą się z elementów instalacji hydraulicznych, takich jak pompy, zawory czy zbiorniki. Stanowisko jest częścią pracowni Diagnostyki procesów przemysłowych w Katedrze Podstaw Konstrukcji Maszyn na wydziale Mechanicznym Technologicznym Politechniki Śląskiej. Instalacja ta jest powszechnie wykorzystywana przez studentów i pracowników katedry w celu prowadzenia badań w zakresie diagnostyki procesów (Cholewa, 2010; Skupnik, 2009; Amarowicz, 2015; Mync, 2017; Hańderek, 2020; Kalisch *i in.*, 2020).

### 5.1. Obiekt badań

Stanowisko laboratoryjne, które zostało zastosowane w badaniach weryfikacyjnych opracowanej metody, składa się z szeregu elementów takich jak zawory, zbiorniki, czujniki czy elementy wykonawcze, zostało przedstawione na Rys. 5.1. Wszystkie wymienione elementy wchodzące w skład stanowiska przedstawionym na Rys. 5.2. Główne elementy to trzy zbiorniki (T1, T2 i T3), gdzie zbiorniki T1 i T2 pełnią rolę rezerwuarów wody, a zbiornik T3 pełni funkcję akumulatora hydraulicznego w układzie, ponieważ pozwala na gromadzenie energii hydraulicznej poprzez sprężanie powietrza znajdującego się wewnątrz. Zbiornik T1 jest połączony z pompą P, która pozwala na wtłoczenie wody do jednego z wymienionych zbiorników. Wybór miejsca docelowego strumienia wody określany jest za pomocą ręcznych zaworów (V1 - V10). Zawory można ustawić w pozycji całkowicie zamkniętej oraz całkowicie otwartej lub jako częściowo otwarte, co symuluje stany pośrednie. Zastosowanie tej techniki pozwala na wprowadzanie do układu uszkodzeń i nieprawidłowości. Zmiana położenia jednego z zaworów tak, aby jedynie częściowo zamykał możliwość przepływu cieczy, wpływa na zwiększenie ciśnienia cieczy przed zaworem i redukcję przepływu cieczy za zaworem. W ten sposób można symulować występowanie takich uszkodzeń jak wyciek czy przytkanie. Jednocześnie zastosowanie manualnych zaworów zapewniających płynną regulację ich otwarcia pozwala na symulację



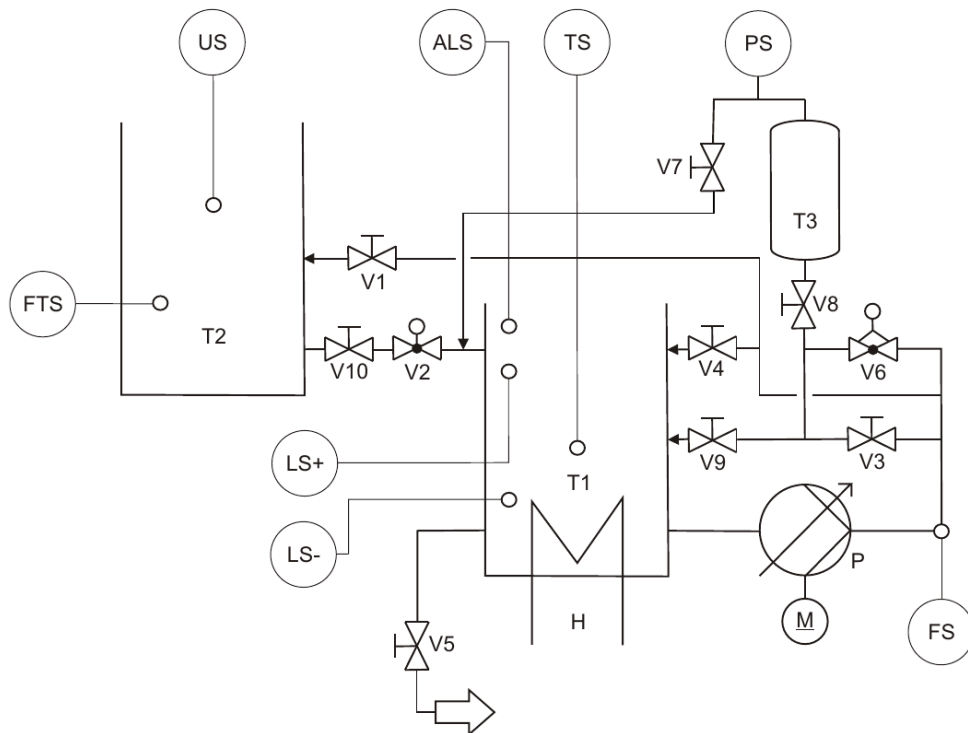
Rys. 5.1: Stanowisko laboratoryjne (Skupnik, 2009)

uszkodzeń o różnym stopniu intensywności. Takie podejście pozwala na wygenerowanie zbioru danych o dużym stopniu zróżnicowania.

Dzięki zastosowaniu technologii OPC dostępnej w zastosowanym sterowniku PLC firmy Siemens możliwa była akwizycja w czasie rzeczywistym wartości zmiennych procesowych ze stanowiska laboratoryjnego. Dane pomiarowe zapisywane były za pomocą oprogramowania MATLAB na dysku twardym komputera, a następnie eksportowane do formatu CSV w celu dalszego przetwarzania, analizy i wykorzystania. Stanowisko laboratoryjne jest wyposażone w cztery czujniki pomiarowe, co odpowiada czterem niezależnym sygnałom procesowym:

- US - ultradźwiękowy czujnik poziomu cieczy w zbiorniku T2,
- PS - czujnik ciśnienia powietrza w zbiorniku T3,
- FS - czujnik przepływu cieczy znajdujący się bezpośrednio za pompą P,
- TS - czujnik temperatury cieczy w zbiorniku T1.

W zbiorniku T1 został umieszczony czujnik temperatury cieczy ponieważ w tym samym zbiorniku znajduje się również grzałka, która pozwala na podniesienie temperatury cieczy w tym zbiorniku. Ze względu na charakter eksperymentu, konieczność wykonania wielu jego powtórzeń w celu zebrania jak największej ilości danych oraz dużą bezwładność procesu podgrzewania i chłodzenia cieczy, zrezygnowano z wykorzystania tego elementu w realizowanym eksperymencie. W oprogramowaniu sterującym pozwalającym na kontrolowanie aktywnych elementów stanowiska zaimplementowano również regulator bazującego na regulatorze PID, który pozwala na sterowanie wybranym akuatorem (w przypadku badań opisanych w pracy, elementem tym jest pompa) w celu osiągnięcia zadanej wartości wybranej zmiennej procesowej, np. istnieje możliwość ustalenia doce-



Rys. 5.2: Schemat stanowiska laboratoryjnego

lowego poziomu cieczy w zbiorniku T2 i dzięki zastosowaniu kontrolera PID pompa zostanie wyłączona po uzyskaniu zadanej wartości pomiarowej. Dzięki zastosowaniu tak skonstruowanego stanowiska laboratoryjnego odzwierciedlającego rzeczywiste instalacje, które wyposażono w czujniki pomiarowe możliwe jest wygenerowanie danych obarczone naturalnym szumem pomiarowym, który zawsze występuje w tego typu rzeczywistych układach pomiarowych.

## 5.2. Opis eksperymentu

Eksperyment został podzielony na 9 etapów, które zostały nazwane stanami funkcjonalnymi (*sf*). W każdym ze stanów funkcjonalnych realizowana jest pojedyncza operacja trwająca określony okres czasu. Poszczególne stany zostały wymienione i opisane w kolejnych punktach:

- $sf_1$  (stan początkowy) - w tym stanie, stanowisko laboratoryjne znajduje się w następującej konfiguracji:
  - czas trwania stanu: do 60s,
  - w zbiorniku T1 znajduje się cała woda z układu hydraulicznego stanowiska,
  - zbiorniki T2 i T3 są puste,
  - wszystkie zawory są zamknięte,
  - pompa jest wyłączona.

- $sf_2$  - ciecz jest pompowana ze zbiornika T1 do zbiornika T3, w którym następuje kompresja powietrza (Rys. 5.3(a)):
  - czas trwania stanu: 30s,
  - zawory V3 i V8 są otwarte,
  - pozostałe zawory są zamknięte,
  - pompa jest włączona.
- $sf_3$  - ciecz jest w obiegu zamkniętym, jest pompowana ze zbiornika T1 do zbiornika T1 (Rys. 5.3(b)):
  - czas trwania stanu: 20s,
  - zawór V4 jest otwarty,
  - pozostałe zawory są zamknięte,
  - pompa jest włączona.
- $sf_4$  - ciecz jest pompowana ze zbiornika T1 do zbiornika T2 (Rys. 5.3(c)):
  - czas trwania stanu: 60s,
  - zawór V1 jest otwarty,
  - pozostałe zawory są zamknięte,
  - pompa jest włączona.
- $sf_5$  - ciecz bez udziału pompy przepływa ze zbiornika T3 do zbiornika T1 (Rys. 5.3(d)):
  - czas trwania stanu: 15s,
  - zawory V8 i V9 są otwarte,
  - pozostałe zawory są zamknięte,
  - pompa jest wyłączona.
- $sf_6$  - ciecz bez udziału pompy przepływa ze zbiornika T2 do zbiornika T1 (Rys. 5.3(e)):
  - czas trwania stanu: 120s,
  - zawory V1 i V4 są otwarte,
  - pozostałe zawory są zamknięte,
  - pompa jest wyłączona.
- $sf_7$  - ciecz jest pompowana ze zbiornika T1 do zbiornika T3 (Rys. 5.3(f)):
  - czas trwania stanu: 30s,
  - zawory V3 i V8 są otwarte,
  - pozostałe zawory są zamknięte,



- pompa jest włączona.
- $sf_8$  - ciecz jest pompowana ze zbiornika T1 do zbiornika T2, jednocześnie ciecz znajdująca się w zbiorniku T3 jest wypuszczana (Rys. 5.3(g)):
  - czas trwania stanu: 60s,
  - zawory V1, V3 i V8 są otwarte,
  - pozostałe zawory są zamknięte,
  - pompa jest włączona.
- $sf_9$  - ciecz bez udziału pompy przepływa ze zbiornika T2 do zbiornika T1 (Rys. 5.3(h)):
  - czas trwania stanu: 120s,
  - zawory V1 i V4 są otwarte,
  - pozostałe zawory są zamknięte,
  - pompa jest włączona.

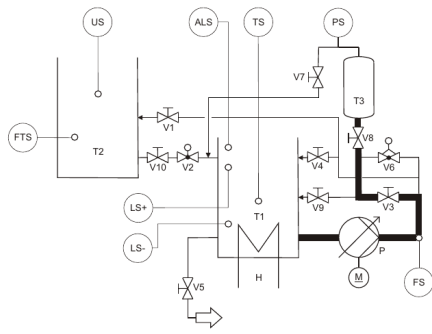
Stan funkcjonalny  $sf_1$  nie ma określonego czasu realizacji, ponieważ w trakcie trwania tego stanu funkcjonalnego operator weryfikuje, czy obiekt jest poprawnie przygotowany do pracy. Oprócz wymienionych wyżej dziewięciu stanów funkcjonalnych, występuje jeszcze jeden stan oznaczany indeksem  $sf_0$ . Stan funkcjonalny  $sf_0$  jest stanem nieustalonym, do którego przypisywane są dane występujące pomiędzy stanami funkcjonalnymi  $sf_1$  -  $sf_9$ . Wynika to z konieczności przygotowania stanowiska do realizacji postępujących po sobie stanów funkcjonalnych. W dalszej części pracy stan funkcjonalny  $sf_0$  również jest brany pod uwagę i badany jest jego wpływ na działanie opracowanej metody detekcji uszkodzeń.

Opisane wcześniej stany funkcjonalne są w dalszej części pracy rozpatrywane jako wartości cechy kontekstowej mające wpływ na działanie procesu detekcji uszkodzeń. Autor pracy zastosował stanowisko laboratoryjne do wygenerowania zbiorów danych dla stanu pełnej zdatności oraz dla stanów z uszkodzeniem. W wyniku analizy dostępnych zmiennych procesowych oraz konstrukcji stanowiska, zaproponowano zasymulowanie takich rodzajów uszkodzeń, które najczęściej występują w tego typu systemach:

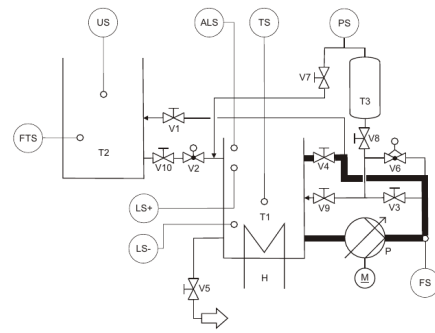
- zatkanie (całkowite lub częściowe) połączenia pomiędzy zbiornikami,
- wyciek cieczy ze zbiornika,
- uszkodzenia czujnika lub toru pomiarowego.

Na tej podstawie zaproponowano szereg uszkodzeń, które zostały zasymulowane w trakcie realizacji eksperymentów na stanowisku laboratoryjnym:

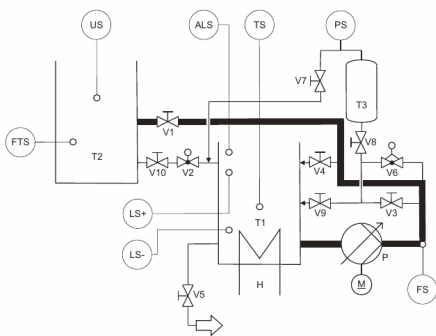
- $f1$  - ograniczony przepływ pomiędzy zbiornikiem T1 i zbiornikiem T3 (częściowo zamknięty zawór V8),
- $f2$  - nieszczelny zbiornik T3 (częściowo otwarty zawór V7),



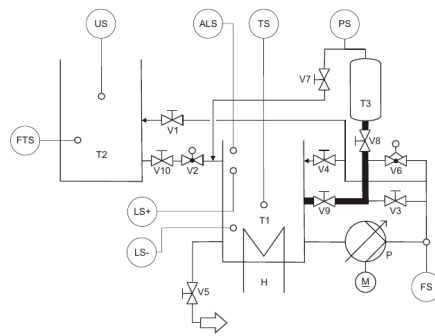
(a) Stan funkcjonalny 2



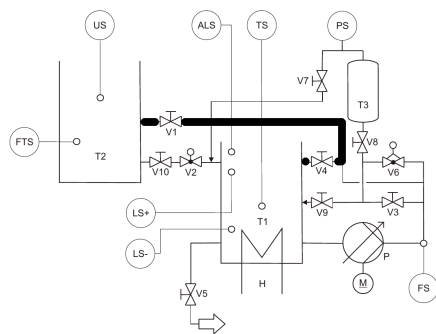
(b) Stan funkcjonalny 3



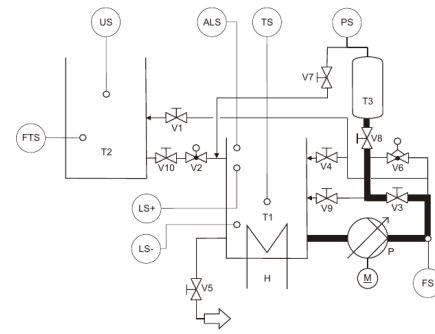
(c) Stan funkcjonalny 4



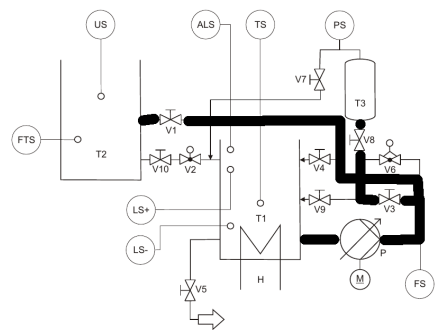
(d) Stan funkcjonalny 5



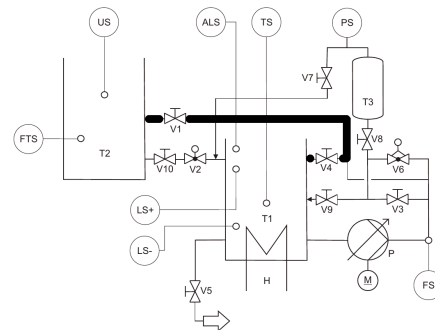
(e) Stan funkcjonalny 6



(f) Stan funkcjonalny 7



(g) Stan funkcjonalny 8



(h) Stan funkcjonalny 9

Rys. 5.3: Przepływ cieczy w stanach funkcjonalnych

- $f_3$  - ograniczony przepływ pomiędzy zbiornikiem T1 i zbiornikiem T2 (częściowo zamknięty zawór V1),
- $f_4$  - nieszczelne połączenie rurociągu pomiędzy pompą P, a zbiornikiem T1 (częściowo otwarty zawór V1),
- $f_5$  - nieszczelne połączenie instalacji pomiędzy zbiornikiem T1 i zbiornikiem T2 (częściowo otwarty zawór V4),
- $f_6$  - nieszczelny zbiornik T2 (częściowe otwarcie zaworu V1 i V4),
- $f_7$  - ograniczony przepływ pomiędzy pompą P i zbiornikiem T1 (częściowo zamknięty zawór V4),
- $f_8$  - niesprawny ultradźwiękowy czujnik US poziomemu cieczy w zbiorniku T2 (fizyczne zasłonięcie części czołowej czujnika),
- $f_9$  - niesprawny ultradźwiękowy czujnik US poziomemu cieczy w zbiorniku T2 (programowa modyfikacja wartości),
- $f_{10}$  - niesprawny czujnik ciśnienia powietrza PS w zbiorniku T3 (programowa modyfikacja wartości),
- $f_{11}$  - niesprawny czujnik przepływu FS (programowa modyfikacja wartości).

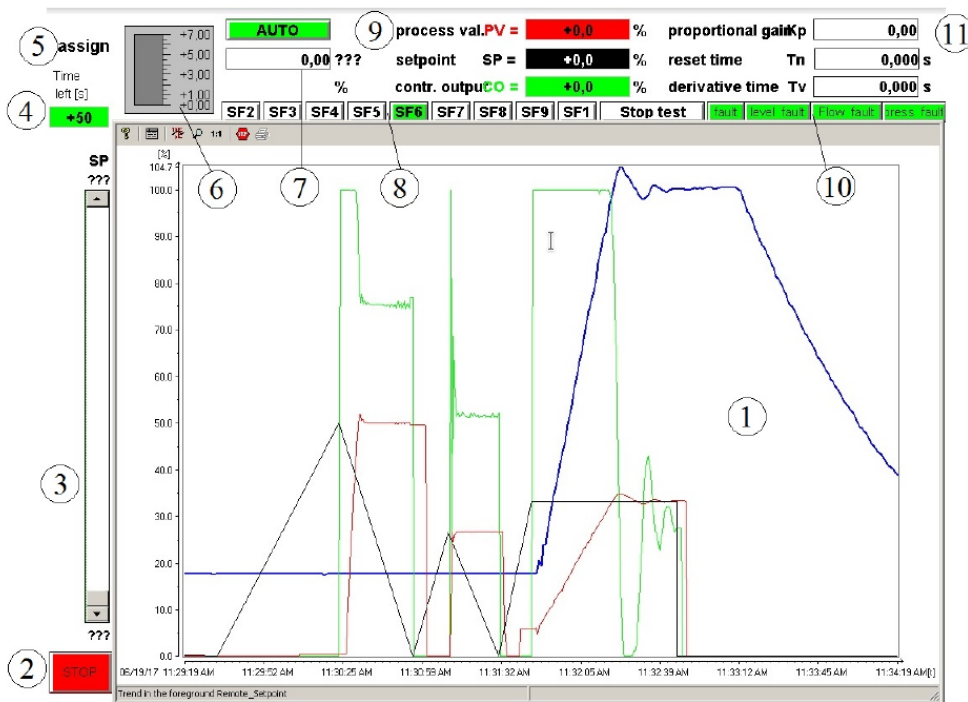
Po zapoznaniu się ze schematem i konstrukcją stanowiska oraz po określeniu listy zmiennych procesowych monitorowanych podczas realizacji eksperymentu, przystąpiono do przygotowania toru pomiarowego oraz oprogramowania rejestrującego wartości wybranych sygnałów.

### 5.3. Akwizycja i przygotowanie danych

Stanowisko laboratoryjne zostało wyposażone w sterownik PLC firmy Siemens, który pozwala na implementację oprogramowania typu SCADA. Na potrzeby niniejszej pracy utworzono aplikację, która w znacznym stopniu wspomaga proces realizacji eksperymentu poprzez np. kontrolę czasu trwania poszczególnych stanów funkcjonalnych. Aplikacja oraz proces generowania danych był przedmiotem pracy (Mync, 2017).

Na rysunku 5.4 przedstawiono ekran kontroli procesu, na którym występują następujące elementy:

1. Wykres prezentujący wartości zmiennych procesowych wyrażone w procentach:
  - kolor zielony - sygnał sterujący pompy,
  - kolor czerwony - aktualna wartość zmiennej procesowej,
  - kolor czarny - sygnał wartości zadanej.
2. Przycisk zatrzymujący realizację eksperymentu.
3. Suwak do zmiany wartości zadanej.
4. Pozostały czas do zakończenia aktualnie trwającego stanu funkcjonalnego.



Rys. 5.4: Widok ekranu kontroli (Mync, 2017)

5. Informacja o aktualnie sterowanej zmiennej procesowej.
6. Wskaźnik pokazujący aktualny przepływ cieczy w l/min.
7. Pole pozwalające na określenie wartości zadanej.
8. Przyciski pozwalające na rozpoczęcie określonego stanu funkcjonalnego.
9. Aktualne wartości parametrów, które również są prezentowane na wykresie (1).
10. Przyciski modyfikujące wartości odczytane z wybranych czujników zainstalowanych na stanowisku w celu umożliwienia wprowadzenia dodatkowych uszkodzeń (błędów pomiarowych).
11. Aktualne nastawy regulatora PID.

Do sterownika podłączono wcześniej wymienione czujniki, dzięki czemu za pomocą dostępnego serwera OPC możliwa jest akwizycja danych pomiarowych jak i pozostałych parametrów w czasie rzeczywistym w trakcie wykonywania eksperymentu. Na komputerze docelowym zainstalowano aplikację MATLAB wraz z narzędziem zawierającym klienta OPC. Taka konfiguracja stanowiska pozwoliła na rejestrowanie wszystkich dostępnych zmiennych procesowych z częstotliwością 10Hz i późniejsze zapisanie ich do plików csv. Rejestrowane były następujące parametry, sygnały i zmienne procesowe:

- data i godzina zarejestrowanej próbki danych,
- F - sygnał analogowy zmian natężenia przepływu cieczy (uzyskiwany z czujnika natężenia przepływu cieczy FS),
- L - sygnał analogowy zmian poziomu cieczy w zbiorniku T2 (uzyskiwany z ultradźwiękowego czujnika poziomu US),

- P - sygnał analogowy zmian ciśnienia powietrza w zbiorniku T3 (uzyskiwany z czujnika ciśnienia PS),
- sygnał analogowy zmian wielkości sterowanej (prędkość obrotowa wirnika pompy P),
- wartości składowych parametrów kontrolera PID,
- aktualna wartość parametru zadanego (natężenie przepływu cieczy, poziom cieczy w zbiorniku T2 lub poziom ciśnienia powietrza w zbiorniku T3) wyrażona w procentach,
- czas pozostały do zakończenia aktualnego stanu funkcjonalnego,
- stan pracy pompy (włączona / wyłączona).

Kolejny zestaw sygnałów wynikał z charakteru eksperymentu i były generowane bezpośrednio przez aplikację sterującą stanowiskiem:

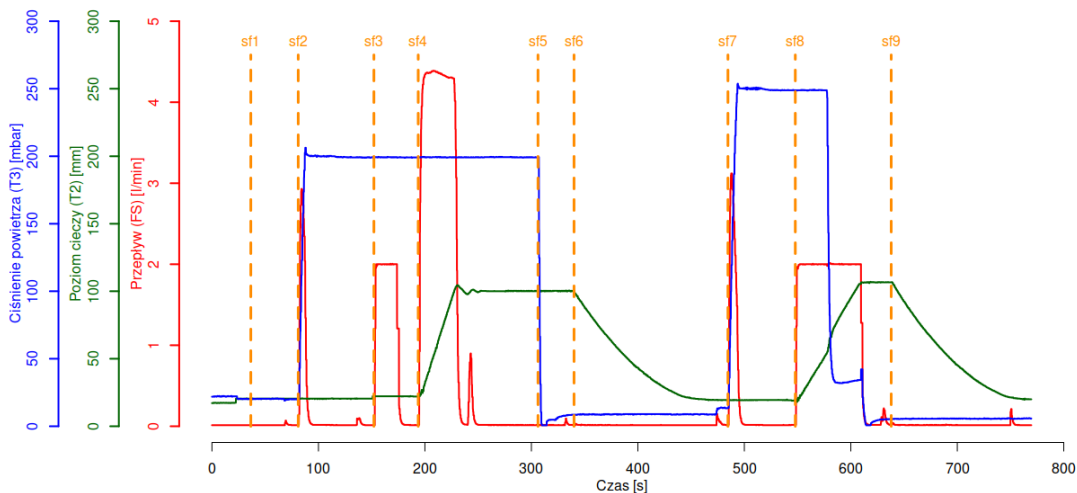
- zestaw binarnych flag (od 1 do 9) określających stan funkcjonalny przypisany do pojedynczego wiersza danych,
- sf - parametr określający numeryczną reprezentację stanu funkcjonalnego.

Ze względu na charakter dalszych badań konieczna była wielokrotna realizacja zaplanowanego eksperymentu w celu pokrycia całej koniecznej przestrzeni wariantów do celów detekcji uszkodzeń. Oznacza to równomierny rozkład danych pomiędzy poszczególnymi stanami w taki sposób, aby ilość danych związana z wszystkimi uszkodzeniami dotyczyła połowy wszystkich dostępnych danych, a stan bez uszkodzeń zawierał się w drugiej połowie danych. W tym celu założono, że dla każdego stanu z uszkodzeniem zrealizowanych zostanie 10 eksperymentów. Ponieważ zaplanowano, że na stanowisku zostanie zasymulowanych 11 uszkodzeń, całkowita liczba eksperymentów dla wszystkich stanów z uszkodzeniami wynosi 110. Aby zrealizować wcześniej opisane założenie konieczne było wykonanie dodatkowych 110 eksperymentów dla stanu pełnej zdadności stanowiska. Całkowita liczba eksperymentów wynosi 220. Czas realizacji pojedynczego eksperymentu trwał od 10 do 20 minut, ze względu na konieczność przeprowadzenia po każdym eksperymencie szeregu operacji związanych z poprawnym zapisaniem zebranych danych czy doprowadzeniem stanowiska do stanu, który pozwala na przeprowadzenie kolejnego eksperymentu. Ostatecznie czas poświęcony na wykonanie wszystkich eksperymentów, analizę zebranych danych, ponowne wykonanie eksperymentów dla błędnych przypadków zajęł około 2 miesięcy.

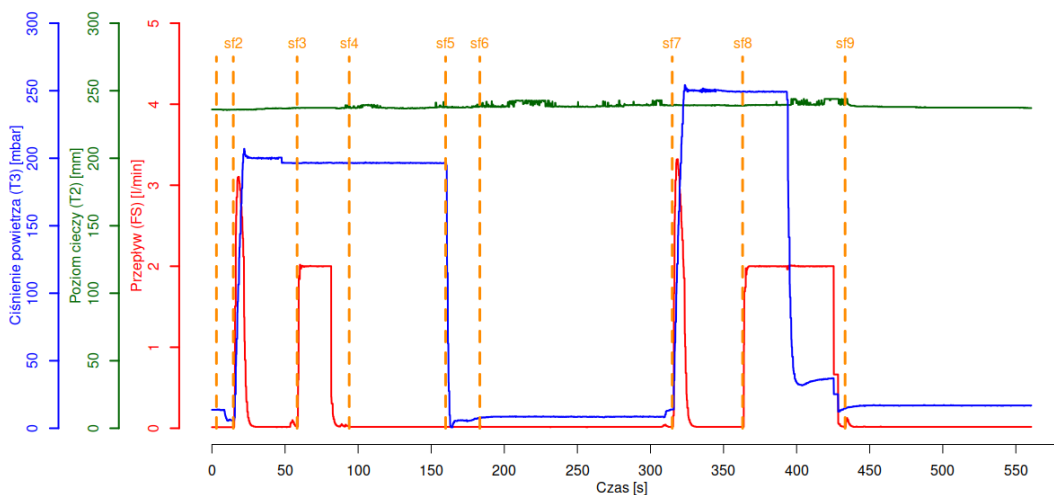
Po zrealizowaniu wszystkich eksperymentów wygenerowanych zostało 220 plików CSV zawierających kompletny zbiór wartości wszystkich zmiennych procesowych zarejestrowanych w czasie  $t$ . Na rysunku 5.5 przedstawiono wykresy reprezentujące zmiany wartości pomiarowych w trakcie wykonywania pojedynczego eksperymentu, w którym nie wystąpiło uszkodzenie (a) oraz przykład, w którym wywołano uszkodzenie (b).

Na wykresach 5.5 oprócz przebiegu wartości zmiennych procesowych oznaczono również punkty w czasie, w których następowało rozpoczęcie danego stanu funkcjonalnego,

gdzie tak jak to zostało opisane w rozdziale 5.2, stan funkcjonalny jest związany z określonym zadaniem realizowanym na stanowisku. Zostały one oznaczone pomarańczowymi przerywanymi liniami wraz z etykietami umieszczonymi w górnej części wykresu. Etykiety zostały zapisane w formie skróconej, gdzie etykieta  $sf$  oznacza stan funkcjonalny, a wartość liczbowa określa numer rozpoczynającego się stanu funkcjonalnego. Dokładne zdefiniowanie obszarów działania stanów funkcjonalnych jest istotne z punktu widzenia kolejnego kroku przetwarzania zebranych danych. Dla poprawy czytelności wykresu pominięto wizualizację stanu funkcjonalnego  $sf_0$ , który jest stanem nieustalonym i występuje w chwilach przejścia pomiędzy stanami funkcjonalnymi.



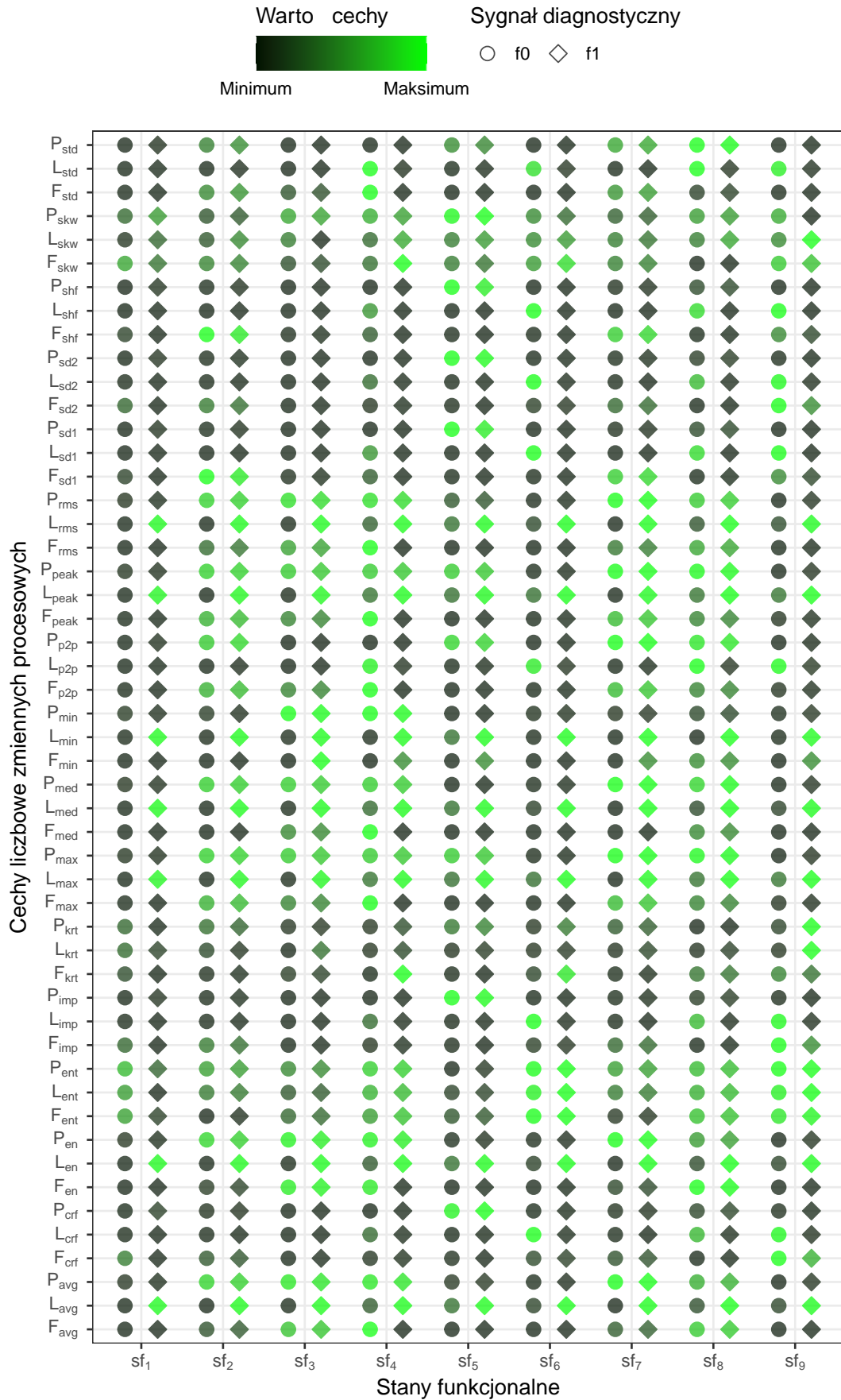
(a) Stan bez uszkodzenia



(b) Uszkodzenie czujnika wysokości poziomu cieczy w zbiorniku 2 (T2)

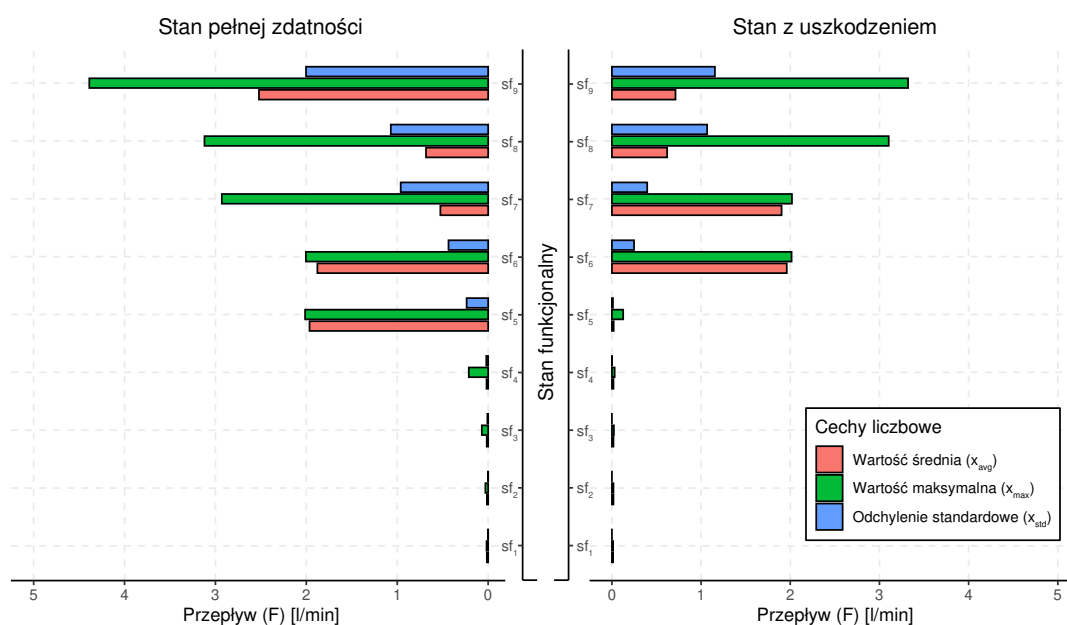
Rys. 5.5: Porównanie przebiegu wartości zmiennych procesowych dla stanu bez uszkodzenia i z uszkodzeniem

Tak jak to zostało opisane wyżej, stan funkcjonalny dotyczy pojedynczego i elementarnego zadania realizowanego na stanowisku laboratoryjnym. Przykładem takiego zadania jest stan funkcjonalny  $sf_4$ , w którym następuje uruchomienie pompy i przetaczanie cieczy ze zbiornika T1 do zbiornika T2 w celu uzyskania określonej wysokości słupa cieczy w zbiorniku T2. W przypadku poprawnie działającego stanowiska (w pełni



Rys. 5.6: Porównanie wszystkich wartości cech liczbowych dla stanu pełnej zdatności i stanu z uszkodzeniem

drożne rury, brak wycieków, sprawna pompa, prawidłowo działający czujnik wysokości słupa cieczy w zbiorniku T2, poprawnie działający sterownik wraz z oprogramowaniem, itp.) każdorazowe powtórzenie eksperymentu powinno dać niemal identyczne przebiegi wartości pomiarowych dla każdego ze stanów funkcjonalnych. Z tego powodu zamiast analizować nieprzetworzone dane zebrane bezpośrednio ze stanowiska, postanowiono opisać je za pomocą szeregu cech liczbowych, które określają ich charakter w ramach pojedynczego stanu funkcjonalnego (wartość średnia, wartość maksymalna, tempo narastania/opadania, współczynnik kształtu przebiegu czasowego czy energia). W przypadku zastosowanego opisywanego eksperymentu zdecydowano się na wykorzystanie 17 cech, które zostały wymienione w rozdziale 4.1.



Rys. 5.7: Porównanie wybranych wartości cech liczbowych wyznaczonych dla przepływu (F) dla stanu pełnej zdatności i stanu z uszkodzeniem

Kolejnym krokiem jest przekształcenie zbioru danych, zawierających przebiegi czasowe z trzech zmiennych procesowych, do zbioru zawierającego wyliczone wartości cech liczbowych w oparciu o stany funkcjonalne. W wyniku tego przekształcenia, które zostało przedstawione w rozdziale 4.1, powstaje tabela, której liczba atrybutów (kolumn) zwiększa się z 3 do 51. Na rysunku 5.6 symbolicznie (za pomocą odcieni koloru zielonego) porównano ze sobą wartości cech liczbowych dla wszystkich cech i zmiennych procesowych, bazując na wybranych pojedynczych eksperymentach zrealizowanych dla stanu bez uszkodzenia i z uszkodzeniem. Wyniki obliczone osobno dla stanu pełnej zdatności (punkty w kształcie koła) i z uszkodzeniem (punkty w kształcie rombu) zostały pogrupowane względem stanów funkcjonalnych wymienionych na osi poziomej wykresu. Wartości przed przedstawieniem ich na wykresie zostały znormalizowane do zakresu od 0 do 1 w oparciu o cały zbiór danych (wszystkie stany techniczne i funkcjonalne) dla określonej cechy i zmiennej procesowej. W każdej parze wyników wartość została przedstawiona symbolicznie jako odcień koloru zielonego, gdzie kolor czarny oznacza wartość najmniejszą, a kolor jaskrawo zielony wartość największą. Natomiast na rysunku



5.7 przedstawiono wartości wybranych cech (wartość średnia, maksymalna, odchylenie standardowe obliczonych na podstawie zmiennej procesowej przepływu) w formie umożliwiającej dokładniejsze porównanie dla stanu pełnej zdatności (lewa część wykresu) i z uszkodzeniem (prawa część wykresu). Wartości zostały dodatkowo pogrupowane względem stanów funkcjonalnych realizowanych w procesie. Tak przygotowane dane zostały wykorzystane w procesie trenowania, testowania i weryfikacji zaproponowanej metody detekcji uszkodzeń bazującej na kontekście.

Kolejnym krokiem realizacji badań było przygotowanie narzędzi oraz środowiska w celu realizacji procesu diagnostycznego. Zbiór danych otrzymany w wyniku akwizycji wartości zmiennych procesowych w trakcie trwania eksperymentu na stanowisko został wykorzystany do zasilenia przygotowanych procesów wnioskowania zarówno jako dane trenujące, testowe jak i walidacyjne.

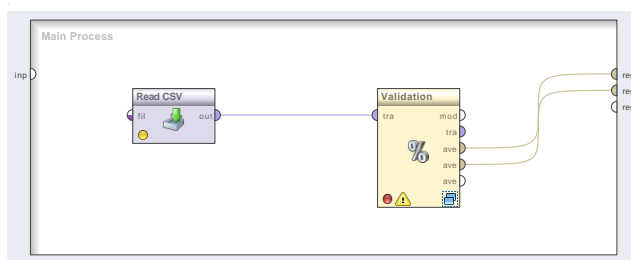
## 5.4. Realizacja procesu diagnostycznego

W celu umożliwienia realizacji eksperymentu dotyczącego trenowania zespołu klasyfikatorów z wykorzystaniem algorytmu optymalizacji ich wybranych parametrów oraz z wykorzystaniem kontekstu, opracowano w języku JAVA odpowiednie narzędzie. Opracowanie własnego zestawu algorytmów było konieczne, ponieważ metoda opisana w niniejszej pracy doktorskiej zawiera szereg niestandardowych elementów i podejść, które nie występują w powszechnie dostępnych narzędziach realizujących zadania uczenia maszynowego. Wykorzystano aplikację RapidMiner w wersji 5, która jest dostępna na zasadach licencji OpenSource. Dodatkowo sposób implementacji tego narzędzia pozwala na wygodne rozszerzanie jego funkcjonalności poprzez opracowywanie i dodawanie wtyczek (*ang. plugins*). Zastosowano wersję aplikacji wraz z wtyczkami, które zostały opracowane przez autora niniejszej pracy w Katedrze Podstaw Konstrukcji Maszyn w ramach projektu DISESOR (Przystałka i Sikora, 2017). Opracowane wtyczki dodają do aplikacji Rapidminer możliwość implementacji procesów wnioskowania w oparciu o sieci Bayesa, czy logikę rozmytą, zarówno w oparciu o dane trenujące, jak i wiedzę użytkownika. Zaimplementowane usprawnienie pozwoliło na dodanie do interfejsu aplikacji narzędzia edycyjnego, pozwalającego inżynierom wiedzy na ręczne utworzenie modelu wnioskowania w edytorze bez udziału danych lub edycję klasyfikatorów wcześniej utworzonych w wyniku działania algorytmów trenujących. Innym elementem utworzonym w ramach projektu DISESOR przez autora niniejszej pracy był meta-klasyfikator kontekstowy, który został wykorzystany w celu realizacji eksperymentu opracowanego w pracy doktorskiej.

W celu realizacji elementu eksperymentu związanego z automatycznym doбором parametrów trenujących klasyfikatorów w komitecie, postanowiono wykorzystać gotowy zestaw bibliotek, umożliwiający implementację zadania optymalizacji parametrów z wykorzystaniem wybranego algorytmu genetycznego MOEA Framework (*MultiObjective Evolutionary Algorithms*) (MOEA Framework, 2023).

### 5.4.1. Trenowanie modeli detekcji uszkodzeń

Realizacja zadania wytrenowania klasyfikatorów wymagała przygotowania odpowiedniego procesu w oprogramowaniu RapidMiner 5. Za pomocą dostępnych narzędzi opracowano graf łączący poszczególne zadania. Zadania pokazane na schematach w postaci bloków są nazywane w aplikacji operatorami i wykonywane są sekwencyjnie, zaczynając od wczytania wcześniej przygotowanych danych. Dane wczytywane są za pomocą operatora „Read CSV”, a następnie przekazywane do operatora realizującego proces walidacji krzyżowej „Validation”, co zostało pokazane na rysunku 5.8.



Rys. 5.8: Widok głównego procesu realizującego zadanie trenowania klasyfikatorów

Głównym zadaniem operatora „Validation” jest podział podanych na wejściu danych na mniejsze grupy. Ze względu na liczebność próbek zdecydowano się na podział całego zbioru danych na 5 podgrup, co przekłada się na 5-krotną realizację procesu walidacji. W każdej iteracji 4 wybrane podgrupy biorą udział w trenowaniu klasyfikatorów, a ostatnia podgrupa jest wykorzystywana do wyznaczenia miar skuteczności klasyfikacji utworzonych modeli. W ramach operatora „Validation” realizowane są zadania trenowania i testowania klasyfikatorów. Operator znajdujący się w obszarze o nazwie „Training” zawiera inny operator, który pozwala na realizację zadania trenowania komitetu klasyfikatorów z uwzględnieniem kontekstu. Na wejściu operatora „Ensemble classifier based on context” podawane są 4 podgrupy danych należące do zbioru trenującego. Na wyjściu generowany jest model wytrenowanych klasyfikatorów. Model przekazywany jest do kolejnego fragmentu grafu nazwanego „Testing”, w którym poddawany jest walidacji za pomocą ostatniej podgrupy danych przeznaczonych do testowania. Operatory „Performance” pozwalają na wyznaczenie szeregu miar, które następnie są przekazywane jako rezultat działania tego fragmentu procesu. Liczba realizacji tego zadania jest zgodna z liczbą iteracji walidacji krzyżowej. Wynikiem działania całego procesu jest lista miar, pozwalająca określić skuteczność działania klasyfikatora, a w szczególności  $TPR$  i  $TNR$ , które są elementami funkcji celu dla algorytmu optymalizacji parametrów procesu.

Kolejnym etapem badań jest wielokrotne uruchomienie opisanego wyżej procesu w ramach działania genetycznego algorytmu optymalizacji NSGA-II. Dla każdego osobnika w pokoleniu, proces utworzony w oprogramowaniu Rapidminer jest uruchamiany dziesięciokrotnie aby dla każdej z rozpatrywanych miar ( $TPR$  i  $TNR$ ), określić wartość średnią oraz odchylenie standardowe uzyskiwanych wyników. Następnie uzyskane statystyki są wstawiane do funkcji celu i oceniane przez algorytm optymalizacji. Sam proces optymalizacji za pomocą algorytmu genetycznego NSGA-II został uruchomiony z następującymi

parametrami:

- Liczba pokoleń: 20,
- Liczba osobników w pokoleniu: 100,
- Operator krzyżowania: Symulowane krzyżowanie binarne (*ang. Simulated Binary Crossover*),
- Prawdopodobieństwo krzyżowania: 0.9,
- Operator mutacji: Mutacja wielomianowa (*ang. Polynomial Mutation*),
- Prawdopodobieństwo mutacji: 0.1.

#### 5.4.2. Zastosowanie modeli detekcji uszkodzeń

Wynikiem działania metody opracowanej w ramach realizacji badań zaprezentowanych w pracy doktorskiej, jest zbiór rozwiązań optymalnych należących do frontu pareto. Każde z tych rozwiązań jest optymalne ze względu na określone kryteria, jednak do eksperta należy decyzja, dotycząca wyboru ostatecznego rozwiązania. Dla wybranego rozwiązania dostępna jest lista optymalizowanych parametrów wraz z wartościami, które należy odczytać i utworzyć klasyfikator w oparciu o dane trenujące i docelowe parametry. W kolejnym rozdziale pokazano i wyjaśniono proces wizualizacji i wyboru docelowego rozwiązania.

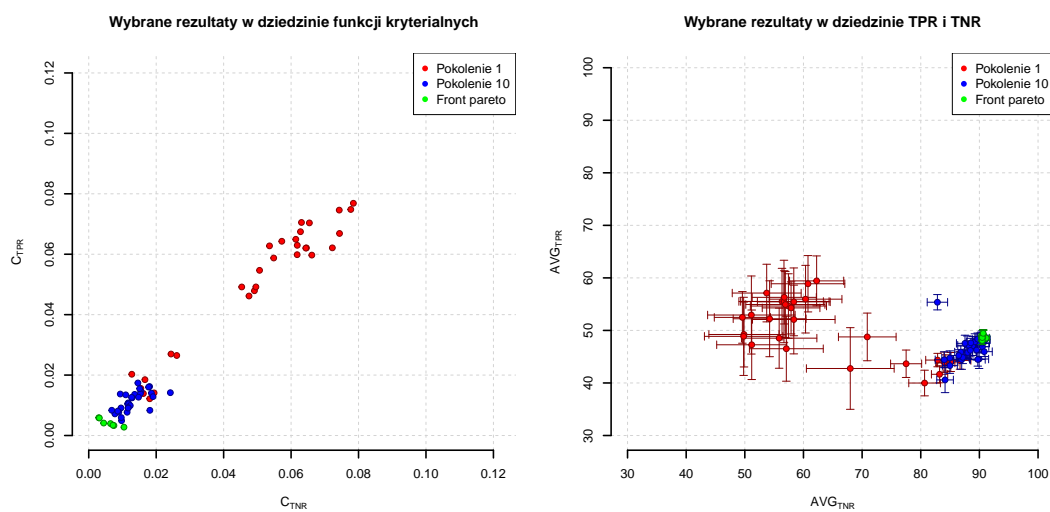
### 5.5. Prezentacja i dyskusja wyników zastosowania opracowanej metody detekcji uszkodzeń

W rozdziale opisano wyniki działania metody kontekstowej detekcji uszkodzeń na przykładzie danych zebranych ze stanowiska laboratoryjnego FESTO. W pierwszej części rozdziału opisano wyniki działania procesu optymalizacji parametrów wraz z wizualizacją. Proces optymalizacji, tak jak zostało to opisane w rozdziale 4.4, ma na celu znalezienie zbioru rozwiązań pareto optymalnych, zgodnych z założonymi kryteriami. Proces wizualizacji i zrozumienia uzyskanych wyników jest istotny z punktu widzenia wybrania jednego rozwiązania i zaimplementowania w procesie detekcji uszkodzeń. W drugiej części rozdziału opisano wyniki działania rozwiązań optymalnych poprzez wytrenowanie klasyfikatora za pomocą danych treningowych w oparciu o parametry przypisane do wybranego rozwiązania.

#### 5.5.1. Analiza wyników procesu optymalizacji

Na kolejnych wykresach pokazano zbiór przykładowych wyników prezentujących wpływ zmian parametrów  $w$  i  $t_{\alpha}$ , w funkcjach celu, na zbiór rozwiązań optymalnych. W pierwszej kolejności zestawiono ze sobą rezultaty uzyskane w oparciu o algorytm drzewa decyzyjnego, dla którego w procesie optymalizacji zastosowano funkcje kryterialne 4.9 i 4.10

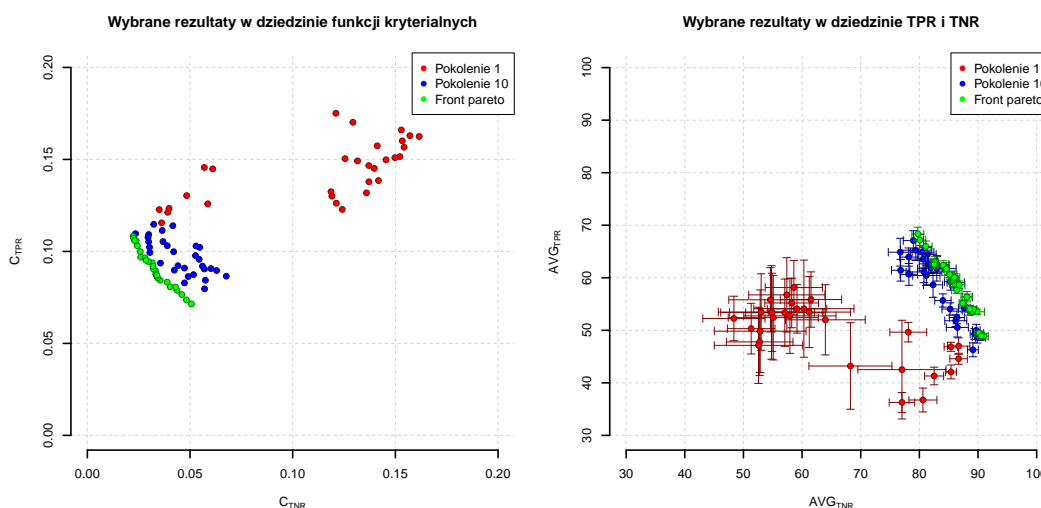
opisane w rozdziale 4.4.1. Na kolejnych wykresach przedstawiono wpływ wartości parametru  $w$  na kształt oraz rozmieszczenie rozwiązań optymalnych z podziałem na wybrane pokolenia oraz ich interpretację w przestrzeni składowych, jako wartość średnia  $TPR$  i  $TNR$  oraz odchylenie standardowe. Postanowiono wybrać pięć wartości parametru  $w$ . Dwie skrajne wartości czyli 0 i 1, wartość środkową 0.5 oraz dwie dodatkowe wartości pośrednie, 0.2 i 0.8. Na wszystkich wykresach przedstawiono wyniki uzyskane dla 2 pokoleń, pokolenia 1 (kolor czerwony) oraz pokolenia 10 (kolor niebieski), oraz pokazano zbiór rozwiązań pareto optymalnych (kolor zielony). Ze względu na dużą liczbę osobników dla każdego z pokoleń, postanowiono ograniczyć liczbę wyświetlanych punktów na wykresach do około 1/3 losowo wybranych osobników dla każdego z pokoleń. Uproszczenie to pozwala na czytelniejsze przedstawienie rozkładu rozwiązań. Pierwszy wykres (rys. 5.9) przedstawia zbiór rozwiązań dla parametru  $w = 0$ .



Rys. 5.9: Wybrane zbiory rozwiązań dla parametru  $w = 0$

Na rysunku 5.9 pokazano dwie reprezentacje w postaci dwóch wykresów punktowych prezentujących wyniki uzyskane w trakcie realizacji procesu optymalizacji. Z lewej strony przedstawiono zbiór rozwiązań w dziedzinie funkcji kryterialnych. Rozwiązania optymalne, oznaczone punktami w kolorze zielonym skupiają się w okolicy punktu 0,0. Wynika to z faktu, że dla parametru  $w = 0$  część wartości funkcji kryterialnej dotycząca wartości średniej jest mnożona razy 0, natomiast część dotycząca wartości odchylenia standardowego jest mnożona razy 1, co w rezultacie przekłada się na wynik funkcji kryterialnej zależny jedynie od wartości odchylenia standardowego. Takie podejście wymusza na algorytmie optymalizacji odpowiednie dobranie parametrów klasyfikatorów, aby minimalizowały odchylenie standardowe uzyskanych wyników. Natomiast wykres znajdujący się z prawej strony rys. 5.9 przedstawia inną reprezentację wyników uzyskanych dla tego samego przypadku. Najważniejszą różnicą w stosunku do poprzedniego wykresu jest fakt, że wartości reprezentowane za pomocą punktów dotyczą jedynie jednego składnika funkcji kryterialnej tj. wartości średniej detekcji stanu pełnej zdatności ( $AVG_{TNR}$ ) i stanu z uszkodzeniem ( $AVG_{TPR}$ ) uzyskanej przez klasyfikatory w trakcie ich trenowania. Sama estymacja wartości przedstawionych na wykresie jest wynikiem zastosowania na tym etapie metody walidacji krzyżowej. Do każdego punktu zostały również dodane

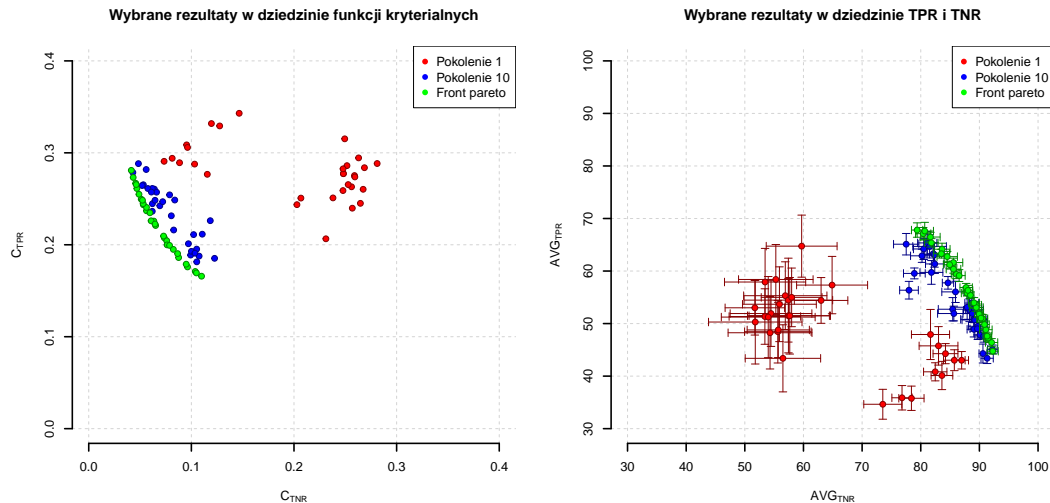
granice w postaci symetrycznych przedziałów reprezentujących wartość odchylenia standardowego dla danego rozwiązania na wykresie. Wartość odchylenia standardowego jest obliczana niezależnie dla stanu pełnej zdatości i stanu z uszkodzeniem, co przekłada się na granice o różnej długości w osi poziomej i pionowej. Na jednym wykresie pokazano zbiór rozwiązań uzyskanych dla trzech różnych pokoleń utworzonych w trakcie realizacji procesu optymalizacji parametrów klasyfikatorów. Czerwonym kolorem oznaczono rozwiązania będące zbiorem początkowym (pokolenie 1), które jest wynikiem losowania wartości parametrów klasyfikatorów. Kolorem niebieskim pokazano zbiór rozwiązań uzyskanych w pokoleniu 10, czyli rezultaty uzyskane w połowie procesu optymalizacji. Kolorem zielonym przedstawiono wyniki optymalne należące do frontu pareto. Celem omawianych wykresów jest pokazanie wpływu procesu optymalizacji (w ramach pojedynczego wykresu) i wartości parametru  $w$  (pomiędzy kolejnymi wykresami) na rozbieżności w uzyskanych wynikach.



Rys. 5.10: Wybrane zbiory rozwiązań dla parametru  $w = 0.2$

Na kolejnych wykresach pokazanych na rysunku 5.10 przedstawiono tę samą formę rozwiązań, co w przypadku rysunku 5.9, jednak tym razem wartość parametru  $w$  była równa 0.2. Na lewym wykresie można zaobserwować wyraźną zmianę w procesie kształtowania się frontu pareto. Położenie punktów reprezentujących otrzymane wyniki jest inne w stosunku do początku układu współrzędnych ponieważ w procesie ich obliczania uwzględniono składnik formuły reprezentujący wartość średnią. Zmiana ta, wyraźnie wpływa też na rozkład wyników w dziedzinie wskaźników  $TPR$  i  $TNR$ , co można zaobserwować na wykresie zamieszczonym z prawej strony. W porównaniu do poprzedniego wyników, przedstawionych na wykresie 5.9, można zauważyć znaczącą zmianę w rozkładzie rozwiązań. Już przy niewielkim uwzględnieniu wartości średnich wskaźników  $TPR$  i  $TNR$ , zbiór rozwiązań optymalnych pokrywa znacząco większy zakres dostępnych rozwiązań. Szczegółowe wyniki porównawcze uzyskane dla wszystkich rozpatrywanych wartości parametru  $w$ , zostały przedstawione w tabeli 5.1 w dalszej części pracy.

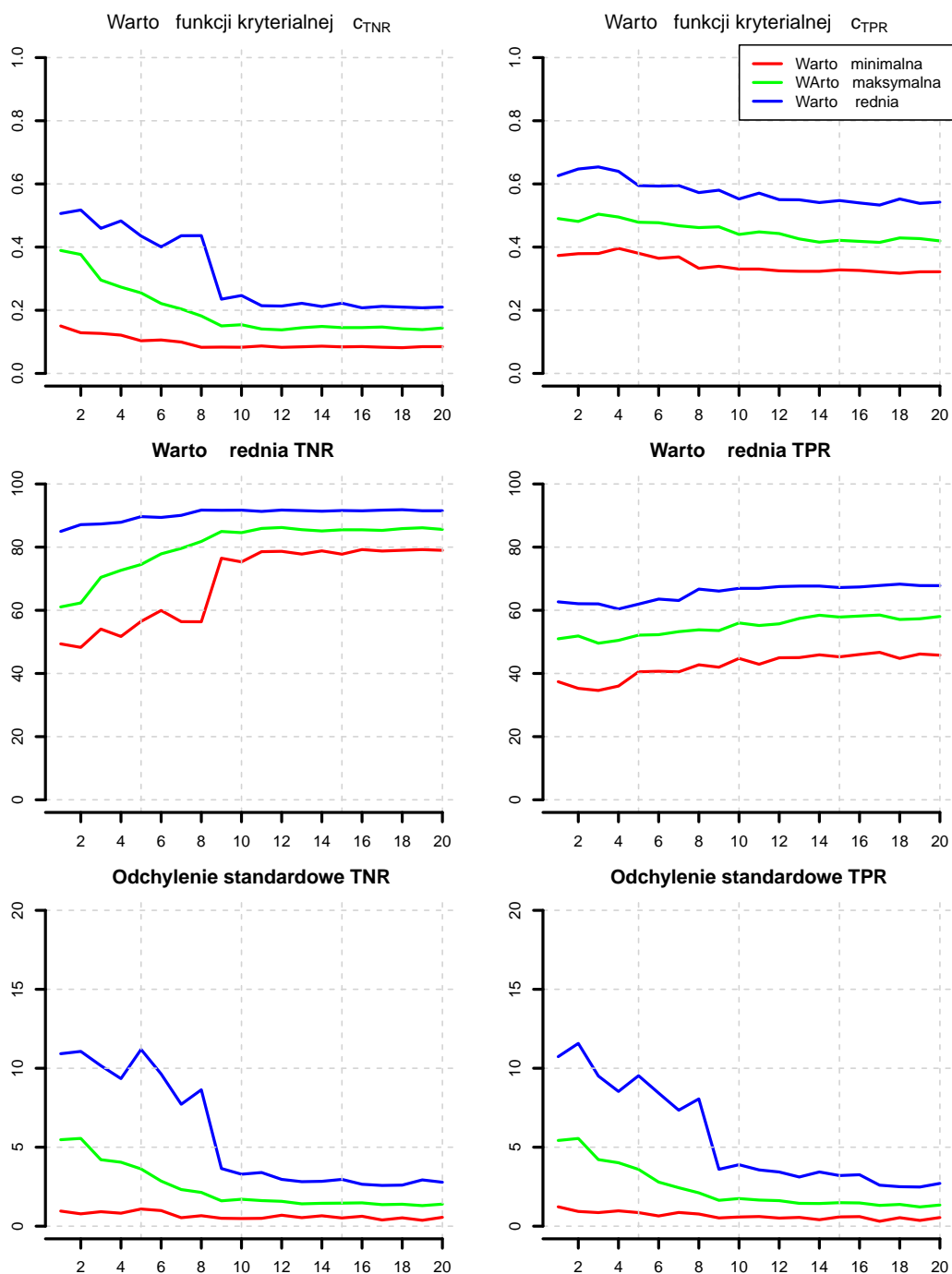
Na kolejnych wykresach przedstawiono analogiczne dane ale uzyskane dla innych wartości parametru  $w$ . Dla  $w = 0.5$  front rozwiązań pareto optymalnych znacząco się rozszerzył i oddalił od początku układu współrzędnych, co zostało pokazano na rysunku



Rys. 5.11: Wybrane zbiory rozwiązań dla parametru  $w = 0.5$

5.11. Jednocześnie rozmieszczenie punktów w dziedzinie wskaźników  $TPR$  i  $TNR$  nie przedstawia aż takich istotnych zmian w rozmieszczeniu punktów na wykresie, porównując uzyskane rezultaty z wykresem przedstawiającym zbiór rozwiązań dla parametru  $w = 0.2$  (5.10). Może to oznaczać, że już niewielkie uwzględnienie wartości średniej w funkcji celu wpływa na ukształtowanie się profilu działania klasyfikatorów, co wynika z mechanizmów w algorytmie optymalizacji. Pozostałe wyniki uzyskane dla parametru  $w$  równego 0.8 i 1.0 potwierdzają tę tezę. Wszystkie wykresy z wynikami zostały zamieszczone w dodatku A na końcu pracy. Szczegółowe różnice zostały przedstawione w formie tabelarycznej (tab. 5.1), w której porównano statystycznie wpływ wartości parametru  $w$  na charakterystykę uzyskanych wyników optymalnych.

W trakcie realizacji procesu optymalizacji za pomocą algorytmu NSGA-II, dla każdego pokolenia wyznaczany jest zbiór rozwiązań, reprezentowany przez populację. Do każdego osobnika w populacji przypisane wartości funkcji kryterialnych. Za pomocą podstawowych miar, możliwe jest wykazanie zmian w populacji względem każdego kolejnego pokolenia. Na rysunku 5.12 zaprezentowano charakter zmian zbiorów wyników dla każdego z pokoleń od 1 do 20, poprzez wyznaczenie wartości minimalnej, średniej oraz maksymalnej określonych wielkości. Dla wykresów umieszczonych w pierwszym rzędzie przedstawiono wartości funkcji kryterialnych  $C_{TNR}$  i  $C_{TPR}$  przedstawione jako wartość maksymalna, minimalna i średnia, obliczona dla każdego pokolenia. Na wykresie zaobserwować można wyraźny spadek wartości funkcji kryterialnej  $C_{TNR}$  i zawężenie puli wyników w zakresie od 1 do 10 pokolenia. W przypadku wartości funkcji kryterialnej  $C_{TPR}$  zmiany są znacznie mniej widoczne. W środkowym wierszu rysunku 5.12 przedstawiono za pomocą wykresów zmianę w otrzymywanych wartościach średnich  $TNR$  i  $TPR$  dla kolejnych pokoleń. Ponownie zakres uzyskiwanych wyników dla pokoleń jest reprezentowany przez wartość minimalną, średnią i maksymalną. W tym przypadku uzyskujemy analogiczne wyniki do tych, które uzyskaliśmy dla funkcji kryterialnych. W ostatnim wierszu umieszczono analogiczne zestawienie dla odchyłeń standardowych, które są składowymi funkcji kryterialnych. Bardzo wyraźnie można zaobserwować, że od 1 do 10 pokolenia odchylenie



Rys. 5.12: Charakterystyka uzyskanych wyników dla poszczególnych pokoleń w trakcie procesu optymalizacji dla  $w = 1.0$

Tab. 5.1: Statystyczne porównanie zbiorów wyników pareto optymalnych ze względu na wartość parametru  $w$ 

Miara		$w = 0.0$	$w = 0.2$	$w = 0.5$	$w = 0.8$	$w = 1.0$
$c_{TPR}$	Minimum	0.003	0.022	0.041	0.066	0.081
	Średnia	0.006	0.033	0.073	0.111	0.136
	Maksimum	0.010	0.051	0.112	0.166	0.206
$c_{TNR}$	Minimum	0.003	0.071	0.166	0.259	0.317
	Średnia	0.004	0.091	0.215	0.345	0.419
	Maksimum	0.006	0.108	0.281	0.448	0.552
$AVG_{TPR}$	Minimum	90.4	79.7	79.2	79.7	79.4
	Średnia	90.6	86.3	86.4	86.7	86.4
	Maksimum	91.2	91.0	92.3	92.0	91.9
$AVG_{TNR}$	Minimum	48.1	48.9	44.8	44.1	44.8
	Średnia	48.7	57.9	58.0	57.2	58.0
	Maksimum	49.5	68.4	68.3	67.9	68.3
$STD_{TPR}$	Minimum	0.297	0.356	0.475	0.445	0.512
	Średnia	0.602	0.742	0.903	1.049	1.222
	Maksimum	1.045	1.273	1.657	1.978	2.570
$STD_{TNR}$	Minimum	0.276	0.476	0.385	0.480	0.363
	Średnia	0.418	0.803	1.058	1.101	1.205
	Maksimum	0.589	1.264	1.740	2.247	2.707

standardowe otrzymanych wyników wyraźnie maleje, a od 10 pokolenia stabilizuje się na zbliżonym poziomie.

W kolejnej części doktoratu szczegółowo omówiono sposób interpretacji i walidacji rozwiązań optymalnych, na przykładzie funkcji kryterialnych 4.8 i 4.7 opisanych w rozdziale 4.4.1. Ponieważ wyniki tam opisane bezpośrednio dotyczą walidacji określonych końcowych rozwiązań uzyskanych w ramach procesu optymalizacji, forma ich przedstawienia związana jest z zagadnieniami detekcji uszkodzeń. Dotychczasowe miary wykorzystywane w zadaniu optymalizacji  $TPR$  i  $TNR$  zastąpione zostały wskaźnikami prawidłowych  $r_{td}$  i fałszywych  $r_{fd}$  alarmów.

### 5.5.2. Analiza wyników detekcji uszkodzeń

W wyniku działania procesu optymalizacji parametrów trenujących klasyfikatory pozyskano zbiór wyników optymalnych. Celem procesu była maksymalizacja wskaźników detekcji stanu pełnej zdatności i stanu z uszkodzeniem z jednoczesnym zachowaniem wysokiej powtarzalności uzyskiwanych wyników. W kolejnym kroku przeanalizowano skuteczność działania pozyskanego rozwiązania poprzez rozdzielanie uzyskanych wyników na poszczególne uszkodzenia oraz stany funkcjonalne. Rezultaty zostaną omówione na podstawie procesu optymalizacji, w którym szukano optymalnych parametrów trenin-



gowych klasyfikatora. Zadaniem klasyfikatora jest detekcja uszkodzenia, gdzie oprócz stanu pełnej zdatności rozpatrywano również wszystkie zasymulowane w eksperymencie uszkodzenia. Jako bazowy klasyfikator kontekstowej metody detekcji wykorzystano klasyfikator drzewa decyzyjnego bazujący na metodzie C4.5. W celu znalezienia rozwiązań optymalnych zastosowano funkcje kryterialne w postaci 4.7 i 4.8, opisane w rozdziale 4.4.1. Funkcje kryterialne posiadają dodatkowy parametr  $t_\alpha$ , który wpływa na istotność odchylenia standardowego na wynik zwracany przez funkcję. W opisywanym przypadku przyjęto wartość parametru  $t_\alpha$  równą 100. Tak przyjęta wartość powoduje, że składowa odpowiedzialna za ocenę powtarzalności uzyskiwanych wyników klasyfikacji jest bardzo znacząca.

### Wstępna analiza rozwiązań optymalnych

Po zakończeniu procesu optymalizacji uzyskano zbiór rozwiązań  $\mathbf{O}$ , który pokazano w dziedzinie wartości średnich detekcji stanu pełnej zdatności i z uszkodzeniem na rysunku 5.13. Do każdego z dostępnych rozwiązań przypisany jest zbiór wartości optymalnych parametrów. Wybór jednego z rozwiązań pozwala na wytrenowanie docelowego klasyfikatora w oparciu o te parametry. Dla każdego klasyfikatora bazującego na drzewie decyzyjnym optymalizowano trzy parametry:

- minimal gain - wartość graniczna definiująca tworzenie dalszych warunków w strukturze drzewa,
- maximal depth - maksymalny rozmiar drzewa obliczany od początku drzewa do najdalszego liścia,
- confidence - wskaźnik pewności wykorzystywany w procesie przycinania drzewa.

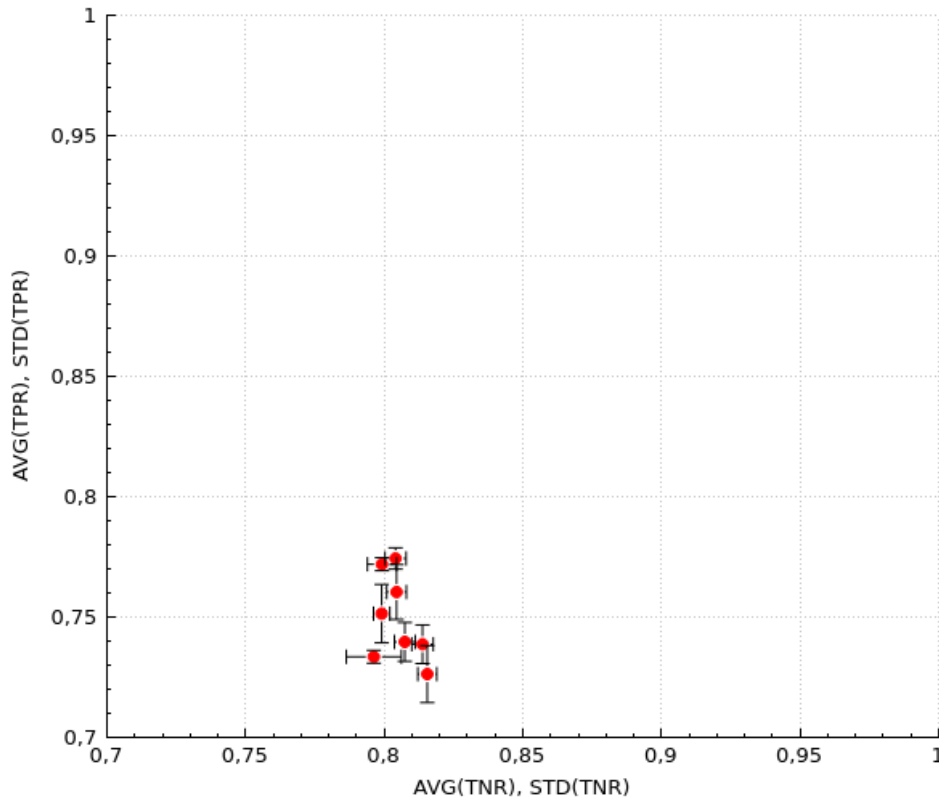
Dla funkcji selekcji atrybutów wejściowych klasyfikatorów optymalizowana jest liczba tych atrybutów (parametr  $k$ ) licząc od najistotniejszego do najmniej istotnego w kontekście problemu detekcji uszkodzeń (klasyfikacji). Parametry klasyfikatorów wybranych rozwiązań optymalnych  $o_i$  pokazano w tabeli 5.3. Poszczególne parametry zostały pogrupowane w tabeli względem operatorów, czyli klasyfikatorów i algorytmów selekcji atrybutów.

Na podstawie zebranych parametrów, dla każdego rozpatrywanego przypadku wytrenowano klasyfikator kontekstowy w oparciu o wszystkie dostępne dane. Następnie wytrenowane klasyfikatory poddano działaniu tych samych danych, aby przeanalizować wpływ stanów funkcjonalnych i uszkodzeń na zdolność rozpoznania przez klasyfikator stanu z uszkodzeniem. W kolejnych tabelach porównano wartości wybranych wskaźników z rozróżnieniem na stany funkcjonalne  $sf$  i dla każdego rozwiązania optymalnego  $o_i$ .

W pierwszej tabeli (5.4) pokazano wskaźniki fałszywych alarmów ( $r_{fd}$ ) dla wszystkich rozwiązań optymalnych z zbioru  $\mathbf{O}$ . W pierwszej kolumnie ( $sf_{0-9}$ ) obliczono wskaźnik fałszywych alarmów dla całego zestawu danych, pokrywających wszystkie stany funkcjonalne ( $sf_0 - sf_9$ ). W kolejnej kolumnie ( $sf_{1-9}$ ) obliczono wskaźnik fałszywych alarmów dla wszystkich głównych stanów funkcjonalnych z pominięciem nieustalonego stanu funkcjonalnego  $sf_0$ . Można zaobserwować różnicę pomiędzy poszczególnymi rozwiązaniami,

Tab. 5.2: Porównanie optymalnych parametrów trzech wybranych rozwiązań ze zbioru  $O$ 

Operator	Parametr	$o_1$	$o_2$	$o_3$
Drzewo decyzyjne 1	minimal gain	0.008	0.006	0.008
	maximal depth	35	42	32
	confidence	0.489	0.171	0.208
Drzewo decyzyjne 2	minimal gain	0.019	0.122	0.096
	maximal depth	30	36	42
	confidence	0.086	0.107	0.100
Drzewo decyzyjne 3	minimal gain	0.139	0.125	0.052
	maximal depth	23	21	24
	confidence	0.465	0.314	0.118
Drzewo decyzyjne 4	minimal gain	0.097	0.116	0.099
	maximal depth	10	45	15
	confidence	0.173	0.496	0.177
Drzewo decyzyjne 5	minimal gain	0.233	0.218	0.195
	maximal depth	6	81	50
	confidence	0.050	0.0198	0.184
Drzewo decyzyjne 6	minimal gain	0.179	0.0777	0.188
	maximal depth	26	47	27
	confidence	0.230	0.347	0.072
Drzewo decyzyjne 7	minimal gain	0.212	0.268	0.207
	maximal depth	95	84	98
	confidence	0.215	0.182	0.204
Drzewo decyzyjne 8	minimal gain	0.179	0.168	0.161
	maximal depth	97	60	5
	confidence	0.412	0.456	0.472
Drzewo decyzyjne 9	minimal gain	0.083	0.043	0.230
	maximal depth	84	48	67
	confidence	0.208	0.408	0.405
Drzewo decyzyjne 10	minimal gain	0.089	0.054	0.095
	maximal depth	24	60	67
	confidence	0.041	0.342	0.036
Selekcja atrybutów 1	k	49	22	8
Selekcja atrybutów 2	k	44	48	20
Selekcja atrybutów 3	k	40	11	19
Selekcja atrybutów 4	k	37	49	50
Selekcja atrybutów 5	k	40	16	42
Selekcja atrybutów 6	k	35	35	31
Selekcja atrybutów 7	k	50	47	49
Selekcja atrybutów 8	k	36	3	38
Selekcja atrybutów 9	k	21	21	22
Selekcja atrybutów 10	k	3	36	35



Rys. 5.13: Zbiór rozwiązań optymalnych dla parametru  $t_\alpha = 100$

Tab. 5.4: Porównanie wskaźników fałszywych alarmów wyrażonych w procentach ze względu na stany funkcjonalne  $sf_i$  dla poszczególnych rozwiązań optymalnych  $o_i$

$O$	$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$o_1$	2.7	2.5	2.8	2.8	0.9	0.9	1.8	0.9	<b>10.0</b>	1.8	1.8	1.8
$o_2$	2.7	1.4	3.9	2.8	1.8	0.0	1.8	0.9	0.9	1.8	2.7	0.0
$o_3$	2.3	2.4	2.2	2.8	0.9	0.9	0.9	2.7	<b>10.0</b>	0.0	1.8	1.8
$o_4$	4.6	6.5	2.6	<b>43.5</b>	0.9	2.7	1.8	1.8	2.7	1.8	0.0	3.7
$o_5$	2.6	3.2	1.9	0.9	0.9	2.7	0.9	3.6	<b>10.0</b>	1.8	4.5	3.7
$o_6$	4.1	5.3	2.9	2.8	0.9	3.6	1.8	1.8	<b>10.0</b>	<b>24.5</b>	0.0	1.8
$o_7$	5.1	7.3	2.8	<b>43.5</b>	0.9	0.0	0.9	2.7	<b>10.0</b>	0.0	5.5	2.8
$o_8$	4.9	5.1	4.7	10.2	2.7	0.0	2.7	1.8	0.9	0.9	<b>22.7</b>	3.7

co wskazuje na istotny wpływ tego stanu funkcjonalnego na liczbę generowanych fałszywych alarmów, co bezpośrednio można zaobserwować w kolejnej kolumnie  $sf_0$ . W każdej z kolumn od  $sf_0$  do  $sf_9$  pokazano procentową liczbę fałszywych alarmów w zbiorze danych przypisanym do konkretnego stanu funkcjonalnego. W większości przypadków można zaobserwować niewielkie różnice w obrębie stanów funkcjonalnych, nieprzekraczających paru procent. Wyjątkiem jest stan funkcjonalny  $sf_6$ , który dla większości badanych rozwiązań uzyskuje wskaźnik liczby fałszywych alarmów równy 10% w sytuacji, gdy klasyfikator został przetestowany tymi samymi danymi, którymi został wytrenowany. Może to wskazywać na znaczące problemy w rozróżnieniu uszkodzeń i stanów pełnej zdadności w tym stanie funkcjonalnym. Wśród innych rozwiązań, można zaobserwować nieliczne przypadki, w których wskaźnik fałszywych alarmów znacząco rośnie w konkretnych sta-

nach funkcjonalnych. Jest to np.  $sf_1$  dla rozwiązań  $o_4$  i  $o_7$ , gdzie wskaźnik fałszywych alarmów osiąga wartość 43.5%. Innym przykładem jest rozwiązanie  $o_6$  w stanie funkcjonalnym  $sf_7$ , czy rozwiązanie  $o_8$  w  $sf_8$ , gdzie wskaźnik  $r_{fd}$  przekroczył wartość 20%. Tego typu anomalie mogą wskazywać na przystosowanie parametrów tych klasyfikatorów w określonych rozwiązaniach  $o_i$  do większej skuteczności detekcji uszkodzeń kosztem niskiego wskaźnika fałszywych alarmów.

W kolejnych tabelach zaprezentowano analogiczne wyniki jak w przypadku tabeli 5.4, jednak w tym przypadku wartość liczbowa zapisana w komórkach tabeli reprezentuje wskaźnik prawidłowych alarmów  $r_{td}$ . Wskaźnik ten został podzielony ze względu na zaimplementowane uszkodzenia, aby przeanalizować wpływ poszczególnych uszkodzeń na skuteczność detekcji w danym rozwiązaniu i stanie funkcjonalnym.

Tab. 5.5: Porównanie wskaźników prawidłowych alarmów wyrażonych w procentach dla uszkodzenia  $f_1$  ze względu na stany funkcjonalne dla poszczególnych rozwiązań

<b>O</b>	$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$o_1$	90.0	84.4	95.6	100.0	90.0	100.0	<b>0.0</b>	100.0	90.0	100.0	90.0	90.0
$o_2$	80.0	76.7	83.3	100.0	100.0	90.0	<b>20.0</b>	100.0	90.0	100.0	0.0	90.0
$o_3$	90.6	93.3	87.8	100.0	90.0	100.0	80.0	100.0	90.0	100.0	90.0	90.0
$o_4$	96.1	96.7	95.6	90.0	100.0	100.0	100.0	100.0	100.0	100.0	90.0	90.0
$o_5$	83.9	77.8	90.0	<b>40.0</b>	100.0	100.0	<b>0.0</b>	100.0	90.0	90.0	90.0	90.0
$o_6$	91.1	86.7	95.6	100.0	100.0	100.0	100.0	100.0	90.0	100.0	0.0	90.0
$o_7$	84.4	73.3	95.6	90.0	100.0	0.0	<b>0.0</b>	100.0	90.0	100.0	100.0	80.0
$o_8$	90.0	86.7	93.3	<b>10.0</b>	100.0	90.0	100.0	100.0	90.0	90.0	100.0	100.0

W tabeli 5.5 przedstawiono wartości prawidłowych alarmów dla uszkodzenia  $f_1$ . Założyć można wyraźny wpływ stanu nieustalonego  $sf_0$ , którego uwzględnienie w większości przypadków pozytywnie wpływa na rozpoznanie analizowanego uszkodzenia. W większości stanów funkcjonalnych udaje się uzyskać wysoki wskaźnik detekcji uszkodzenia oprócz pojedynczych przypadków dla konkretnych stanów funkcjonalnych i rozwiązań, tj. stan funkcjonalny  $sf_1$  w rozwiązaniach  $o_5$  i  $o_8$ , gdzie  $r_{td}$  wyniósł odpowiednio 40% i 10%. Istotnym elementem w analizowanych wynikach jest stan funkcjonalny  $sf_4$ , w którym dla połowy rozwiązań uzyskano wynik równy lub bliski 0%. Może to wskazywać na późniejsze problemy w poprawnej detekcji uszkodzenia w tym stanie funkcjonalnym.

Tab. 5.6: Porównanie wskaźników prawidłowych alarmów wyrażonych w procentach dla uszkodzenia  $f_7$  ze względu na stany funkcjonalne dla poszczególnych rozwiązań

<b>O</b>	$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$o_1$	77.8	64.4	91.1	90.0	70.0	100.0	0.0	0.0	50.0	80.0	100.0	<b>90.0</b>
$o_2$	<b>58.9</b>	<b>46.7</b>	71.1	90.0	90.0	90.0	<b>0.0</b>	<b>0.0</b>	<b>40.0</b>	<b>20.0</b>	<b>0.0</b>	<b>90.0</b>
$o_3$	81.7	80.0	83.3	90.0	80.0	80.0	90.0	60.0	50.0	80.0	100.0	<b>90.0</b>
$o_4$	80.6	71.1	90.0	90.0	60.0	70.0	80.0	60.0	80.0	80.0	30.0	<b>90.0</b>
$o_5$	71.1	60.0	82.2	20.0	70.0	70.0	30.0	90.0	50.0	20.0	100.0	<b>90.0</b>
$o_6$	76.7	62.2	91.1	90.0	80.0	80.0	80.0	60.0	50.0	30.0	0.0	<b>90.0</b>
$o_7$	70.6	50.0	91.1	90.0	70.0	0.0	30.0	0.0	50.0	20.0	100.0	<b>90.0</b>
$o_8$	73.9	68.9	78.9	10.0	80.0	90.0	90.0	80.0	40.0	60.0	70.0	<b>100</b>

W tabeli 5.6 można zaobserwować bardzo wyraźne pogorszenie detekcji uszkodzenia  $f_7$ . Wskaźnik prawidłowych alarmów dla wszystkich danych i większości rozwiązań optymalnych nie przekracza wartości 80%, a przy ograniczeniu danych tylko do stanów funkcjonalnych od  $sf_1$  do  $sf_9$  zauważyć można znaczące pogorszenie, co wskazuje na dobre przystosowanie klasyfikatora działającego w stanie funkcjonalnym  $sf_0$  do detekcji tego uszkodzenia. Dla kolejnych stanów funkcjonalnych wyniki znacząco się od siebie różnią, gdzie w stanie funkcjonalnym  $sf_9$  dla wszystkich analizowanych rozwiązań wskaźnik prawidłowych alarmów jest równy 90% lub 100%, a dla pozostałych stanów funkcjonalnych wyniki w znacznym stopniu zależą od analizowanego rozwiązania. Rozwiązanie  $o_2$  wyróżnia się wyraźnie gorszym wynikiem ogólnym, co przekłada się na wyniki w określonych stanach funkcjonalnych. Wyniki wskazują, że w dalszej części weryfikacji rozwiązań, można spodziewać się znaczących trudności w detekcji uszkodzenia  $f_7$ .

Dla wszystkich pozostałych uszkodzeń klasyfikator kontekstowy osiągał bardzo dobre wyniki, często wykraczające ponad wartość 90%. Jedynie w pojedynczych przypadkach można zaobserwować wyraźne pogorszenie detekcji danego uszkodzenia, które jest związane z konkretnym rozwiązaniem i stanem funkcjonalnym. Pozostałe wyniki zostały przedstawione w dodatku A niniejszej pracy. Ciekawym przypadkiem jest detekcja uszkodzeń w rozwiązaniu  $o_7$  w stanie funkcjonalnym  $sf_3$ . Wskaźnik prawidłowych alarmów dla wszystkich uszkodzeń w tym konkretnym przypadku jest równy 0%, co oznacza brak zdolności klasyfikatora do detekcji jakiegokolwiek uszkodzenia. Zostało to również potwierdzone w tabeli 5.7, która prezentuje zbiorcze wyniki detekcji wszystkich uszkodzeń w każdym rozwiązaniu optymalnym z uwzględnieniem podziału na stany funkcjonalne.

Tab. 5.7: Porównanie wskaźników prawidłowych alarmów wyrażonych w procentach dla wszystkich uszkodzeń ze względu na stany funkcjonalne dla poszczególnych rozwiązań

O	$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$o_1$	95.0	92.9	97.2	98.1	91.9	97.1	80.7	87.2	89.3	97.1	97.3	97.4
$o_2$	89.5	88.4	90.7	98.1	97.3	97.1	82.6	87.2	87.5	91.3	57.3	97.4
$o_3$	95.2	95.1	95.2	98.1	95.5	97.1	97.2	93.6	84.8	95.2	97.3	97.4
$o_4$	96.2	95.2	97.2	98.1	91.9	96.2	98.2	93.6	96.4	96.2	89.1	97.4
$o_5$	92.1	89.3	95.0	69.4	92.8	96.2	80.7	98.2	84.8	86.5	97.3	97.4
$o_6$	94.2	90.9	97.4	98.1	96.4	98.1	98.2	93.6	89.3	91.3	56.4	97.4
$o_7$	89.4	81.7	97.2	98.1	94.6	<b>0.0</b>	80.7	87.2	84.8	89.4	100.0	95.6
$o_8$	91.9	91.4	92.4	55.6	95.5	97.1	99.1	97.2	87.5	95.2	96.4	99.1

Przeanalizowano skuteczność działania wytrenowanych klasyfikatorów kontekstowych dla wszystkich rozwiązań optymalnych. Weryfikacja została zrealizowana w oparciu o te same dane, na których klasyfikatory były trenowane i jej celem była wstępna ocena trudności detekcji uszkodzeń oraz stanu pełnej zdatności w poszczególnych stanach funkcjonalnych. W kolejnym podrozdziale poddano wybrane rozwiązania dodatkowej analizie z wykorzystaniem weryfikacyjnego zbioru danych.

## Walidacja rozwiązań optymalnych

Walidacja wybranych rozwiązań została przeprowadzona w oparciu o dwa niezależne zbiory danych. Pierwszy z nich wykorzystany do wytrenowania klasyfikatorów składał się z danych zebranych podczas 99 eksperymentów dla stanu pełnej zdadności oraz dla 11 uszkodzeń po 9 eksperymentów każdy. W rezultacie dostępny jest zbiór danych treningowych składający się z 99 eksperymentów dla stanu pełnej zdadności i 99 eksperymentów dla stanu z uszkodzeniem. Tak przygotowane dane są bardzo dobrze zbalansowane dla problemu detekcji. Z pozostałych danych utworzony został zbiór danych walidacyjnych, które zawierają dane z 11 eksperymentów dla stanu pełnej zdadności i 11 eksperymentów dla stanu z uszkodzeniem (po 1 eksperymencie dla każdego z uszkodzeń). Dane zostały przydzielone do zbiorów treningowych i testowych w sposób losowy. Dodatkowo aby nie ograniczać się tylko do jednego przypadku walidacji, w analogiczny sposób przygotowano 4 dodatkowe zestawy danych treningowo walidacyjnych, co ostatecznie daje 5 zestawów danych. W kolejnych tabelach przedstawiono wyniki tej analizy, gdzie każda tabela zawiera wartości wszystkich wskaźników wyliczonych dla stanów funkcjonalnych i uszkodzeń dla danego rozwiązania. W wierszu  $f_0$  zawarto wartości wskaźników fałszywych alarmów  $r_{fd}$ , a w pozostałych wierszach ( $f_1 - f_{11}$ ) wskaźniki prawidłowych alarmów  $r_{td}$ . W ostatnim wierszu ( $f_{1-11}$ ) pokazano ogólny wskaźnik prawidłowych alarmów dla wszystkich uszkodzeń. Kolumny, podobnie jak w przypadku tabel dotyczących wstępnej weryfikacji, dotyczą poszczególnych stanów funkcjonalnych, gdzie dwie pierwsze kolumny określają wyniki dla wszystkich danych oraz wszystkich danych z pominięciem stanu funkcjonalnego  $sf_0$ . Ze względu na bardzo ograniczoną liczbę danych walidacyjnych, rozdzielczość zmian w poszczególnych wskaźnikach sięga paru procent już dla pojedynczej próbki danych. Wyniki w kolejnych tabelach przedstawiono w postaci zbiorczej dla wszystkich 5 zestawów danych treningowych i walidacyjnych. Górna wartość w komórce oznacza wartość średnią danego wskaźnika dla rozwiązania i stanu funkcjonalnego, natomiast dolna wartość w nawiasie oznacza odchylenie standardowe tej wartości.

W pierwszym rozpatrywanym rozwiązaniu przedstawionym w tabeli 5.8, można zauważyć wysoki wskaźnik fałszywych alarmów dla stanu funkcjonalnego  $sf_0$ . Usunięcie tego stanu ze zbioru danych walidacyjnych pozwala na wyraźną poprawę tego wskaźnika. W innych stanach funkcjonalnych ( $sf_1, sf_3, sf_6, sf_9$ ) również można zaobserwować wysoki wskaźnik fałszywych alarmów, które mają istotny wpływ na ogólną miarę  $r_{fd}$ . Dla przedziałów z uszkodzeniem, usunięcie danych przypisanych do stanu funkcjonalnego  $sf_0$  najczęściej poprawia stopień detekcji konkretnego uszkodzenia. Wśród wszystkich analizowanych uszkodzeń, dla dwóch wskaźnik poprawnych alarmów utrzymuje się powyżej wartości 90% zarówno dla pełnego zakresu stanów funkcjonalnych jak i dla grupy stanów funkcjonalnych bez  $sf_0$ , tj.  $f_2$  i  $f_8$ . Innym uszkodzeniem, dla którego klasyfikatory wykazują wysoki stopień detekcji stanu z uszkodzeniem jest  $f_3$ . W tym przypadku dopiero pominięcie stanu funkcjonalnego  $sf_0$  pozwoliło na uzyskanie wartości wskaźnika prawidłowych alarmów powyżej 90%. Analizując uzyskane wyniki z perspektywy najgorszej rozpoznanych uszkodzeń łatwo można zauważyć uszkodzenie  $f_7$ , którego wskaźnik

Tab. 5.8: Zestawienie wskaźników  $r_{fd}$  i  $r_{td}$  wyrażonych w procentach dla rozwiązania optymalnego  $o_1$ 

	$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$f_0$	19.1 (3.9)	8.4 (2.6)	29.8 (5.4)	14.0 (8.9)	8.0 (8.4)	22.0 (14.8)	2.0 (4.5)	0.0 (0.0)	10.0 (10.0)	8.0 (8.4)	2.0 (4.5)	10.0 (10.0)
$f_1$	71.1 (17.3)	75.6 (12.2)	66.7 (26.1)	100.0 (0.0)	100.0 (0.0)	20.0 (44.7)	0.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)
$f_2$	<b>90.2</b> (9.6)	<b>95.8</b> (5.8)	84.4 (14.9)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)
$f_3$	86.7 (6.3)	<b>91.1</b> (9.3)	82.2 (9.9)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	60.0 (54.8)	100.0 (0.0)
$f_4$	85.6 (8.4)	88.9 (7.9)	82.2 (14.9)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	60.0 (54.8)	80.0 (44.7)
$f_5$	84.4 (7.2)	86.7 (18.3)	82.2 (16.9)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	60.0 (54.8)
$f_6$	83.3 (8.8)	91.1 (12.2)	75.6 (9.3)	60.0 (54.8)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)
$f_7$	<b>26.7</b> (8.2)	<b>31.1</b> (12.2)	22.2 (7.9)	40.0 (54.8)	60.0 (54.8)	20.0 (44.7)	20.0 (44.7)	20.0 (44.7)	20.0 (44.7)	0.0 (0.0)	60.0 (54.8)	40.0 (54.8)
$f_8$	<b>95.6</b> (4.6)	<b>91.1</b> (9.3)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	40.0 (54.8)	100.0 (0.0)
$f_9$	65.7 (9.8)	79.7 (9.0)	51.7 (26.0)	40.0 (54.8)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	60.0 (54.8)	60.0 (54.8)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)
$f_{10}$	72.2 (14.2)	84.4 (6.1)	60.0 (29.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	40.0 (54.8)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)
$f_{11}$	81.9 (10.7)	85.8 (14.9)	77.7 (10.0)	60.0 (54.8)	100.0 (0.0)	20.0 (44.7)	80.0 (44.7)	80.0 (44.7)	80.0 (44.7)	20.0 (44.7)	100.0 (0.0)	80.0 (44.7)
$f_{1-11}$	76.6 (2.7)	81.9 (1.6)	71.4 (5.6)	78.2 (10.4)	94.5 (5.0)	76.4 (5.9)	77.6 (5.8)	78.2 (18.9)	78.5 (13.9)	86.4 (9.2)	81.8 (9.1)	84.7 (8.9)

prawkłowych alarmów wynosi niewiele powyżej 31% dla grupy stanów funkcjonalnych bez  $sf_0$ . Wynik uzyskany dla uszkodzenia  $f_7$  potwierdza wcześniejsze obserwacje, które były rezultatem wstępnej analizy rozwiązań optymalnych. Wskaźnik prawkłowych alarmów dla pozostałych uszkodzeń zamyka się w przedziale od 75% do 90%. Analizując uzyskane wyniki z perspektywy stanów funkcjonalnych, można zaobserwować licznie występujące wskaźniki prawkłowych alarmów równe wartości 100%, co wskazuje na bardzo wysoką zdolność tych klasyfikatorów do detekcji określonego uszkodzenia.

W tabeli 5.9 przedstawiono wyniki dla rozwiązania optymalnego  $o_2$ . Porównując te wyniki do wyników uzyskanych dla rozwiązania  $o_1$ , można zauważyć pewne podobieństwa. Dla uszkodzeń  $f_2$ ,  $f_3$ ,  $f_8$  klasyfikator kontekstowy wciąż wykazuje najwyższą wartość wskaźnika prawkłowych alarmów przy jednocześnie najniższym wskaźniku dla uszkodzenia  $f_7$ . Dla większości pozostałych uszkodzeń klasyfikator charakteryzuje się niższą skutecznością detekcji tych uszkodzeń w porównaniu do wyników uzyskanych dla rozwiązania  $o_1$ . Potwierdzeniem tego jest ogólny wskaźnik detekcji uszkodzenia znajdujący się w wierszu  $f_{1-11}$ , gdzie wartość 77.8% uzyskana dla wszystkich stanów funkcjonalnych bez  $sf_0$  jest o około 4 punkty procentowe gorsza w stosunku do analogicznej wartości w rozwiązaniu  $o_1$ . Jednak porównując wskaźnik prawkłowych alarmów w kolumnie

Tab. 5.9: Zestawienie wskaźników  $r_{fd}$  i  $r_{td}$  wyrażonych w procentach dla rozwiązania optymalnego  $o_2$ 

	$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$f_0$	15.3 (3.2)	<b>6.7</b> (1.1)	24.0 (6.9)	6.0 (8.9)	12.0 (8.4)	8.0 (11.0)	4.0 (8.9)	0.0 (0.0)	10.0 (10.0)	4.0 (8.9)	0.0 (0.0)	16.0 (5.5)
$f_1$	62.2 (22.7)	62.2 (16.9)	62.2 (33.9)	60.0 (54.8)	100.0 (0.0)	20.0 (44.7)	20.0 (44.7)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	0.0 (0.0)	80.0 (44.7)
$f_2$	<b>95.6</b> (2.5)	<b>97.8</b> (5.0)	93.3 (6.1)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)
$f_3$	<b>94.4</b> (5.6)	<b>95.6</b> (9.9)	93.3 (9.9)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)
$f_4$	87.8 (9.9)	88.9 (11.1)	86.7 (9.3)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	40.0 (54.8)	60.0 (54.8)
$f_5$	78.9 (14.9)	77.8 (28.3)	80.0 (9.3)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	80.0 (44.7)	80.0 (44.7)	40.0 (54.8)	60.0 (54.8)
$f_6$	80.0 (6.3)	91.1 (9.3)	68.9 (9.3)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)
$f_7$	<b>22.2</b> (5.6)	<b>20.0</b> (12.2)	24.4 (9.3)	20.0 (44.7)	60.0 (54.8)	20.0 (44.7)	20.0 (44.7)	0.0 (0.0)	40.0 (54.8)	0.0 (0.0)	0.0 (0.0)	20.0 (44.7)
$f_8$	<b>95.6</b> (4.6)	<b>91.1</b> (9.3)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	40.0 (54.8)	100.0 (0.0)
$f_9$	68.5 (8.5)	75.3 (14.2)	61.7 (21.2)	60.0 (54.8)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	60.0 (54.8)	40.0 (54.8)	80.0 (44.7)	40.0 (54.8)	100.0 (0.0)
$f_{10}$	70.0 (11.5)	71.1 (9.9)	68.9 (14.5)	40.0 (54.8)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	40.0 (54.8)	100.0 (0.0)	20.0 (44.7)	80.0 (44.7)
$f_{11}$	75.5 (9.2)	85.8 (12.0)	65.2 (20.5)	60.0 (54.8)	100.0 (0.0)	20.0 (44.7)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	20.0 (44.7)	60.0 (54.8)	100.0 (0.0)
$f_{1-11}$	75.5 (4.7)	<b>77.8</b> (5.4)	73.2 (4.9)	72.7 (17.0)	96.4 (5.0)	80.2 (7.4)	79.5 (8.3)	78.2 (15.2)	80.3 (10.1)	82.4 (11.3)	47.3 (25.2)	83.2 (11.7)

$sf_{1-9}$ , można zaobserwować poprawę w rozwiązaniu  $o_2$  w stosunku do rozwiązania  $o_1$ .

Dla kolejnego rozwiązania optymalnego  $o_3$  (tab. 5.10), widoczna jest ogólna poprawa detekcji stanu z uszkodzeniem, co jest widoczne zarówno w ostatnim wierszu  $f_{1-11}$ , który zawiera całościową wartość wskaźnika prawidłowych alarmów, jak i w poszczególnych uszkodzeniach, gdzie oprócz uszkodzeń  $f_2$ ,  $f_3$  i  $f_8$  pojawiają się dwa dodatkowe uszkodzenia ( $f_6$  i  $f_{11}$ ), które dla wszystkich stanów funkcjonalnych bez  $sf_0$  również przekroczyły próg 90% skuteczności detekcji stanu z uszkodzeniem przez klasyfikator kontekstowy. Uszkodzenie  $f_7$  pomimo wciąż bardzo niskiej wartości wskaźnika prawidłowych alarmów, w rozwiązaniu  $o_3$  uzyskało najwyższy wynik spośród wszystkich analizowanych rozwiązań. Negatywną konsekwencją wysokiej sprawności klasyfikatora kontekstowego do detekcji stanu z uszkodzeniem jest jeden z najwyższych wskaźników fałszywych alarmów w porównaniu do wszystkich rozpatrywanych rozwiązań.

Dla kolejnych rozwiązań opisany wyżej schemat rozkładu wskaźników  $r_{fd}$  i  $r_{td}$  kształtuje się analogicznie do wyżej opisanych rezultatów, dlatego postanowiono w dalszej części rozdziału zamieścić skrócone wersje wyników dla pozostałych rozwiązań, ograniczające się jedynie do wskaźnika fałszywych alarmów i prawidłowych alarmów wyznaczonych dla wszystkich rozpatrywanych uszkodzeń. Pełne wersje tabel z wynikami zostały



Tab. 5.10: Zestawienie wskaźników  $r_{fd}$  i  $r_{td}$  wyrażonych w procentach dla rozwiązania optymalnego  $o_3$ 

	$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$f_0$	19.8 (2.7)	12.4 (4.4)	27.1 (2.8)	18.0 (8.4)	18.0 (14.8)	24.0 (18.2)	4.0 (5.5)	8.0 (8.4)	14.0 (8.9)	8.0 (8.4)	2.0 (4.5)	16.0 (5.5)
$f_1$	74.4 (16.0)	84.4 (12.7)	64.4 (25.3)	80.0 (44.7)	100.0 (0.0)	20.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)
$f_2$	<b>92.4</b> (6.0)	<b>98.0</b> (4.5)	86.7 (9.3)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)
$f_3$	88.9 (5.6)	<b>91.1</b> (9.3)	86.7 (9.3)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	60.0 (54.8)	100.0 (0.0)
$f_4$	85.6 (10.8)	88.9 (7.9)	82.2 (14.9)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	60.0 (54.8)	60.0 (54.8)
$f_5$	86.7 (3.0)	88.9 (13.6)	84.4 (12.7)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	60.0 (54.8)
$f_6$	82.2 (4.6)	<b>95.6</b> (6.1)	68.9 (9.3)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)
$f_7$	<b>35.6</b> (15.0)	<b>40.0</b> (14.9)	31.1 (21.4)	20.0 (44.7)	60.0 (54.8)	20.0 (44.7)	60.0 (54.8)	40.0 (54.8)	60.0 (54.8)	20.0 (44.7)	60.0 (54.8)	20.0 (44.7)
$f_8$	<b>96.7</b> (3.0)	<b>93.3</b> (6.1)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	40.0 (54.8)	100.0 (0.0)
$f_9$	75.0 (9.2)	84.4 (12.7)	65.6 (14.9)	40.0 (54.8)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	100.0 (0.0)	60.0 (54.8)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)
$f_{10}$	71.1 (16.4)	82.2 (6.1)	60.0 (32.0)	60.0 (54.8)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	60.0 (54.8)	60.0 (54.8)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)
$f_{11}$	83.1 (11.6)	<b>90.3</b> (10.4)	75.8 (14.6)	60.0 (54.8)	100.0 (0.0)	20.0 (44.7)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	20.0 (44.7)	100.0 (0.0)	100.0 (0.0)
$f_{1-11}$	79.2 (4.5)	85.1 (2.0)	73.2 (8.3)	74.5 (17.5)	94.5 (5.0)	80.4 (7.1)	96.2 (5.2)	85.5 (10.4)	83.9 (13.4)	86.4 (9.2)	81.8 (9.1)	83.2 (11.7)

zamieszczone w dodatku A w końcowej części pracy.

Celem zadania optymalizacji było uzyskanie rozwiązań, które minimalizują liczbę fałszywych alarmów, maksymalizują liczbę prawidłowych alarmów oraz minimalizuje wartość odchylenia standardowego dla każdej z tych miar. Wszystkie analizowane rozwiązania od  $o_1$  do  $o_8$  są rozwiązaniami pareto optymalnymi, co oznacza że każde z tych rozwiązań spełnia kryterium optymalności, jednak w zależności od umiejscowienia rozwiązania w przestrzeni dziedzin funkcji kryterialnych, dane rozwiązanie może charakteryzować się innym wynikiem, co można obserwować na omówionych przykładach. W tabeli 5.11 przedstawiono uogólnione wyniki dla pozostałych rozwiązań. W tabeli ograniczono się jedynie do porównania wskaźników fałszywych alarmów  $r_{fd}$  i prawidłowych alarmów  $r_{td}$  uwzględniając wszystkie uszkodzenia. Podobnie jak w poprzednich tabelach przedstawiono również jak wymienione wskaźniki zmieniają się ze względu na stany funkcjonalne. Zmienność wyników pokazanych w tabeli 5.11 w znacznym stopniu zależy od rozwiązania  $o_i$ , co pokazuje, że dobrane parametry algorytmów trenujących klasyfikatory mają istotny wpływ na model klasyfikatora i jego zdolność klasyfikacji w różnych przypadkach zależnych od stanów funkcjonalnych i rozpatrywanych uszkodzeń.

W ostatniej tabeli z wynikami (tab. 5.12) pokazano wartości wskaźników  $r_{fd}$  i  $r_{td}$ , dla

Tab. 5.11: Zestawienie wskaźników  $r_{fd}$  i  $r_{td}$  wyrażonych w procentach dla pozostałych rozwiązań optymalnych

		$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$o_4$	$f_0$	19.9 (2.5)	11.3 (2.9)	28.4 (4.2)	32.0 (14.8)	12.0 (8.4)	8.0 (11.0)	10.0 (7.1)	4.0 (5.5)	14.0 (13.4)	6.0 (5.5)	4.0 (5.5)	12.0 (8.4)
	$f_{1-11}$	76.9 (2.6)	83.5 (2.5)	70.3 (5.7)	81.8 (17.0)	94.5 (5.0)	80.2 (7.4)	86.9 (10.6)	80.0 (16.3)	85.5 (10.4)	86.4 (9.2)	72.7 (17.0)	82.9 (8.2)
$o_5$	$f_0$	17.6 (1.7)	8.0 (1.8)	27.1 (3.4)	12.0 (8.4)	12.0 (8.4)	8.0 (11.0)	6.0 (13.4)	0.0 (0.0)	14.0 (8.9)	8.0 (8.4)	0.0 (0.0)	12.0 (8.4)
	$f_{1-11}$	77.7 (3.9)	83.3 (2.8)	72.2 (8.7)	69.1 (15.2)	94.5 (5.0)	80.2 (7.4)	81.3 (9.8)	85.5 (12.2)	83.9 (13.4)	88.4 (4.7)	81.8 (12.9)	84.7 (8.9)
$o_6$	$f_0$	20.9 (4.6)	12.2 (4.0)	29.6 (6.1)	18.0 (8.4)	14.0 (11.4)	22.0 (16.4)	12.0 (8.4)	0.0 (0.0)	10.0 (10.0)	24.0 (11.4)	0.0 (0.0)	10.0 (7.1)
	$f_{1-11}$	74.9 (3.1)	78.2 (3.7)	71.6 (6.3)	76.4 (10.4)	94.5 (5.0)	76.4 (5.9)	88.7 (8.1)	78.2 (15.2)	78.5 (13.9)	78.7 (5.0)	47.3 (23.5)	84.7 (8.9)
$o_7$	$f_0$	20.8 (3.8)	11.8 (3.6)	29.8 (5.4)	44.0 (11.4)	14.0 (11.4)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	10.0 (10.0)	8.0 (8.4)	8.0 (11.0)	22.0 (21.7)
	$f_{1-11}$	72.6 (3.1)	73.9 (1.7)	71.4 (5.6)	98.2 (4.1)	94.5 (5.0)	0.0 (0.0)	77.6 (8.7)	78.2 (8.1)	78.5 (13.9)	84.4 (13.6)	80.0 (11.9)	69.4 (11.8)
$o_8$	$f_0$	18.4 (4.8)	11.3 (3.5)	25.6 (6.6)	4.0 (5.5)	26.0 (18.2)	8.0 (11.0)	4.0 (8.9)	8.0 (8.4)	10.0 (10.0)	8.0 (11.0)	22.0 (17.9)	12.0 (8.4)
	$f_{1-11}$	74.1 (3.2)	79.0 (4.1)	69.1 (3.8)	52.7 (11.9)	85.5 (10.4)	80.2 (7.4)	79.5 (8.3)	78.2 (8.1)	80.3 (10.1)	88.4 (4.7)	83.6 (17.5)	83.0 (9.9)

pojedynczego klasyfikatora drzewa decyzyjnego. Parametry klasyfikatora zostały dobrane arbitralnie bazując na doświadczeniu, wiedzy i wynikach uzyskiwanych za pomocą metody walidacji krzyżowej. Podczas dostrajania parametrów algorytmu trenującego drzewo decyzyjne, dążono do minimalizacji ogólnego błędu klasyfikacji oraz minimalizacji wskaźnika fałszywych alarmów. Rozpatrzono dwa podejścia, pierwsze polegało na uwzględnieniu cechy kontekstowej jako cechę podstawową, która była interpretowana przez klasyfikator podobnie jak pozostałe cechy w danych trenujących. W tym przypadku algorytm trenujący drzewo decyzyjne sam decydował o tym, czy daną cechę wykorzystać w warunkach będących elementami struktury drzewa. Drugie podejście usuwało cechę kontekstową z danych trenujących, co odgórnie wykluczało wykorzystanie tej cechy przez klasyfikator. W obu przypadkach uzyskano dokładnie te same wyniki, co pozwala sądzić, że z punktu widzenia algorytmu trenującego drzewo decyzyjne, informacja zawarta w cesze kontekstowej nie jest istotna w aspekcie detekcji stanu pełnej zdatności i stanu z uszkodzeniem. Porównując wyniki z tabeli 5.12 z wynikami uzyskanymi dla rozwiązań bazujących na klasyfikatorze kontekstowym, można zaobserwować pewne prawidłowości. Pojedynczy klasyfikator uzyskuje nieco lepsze wyniki dla wszystkich stanów funkcjonalnych, co widoczne jest w kolumnie  $sf_{0-9}$ . Usunięcie danych związanych ze stanem funkcjonalnym  $sf_0$  (kolumna  $sf_{1-9}$ ) w większości przypadków poprawia wartości wskaźników. Jednak porównanie tych wyników z analogicznymi wartościami uzyskanymi dla klasyfikatorów kontekstowych wyraźnie pokazuje korzyści stosowania tych drugich. Wśród rozwiązań optymalnych można znaleźć takie, które uzyskują lepsze wyniki zarówno dla stanu pełnej zdatności jak i dla stanu z uszkodzeniem z jednoczesnym zredukowaniem wartości

Tab. 5.12: Zestawienie wskaźników  $r_{fd}$  i  $r_{td}$  wyrażonych w procentach dla pojedynczego klasyfikatora drzewa decyzyjnego

	$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$f_0$	16.9 (2.6)	<b>13.6</b> (4.1)	20.2 (1.4)	18.0 (13.0)	22.0 (13.0)	10.0 (14.1)	2.0 (4.5)	20.0 (18.7)	12.0 (8.4)	8.0 (11.0)	10.0 (14.1)	20.0 (12.2)
$f_1$	60.0 (17.3)	77.8 (15.7)	42.2 (24.1)	60.0 (54.8)	80.0 (44.7)	60.0 (54.8)	20.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)
$f_2$	89.1 (7.7)	89.1 (13.6)	88.9 (7.9)	60.0 (54.8)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	100.0 (0.0)	60.0 (54.8)
$f_3$	88.9 (8.8)	93.3 (9.9)	84.4 (12.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)
$f_4$	90.0 (7.2)	88.9 (7.9)	91.1 (9.3)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	60.0 (54.8)
$f_5$	80.0 (3.0)	86.7 (18.3)	73.3 (14.9)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	80.0 (44.7)	60.0 (54.8)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)
$f_6$	80.0 (8.4)	88.9 (13.6)	71.1 (6.1)	80.0 (44.7)	80.0 (44.7)	60.0 (54.8)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)
$f_7$	16.7 (6.8)	11.1 (7.9)	22.2 (13.6)	20.0 (44.7)	20.0 (44.7)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	20.0 (44.7)	0.0 (0.0)	20.0 (44.7)	20.0 (44.7)
$f_8$	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)
$f_9$	75.8 (11.1)	76.9 (12.3)	74.7 (13.0)	60.0 (54.8)	80.0 (44.7)	40.0 (54.8)	80.0 (44.7)	80.0 (44.7)	80.0 (44.7)	100.0 (0.0)	60.0 (54.8)	100.0 (0.0)
$f_{10}$	63.3 (21.0)	75.6 (9.3)	51.1 (35.7)	40.0 (54.8)	100.0 (0.0)	40.0 (54.8)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	40.0 (54.8)	80.0 (44.7)
$f_{11}$	92.8 (6.6)	92.8 (6.6)	92.8 (6.6)	60.0 (54.8)	100.0 (0.0)	20.0 (44.7)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	20.0 (44.7)	100.0 (0.0)	100.0 (0.0)
$f_{1-11}$	75.9 (3.4)	<b>80.0</b> (3.9)	71.8 (6.5)	70.9 (7.6)	85.5 (5.0)	68.5 (8.6)	77.8 (4.7)	87.3 (8.1)	82.0 (18.3)	82.4 (11.3)	80.0 (7.6)	84.5 (12.3)

odchylenia standardowego w stosunku do pojedynczego klasyfikatora.

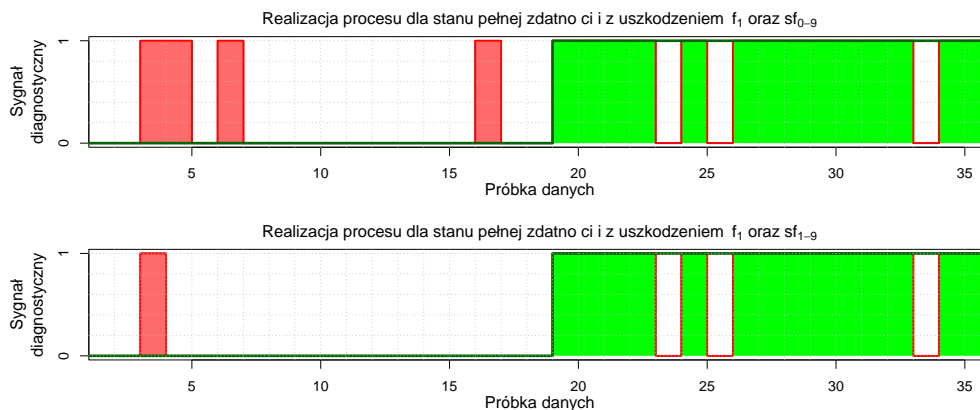
Na kolejnych wykresach pokazano wpływ nieustalonego stanu funkcjonalnego  $sf_0$  na wyniki uzyskiwane w ramach detekcji uszkodzenia podczas trwania procesu. W pierwszej kolejności zbiór danych walidacyjnych został połączony w podzbiory, w których pierwsza połowa zawiera dane dla stanu pełnej zdatności, a druga połowa dla stanu z określonym uszkodzeniem. Takie podejście pozwala na przedstawienie wyników w postaci symulowanego cyklicznego przebiegu roboczego stanowiska, gdzie pierwszy przebieg jest realizowany w warunkach pełnej zdatności stanowiska, a kolejny przebieg z udziałem wybranego uszkodzenia. Na osi poziomej wykresu pokazano kolejne próbki danych uzyskane w trakcie działania procesu. Próbkę te związane są z kolejnymi stanami funkcjonalnymi przeplatany nieustalonym stanem funkcjonalnym  $sf_0$ . Dokładny szereg stanów funkcjonalnych przypisanych do kolejnych próbek został pokazany w tabeli 5.13.

Na wykresie 5.14 porównano ze sobą dane testowe, składające się z próbek wygenerowanych dla eksperymentu zrealizowanego dla stanu pełnej zdatności i stanu z uszkodzeniem  $f_1$ . Ciemno zielona linia wyznacza rzeczywistą granicę pomiędzy stanem pełnej zdatności, a stanem z uszkodzeniem. Czerwone prostokąty wyznaczają próbki, dla których zarejestrowano fałszywe alarmy. Dla przykładu pokazanego na górnym wykresie

Tab. 5.13: Przyporządkowanie stanu funkcjonalnego do numeru próbki dla dalszych wykresów

Stan pełnej zdatności																	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$sf_0$	$sf_1$	$sf_2$	$sf_0$	$sf_3$	$sf_0$	$sf_4$	$sf_0$	$sf_5$	$sf_0$	$sf_6$	$sf_0$	$sf_7$	$sf_0$	$sf_8$	$sf_0$	$sf_9$	$sf_0$
Stan z uszkodzeniem																	
19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
$sf_0$	$sf_1$	$sf_2$	$sf_0$	$sf_3$	$sf_0$	$sf_4$	$sf_0$	$sf_5$	$sf_0$	$sf_6$	$sf_0$	$sf_7$	$sf_0$	$sf_8$	$sf_0$	$sf_9$	$sf_0$

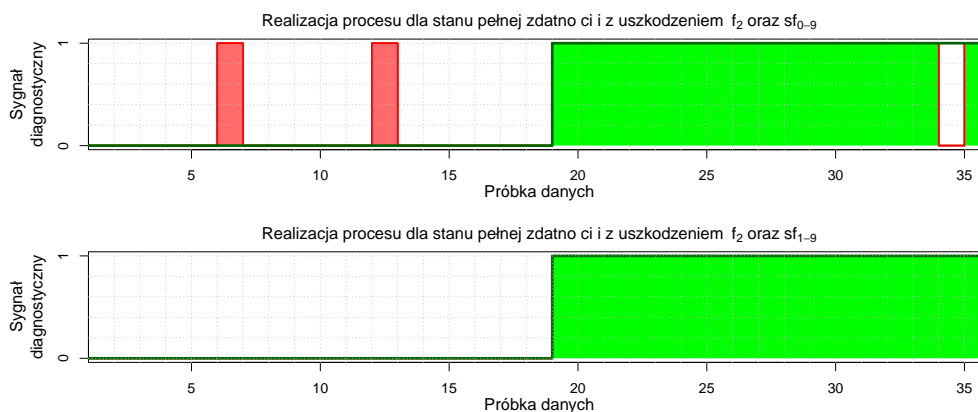
rysunku 5.14 są to próbki 3, 4, 6, 16, czyli w odniesieniu do tabeli 5.13 stany funkcjonalne  $sf_2, sf_0, sf_0, sf_0$ . Dalsza część wykresu, oznaczona zielonym kolorem, odzwierciedla liczbę próbek sklasyfikowanych jako uszkodzenie. Jeżeli dla jakiejś próbki w tym obszarze występuje przerwa, oznacza to brak wykrycia występującego tam uszkodzenia. Na przykładzie wykresu na rysunku 5.14, są to próbki 23, 25, 33, co w odniesieniu do tabeli 5.13, klasyfikator błędnie nie wykrył uszkodzenia w stanach funkcjonalnych  $sf_3, sf_4, sf_8$ . Dolny wykres na rysunku przedstawia te same dane, ale ignorowany jest nieustalony stan funkcjonalny  $sf_0$ . Dla zachowania skali i łatwości porównania obu wykresów, nie usunięto z dolnego wykresu próbek związanych ze stanem funkcjonalnym  $sf_0$ , a zastąpiono te wartości domyślnymi, czyli wartością 0 dla obszaru bez uszkodzenia i wartością 1, dla zakresu danych z uszkodzeniem.

Rys. 5.14: Porównanie wpływu stanu funkcjonalnego  $sf_0$  na liczbę generowanych fałszywych i prawidłowych alarmów

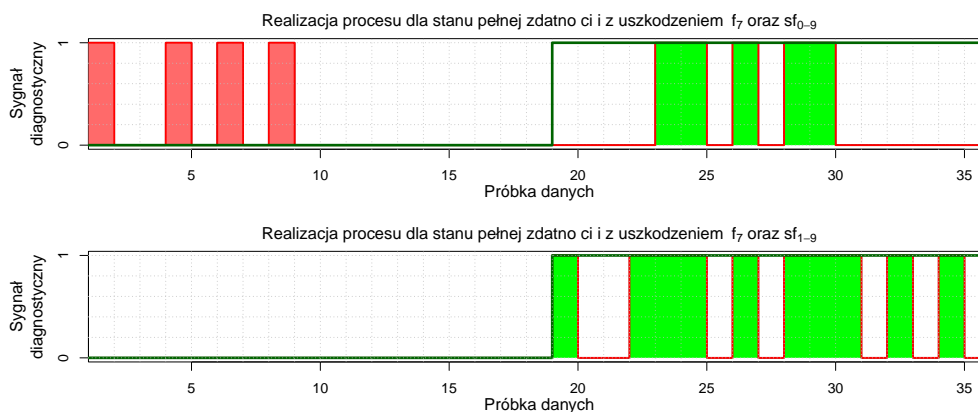
Dzięki takiej prezentacji przykładowych wyników, można wyraźnie zaobserwować redukcję liczby fałszywych alarmów w przypadku, gdy dane związane ze stanem funkcjonalnym  $sf_0$  są ignorowane lub usuwane ze zbioru.

W przypadku kolejnego wykresu (5.15), zestawiono ze sobą dane zebrane podczas prawidłowo działającego procesu przemysłowego oraz dane zarejestrowane w momencie wystąpienia uszkodzenia  $f_2$ . Porównując górny wykres z dolnym, wyraźnie widać, że usunięcie stanu funkcjonalnego  $sf_0$  pozwoliło na zupełne wyeliminowanie fałszywych alarmów oraz fałszywych stanów pełnej zdatności.

Innym interesującym przykładem jest uszkodzenie  $f_7$ , które według poprzednich analiz, jest najtrudniejszym uszkodzeniem do detekcji. Na rysunku 5.16 pokazano przykład



Rys. 5.15: Porównanie wpływu stanu funkcjonalnego  $sf_0$  na liczbę generowanych fałszywych i prawidłowych alarmów



Rys. 5.16: Porównanie wpływu stanu funkcjonalnego  $sf_0$  na liczbę generowanych fałszywych i prawidłowych alarmów

realizacji procesu detekcji dla takiego uszkodzenia. Na górnym wykresie można zaobserwować wyraźnie występujące fałszywe alarmy oraz nieliczne prawdziwe alarmy. Usunięcie ze zbioru danych testowych danych zarejestrowanych podczas występowania stanu funkcjonalnego  $sf_0$ , pozwoliło zupełnie wyeliminować fałszywe alarmy i częściowo zwiększyć liczbę prawidłowych alarmów, jednak wciąż dla części stanów funkcjonalnych klasyfikator kontekstowy nie był w stanie wykryć tego uszkodzenia.

Wykresy dla pozostały uszkodzeń zostały zamieszczone w rozdziale z dodatkami. Analizując kolejne wykresy można odnieść wrażenie, że usunięcie nieustalonego stanu funkcjonalnego  $sf_0$  poprawia wskaźniki  $r_{fd}$  i  $r_{td}$  i w większości przypadków takie założenie pokrywa się z rzeczywistością. Należy jednak pamiętać, że usunięcie z danych weryfikacyjnych próbek, stanu funkcjonalnego  $sf_0$  powoduje redukcję liczby próbek z 18 do 9. W skrajnym przypadku, gdy wszystkie fałszywe alarmy występują w stanach funkcjonalnych od  $sf_1$  do  $sf_9$ , usunięcie z danych próbek należących do  $sf_0$  pogarsza ostateczną wartość wskaźnika fałszywych alarmów. Jednak z poprzednich analiz wynika, że ze względu na charakter danych przypisanych do stanu funkcjonalnego  $sf_0$ , dane te są głównym czynnikiem wpływającym na pogorszenie skuteczności działania klasyfikatorów.

### Analiza złożoności klasyfikatorów

Klasyfikator bazujący na drzewie decyzyjnym charakteryzuje się dobrą interpretowalnością podobnie jak to jest w przypadku klasyfikatorów regułowych. Schemat drzewa pozwala na analizę kolejnych węzłów, gdzie każdy z węzłów będący warunkiem określa nazwę zmiennej oraz jej wartość progową. W zależności od uzyskanego wyniku aktywowany jest kolejny warunek aż do uzyskania wyniku zapisanego w ostatnim węźle (liściu). Duża złożoność takiego klasyfikatora znacząco utrudnia jego interpretację. W literaturze stosowane są różne sposoby obliczania złożoności struktury drzewa decyzyjnego. Stosuje się np. głębokość drzewa czy liczbę węzłów (Azad i Moshkov, 2015). W kolejnych tabelach porównano złożoności uzyskanych rozwiązań poprzez sumowanie liczby węzłów w drzewie, zarówno węzłów warunkowych jak i wynikowych. Większa liczba węzłów oznacza większą złożoność klasyfikatora.

Tab. 5.14: Porównanie poziomów złożoności uzyskanych klasyfikatorów drzewa decyzyjnego w zależności od zbioru danych treningowych  $t$

Dane treningowe	$Et_1$	$Et_2$	$Et_3$	$Et_4$	$Et_5$
Liczba połączeń	469	451	467	427	411

W tabeli 5.14 porównano złożoności podstawowych klasyfikatorów opartych na drzewie decyzyjnym. Uzyskano pięć klasyfikatorów, każdy z klasyfikatorów został wytrenowany z wykorzystaniem innego zbioru danych trenujących ( $Et_1 - Et_5$ ). Złożoność, wyrażona za pomocą liczby połączeń między węzłami osiąga wartości przekraczające 400, co zostało zobrazowane na rysunku 5.17 w dalszej części rozdziału. Wartości w tabeli 5.14 mogą posłużyć jako referencja dla złożoności uzyskanych w klasyfikatorach kontekstowych.

Tab. 5.15: Porównanie poziomów złożoności uzyskanych klasyfikatorów kontekstowych dla zbioru danych treningowych  $Et_1$  ze względu na rozwiązanie optymalne  $o$  z podziałem na stany funkcjonalne  $sf$

$O$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$	$sf_{1-9}$
$o_1$	351	33	15	39	5	9	17	17	29	33	165
$o_2$	411	35	27	35	5	7	31	7	3	33	151
$o_3$	401	33	15	39	23	17	17	21	29	31	195
$o_4$	363	49	15	31	19	21	31	19	21	29	207
$o_5$	411	33	15	31	5	17	5	7	17	29	131
$o_6$	355	33	23	39	21	23	17	7	7	31	171
$o_7$	351	9	29	1	5	7	17	21	23	33	113
$o_8$	457	3	39	35	5	17	31	15	9	27	155

W tabeli 5.15 przedstawiono wyniki złożoności dla klasyfikatorów kontekstowych, wytrenowanych za pomocą danych treningowych  $Et_1$ . Zarówno w tej tabeli, jak i w kolejnych opracowanych na bazie wyników uzyskanych z klasyfikatorów kontekstowych, w wierszach zawarto kolejne rozwiązania optymalne od  $o_1$  do  $o_8$ . W kolumnach natomiast

zawarto złożoności poszczególnych klasyfikatorów cząstkowych wchodzących w skład klasyfikatora kontekstowego, gdzie każdy z klasyfikatorów przypisany jest do konkretnego stanu funkcjonalnego  $sf_i$ . W ostatniej kolumnie zamieszczono sumę złożoności klasyfikatorów utworzonych w ramach stanów funkcjonalnych od  $sf_1$  do  $sf_9$ , czyli z pominięciem stanu nieustalonego  $sf_0$ . Analizując wyniki zamieszczone w tabeli 5.15, można zauważyć wysoką złożoność klasyfikatora utworzonego dla stanu funkcjonalnego  $sf_0$ . Poziom złożoności dla niektórych rozwiązań może osiągnąć wartości bliskie złożoności pojedynczego klasyfikatora (rys. 5.14). Analizując złożoności kolejnych klasyfikatorów utworzonych dla stanów funkcjonalnych od 1 do 9, można wyraźnie zaobserwować znaczącą redukcję liczby węzłów w drzewie decyzyjnym, co bezpośrednio przekłada się na znacznie mniejszy rozmiar drzewa. Suma wskaźników złożoności uzyskana dla tych stanów funkcjonalnych ( $sf_1 - sf_9$ ), przedstawiona w ostatniej kolumnie wciąż charakteryzuje się znacząco niższą wartością w porównaniu do pojedynczych klasyfikatorów.

Analizując klasyfikatory kontekstowe całościowo i sumując wskaźniki złożoności uzyskane dla wszystkich stanów funkcjonalnych w rozwiązaniu, można zaobserwować, że całkowita wartość złożoności znacząco przekracza wyniki uzyskiwane dla pojedynczych klasyfikatorów, co może sugerować znacznie większą złożoność klasyfikatorów kontekstowych w stosunku do klasyfikatorów podstawowych. Jednak patrząc na te wartości z perspektywy wyników przedstawionych w tabelach 5.8, 5.9, 5.10 i 5.11, wyraźnie widać, że zignorowanie wyników związanych ze stanem funkcjonalnym nieustalonym ( $sf_0$ ) wyraźnie poprawia wskaźniki fałszywych alarmów zachowując lub poprawiając wskaźniki prawidłowych alarmów. Wykluczając więc klasyfikator związany ze stanem funkcjonalnym  $sf_0$ , co wpływa na poprawę uzyskiwanych wyników detekcji uszkodzeń, uzyskuje się klasyfikator kontekstowy, realizujący proces detekcji tylko w stanach funkcjonalnych od  $sf_1$  do  $sf_9$ . W tym przedziale otrzymuje się klasyfikator parokrotnie mniej złożony od podstawowego klasyfikatora. Dodatkową zaletą jest możliwość interpretacji klasyfikatorów bazowych dla każdego stanu funkcjonalnego niezależnie.

Wskaźniki złożoności uzyskane dla pozostałych zbiorów treningowych  $Et_2, Et_3, Et_4, Et_5$  wykazują bardzo podobny charakter w zależności od rozpatrywanego rozwiązania optymalnego  $o_i$  oraz stanu funkcjonalnego  $sf_i$  w porównaniu do wyników przedstawionych w tabeli 5.15. Wynika to z faktu, że w ramach każdego rozwiązania znane są stałe wartości parametrów algorytmów trenujących klasyfikatory drzew decyzyjnych. A parametry te mają największy wpływ na końcowy kształt struktury drzewa. Tabele opisujące złożoność klasyfikatorów dla pozostałych zbiorów treningowych zostały zamieszczone w dodatku A w końcowej części pracy.

Klasyfikator drzewa decyzyjnego pozwala na łatwą wizualizację modelu klasyfikatora i jego wzrokową ocenę oraz interpretację. Na rysunku 5.17 pokazano strukturę całego drzewa decyzyjnego, utworzonego za pomocą pojedynczego klasyfikatora wytrenowanego za pomocą jednego z dostępnych zbiorów danych trenujących. Drzewo zostało przedstawione w postaci gałęzi promieniście rozchodzących się wokół pierwszego węzła drzewa. Taka forma prezentacji pozwoliła na zamieszczenie w pracy poglądowej struktury drzewa, pokazującej pełną jego złożoność.

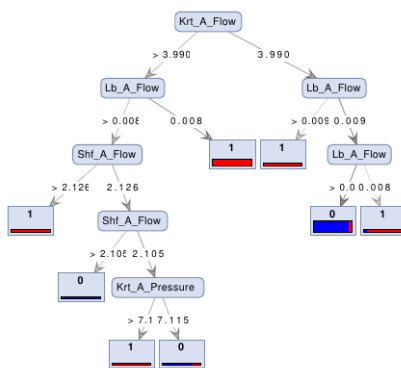


Rys. 5.17: Wizualizacja struktury drzewa decyzyjnego uzyskanego dla zbioru danych treningowych  $Et_1$

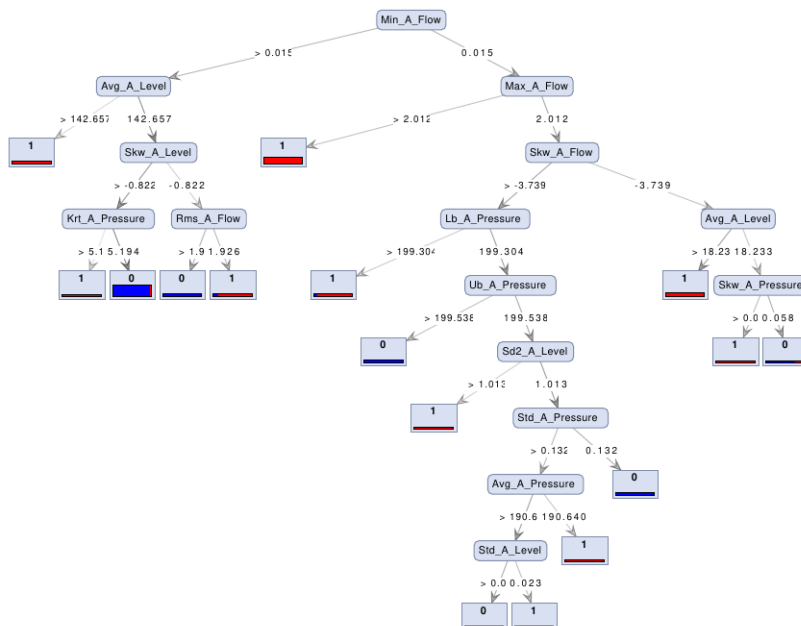
Na kolejnych rysunkach pokazano wybrane struktury klasyfikatorów bazowych, będących częścią klasyfikatora kontekstowego, gdzie każdy klasyfikator bazowy jest przypisany do konkretnego stanu funkcjonalnego. Na rysunku 5.18 pokazano strukturę drzewa decyzyjnego utworzonego dla stanu funkcjonalnego  $sf_2$ . Poziom złożoności drzewa pozwala na łatwą wzrokową interpretację klasyfikatora. Analizując zmienne uwzględnione w węzłach warunkowych można zaobserwować wysoki poziom skupienia klasyfikatora na cechach wyliczonych ze zmiennej procesowej, w której rejestrowany jest przepływ cieczy przy zastosowaniu pompy. Dodatkowo, w strukturze drzewa występuje węzeł decyzyjny, którego warunek bierze pod uwagę cechę obliczoną na podstawie ciśnienia powietrza zmierzonego w zbiorniku  $T3$ . Taki kierunek działania tego klasyfikatora wydaje się być uzasadniony, ponieważ w stanie funkcjonalnym  $sf_2$ , ciecz jest pompowana do zbiornika  $T3$ , który pełni funkcję akumulatora i jest wyposażony w czujnik ciśnienia powietrza.

Klasyfikator utworzony dla stanu funkcjonalnego  $sf_3$  (rys. 5.19) charakteryzuje się największą złożonością spośród wszystkich klasyfikatorów utworzonych dla stanów funkcjonalnych, od  $sf_1$  do  $sf_9$ , dla rozpatrywanego rozwiązania. W węzłach decyzyjnych uwzględniane są cechy obliczone ze wszystkich rejestrowanych zmiennych procesowych.

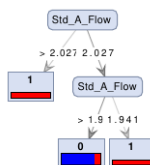




Rys. 5.18: Wizualizacja struktury drzewa decyzyjnego, utworzonego w ramach klasyfikatora kontekstowego dla stanu funkcjonalnego  $sf_2$



Rys. 5.19: Wizualizacja struktury drzewa decyzyjnego, utworzonego w ramach klasyfikatora kontekstowego dla stanu funkcjonalnego  $sf_3$



Rys. 5.20: Wizualizacja struktury drzewa decyzyjnego, utworzonego w ramach klasyfikatora kontekstowego dla stanu funkcjonalnego  $sf_4$

Na rysunku 5.20 przedstawiono strukturę drzewa decyzyjnego zbudowanego dla stanu funkcjonalnego  $sf_4$ . Ten przykład reprezentuje najmniejszą uzyskaną złożoność dla klasyfikatora. W strukturze drzewa znajdują się tylko 2 węzły decyzyjne, oba opierające się na tej samej ceście tej samej zmiennej procesowej, tj. odchylenie standardowe przepływu mierzonego za pompą. Zadanie realizowane w ramach stanu funkcjonalnego  $sf_4$  polega na pompowaniu wody do zbiornika  $T2$ . Interpretując schemat otrzymanego klasyfikatora z charakterem stanu funkcjonalnego, wybór przez klasyfikator cech bazujących na przepływie jest uzasadnione i bardzo dobrze oddaje realizowany proces. Rysunki zawierające struktury pozostałych drzew zostały umieszczone w dodatku A na końcu pracy.

## 5.6. Podsumowanie

W rozdziale przedstawiono wyniki praktycznego zastosowania opracowanej metody, stanowiące podstawę jej weryfikacji. Weryfikacja została przeprowadzona poprzez zrealizowanie eksperymentu z wykorzystaniem stanowiska laboratoryjnego FESTO S7 Edu-Trainer compact Siemens S7-300 CPU313C. Eksperyment obejmował swoim zakresem pojedynczy symulowany przebieg procesu przemysłowego. Pierwszy etap weryfikacji polegał na wielokrotnym powtórzeniu tego samego eksperymentu z uwzględnieniem zarówno stanu pełnej zdadności oraz stanów z uszkodzeniami, co pozwoliło na zebranie niezbędnych danych, wymaganych do realizacji właściwej części badań. Na podstawie zebranych danych obliczono zestaw cech liczbowych, które zostały wykorzystane do zasilenia algorytmów trenujących wybrane rodzaje klasyfikatorów z wykorzystaniem metody opisanej w rozdziale 4. Bazując na kontekście weryfikowano dwa podejścia doboru parametrów klasyfikatorów poprzez zastosowanie odpowiednich funkcji kryterialnych metody optymalizacji genetycznej NSGA-II. W obu przypadkach dążono do maksymalizacji wskaźników  $TPR$  i  $TNR$  wraz z zachowaniem wysokiej powtarzalności uzyskiwanych wyników poprzez minimalizację odchylenia standardowego. Zastosowanie algorytmu optymalizacji pozwoliło na automatyczny dobór parametrów trenowania klasyfikatorów, aby uzyskać klasyfikatory przystosowane do realizacji określonego celu. Uzyskane rozwiązania optymalne posłużyły w dalszej części rozdziału do przeprowadzenia procesu walidacji metody za pomocą odrębnych zbiorów danych. Dalsza analiza pozwoliła na porównanie uzyskanych wyników będących rezultatem działania klasyfikatora kontekstowego z pojedynczym klasyfikatorem. W celu opisanego powyższych wyników zastosowano miary powszechnie stosowane w diagnostyce procesów, czyli wskaźnik fałszywych alarmów  $r_{fd}$  i wskaźnik prawdziwych alarmów  $r_{td}$ . Uzyskane wyniki potwierdziły zasadność stosowania komitetów klasyfikatorów bazujących na kontekście.

## Rozdział 6

# Podsumowanie i wnioski

W pracy zrealizowano badania, których celem było opracowanie metody detekcji uszkodzeń. Charakterystyczną cechą zaproponowanego podejścia jest uwzględnienie kontekstu działania obiektu technicznego. W pracy zaproponowano metodę bazującą na zastosowaniu komitetu klasyfikatorów, gdzie każdy z klasyfikatorów związany jest z innym wariantem kontekstu. Pojęcie kontekstu, jako uogólnionej wiedzy występuje w obszarze diagnostyki technicznej, a w szczególności w zagadnieniach systemów ekspertowych, jednak często samo wykorzystanie i definicja kontekstu różni się od obszaru jego implementacji. W wyniku prowadzonych badań zaprezentowana została koncepcja bazująca na znanym i dyskretnym kontekście, który przyjmuje jeden z dostępnych wariantów z ograniczonego zbioru. Pozwala to na wybór odpowiednich danych dla określonego klasyfikatora. Kolejnym elementem badań, było rozszerzenie procesu trenowania klasyfikatorów w taki sposób, aby za pomocą algorytmu optymalizacji wielokryterialnej wskazać cechy relewantne oraz dostroić parametry algorytmów uczących klasyfikatory w celu uzyskania jak najkorzystniejszych rezultatów z punktu widzenia określonych miar weryfikacji sprawności detekcji uszkodzeń. W niniejszej pracy zaproponowano metodę detekcji uszkodzeń z uwzględnieniem kontekstu, która maksymalizuje skuteczność detekcji uszkodzeń z jednoczesną minimalizacją wskaźników fałszywych alarmów. Dodatkowym celem opracowanej metody, było dążenie zachowania wysokiej powtarzalności uzyskiwanych rezultatów. Zaproponowano oraz opisano sposób prezentacji uzyskanych wyników optymalnych oraz przedstawiono wyniki walidacji wybranych rozwiązań optymalnych.

Istotną zaletą opracowanej metody jest kontekstowy komitet klasyfikatorów, co wyraźnie wpływa na proces trenowania bazowych klasyfikatorów w komitecie, pozwalając na uczenie ich za pomocą znacznie mniejszego zbioru danych, ograniczonego kontekstem. Wpływa to pozytywnie na poziom skomplikowania klasyfikatora oraz sprawność w porównaniu do metody bazującej na pojedynczym klasyfikatorze.

Zaprezentowana metoda charakteryzuje się wysoką elastycznością, pozwalając zastosować inne typy klasyfikatorów lub określić nowe funkcje kryterialne dla algorytmu optymalizacji w celu ukierunkowania algorytmu trenowania klasyfikatorów na określony problem. Rozwiązanie to stanowi również bardzo dobrą podstawę pod kolejne metody bazujące na kontekście, który jak wynika z rezultatów pracy, jest bardzo istotnym elementem w diagnostyce procesów.

## 6.1. Wnioski końcowe

Do najistotniejszych wniosków końcowych, które sformułowano na podstawie wyników z przeprowadzonych badań, należą:

- Klasyfikator kontekstowy pozwala na uzyskanie lepszych wyników w zakresie wskaźników fałszywych i prawdziwych alarmów w porównaniu do podstawowej metody klasyfikacji z jednoczesną znaczną redukcją złożoności klasyfikatora.
- Ograniczenie ilości danych uczących klasyfikatora do zakresu związanego z danym kontekstem, pozwala na wyspecjalizowanie klasyfikatora dla konkretnego problemu, co pozytywnie wpływa na wnioskowanie w oparciu o konkretny fragment wiedzy.
- Zastosowanie algorytmu optymalizacji parametrów klasyfikatorów pozwala na znalezienie optymalnych wartości tych parametrów oraz wytrenowanie klasyfikatora, którego wskaźniki będą pokrywały się z założeniami zdefiniowanymi w funkcjach kryterialnych, tj. minimalizacja liczby fałszywych alarmów, maksymalizacja wskaźnika prawdziwych alarmów oraz zachowanie wysokiej powtarzalności wyników.
- Zastosowanie algorytmu optymalizacji wielokryterialnej daje możliwość wybrania takiego rozwiązania spośród wszystkich rozwiązań pareto optymalnych, które spełnia wymagania w zakresie detekcji stanu z uszkodzeniem w stosunku do detekcji stanu pełnej zdatności.

## 6.2. Plan przyszłych badań

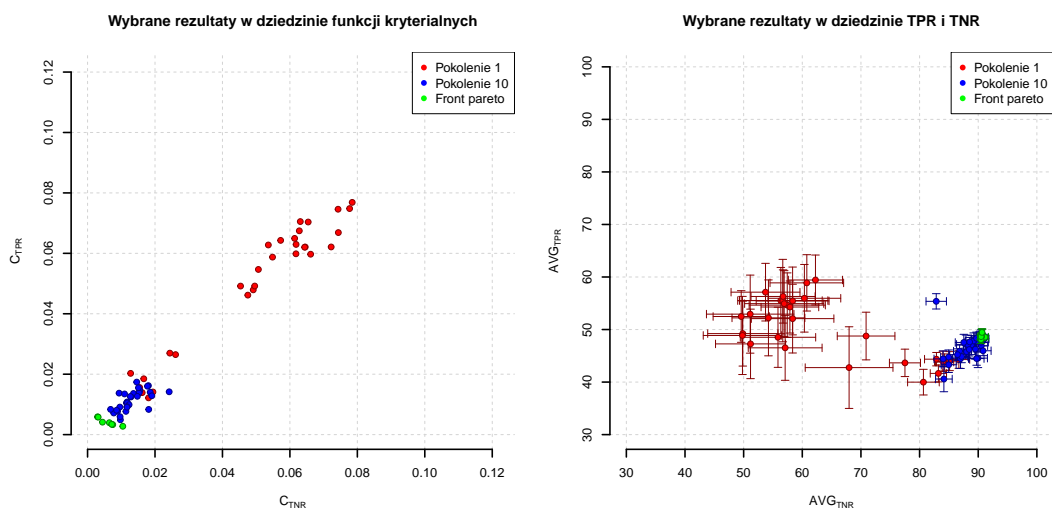
Opracowana metoda oraz przeprowadzone badania weryfikacyjne nie wyczerpują prezentowanej w pracy tematyki związanej z diagnostyką procesów przemysłowych z uwzględnieniem kontekstu. Stanowią natomiast podstawę do dalszych badań, które zdaniem autora powinny obejmować:

- możliwość wykorzystania opracowanej metody do izolacji uszkodzeń,
- opracowanie metod bazujących na innych podejściach tworzenia komitetów klasyfikatorów uwzględniających kontekst,
- uwzględnienie wielu kontekstów różnego typu oraz opracowanie metod odkrywania kontekstu ukrytego,
- zastosowanie innych metod uczenia maszynowego, jak np. klasyfikatory SVM, czy bazujące na sieciach neuronowych,
- zastosowanie różnego rodzaju klasyfikatorów dla kontekstów w ramach jednego komitetu,
- weryfikacja skuteczności działania opracowanej metody z wykorzystaniem danych pochodzących z rzeczywistych procesów przemysłowych.

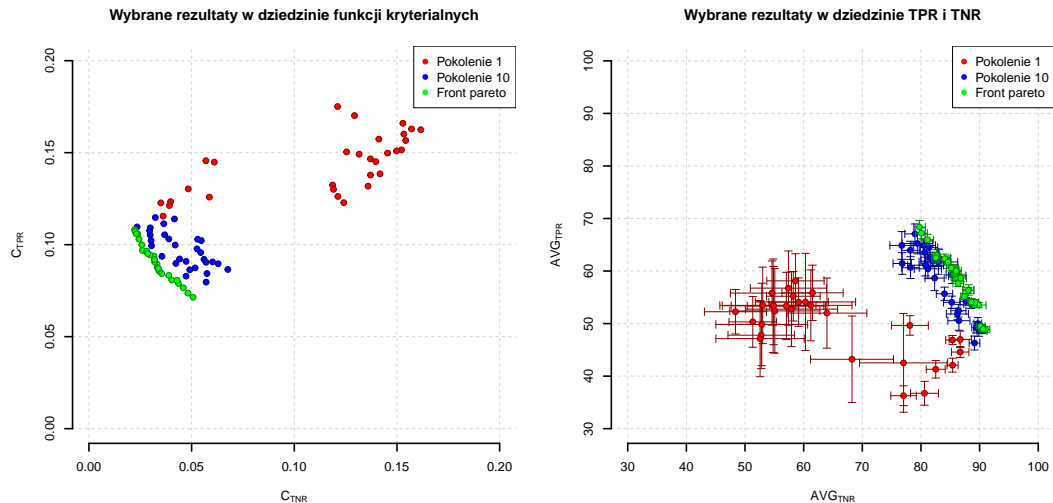
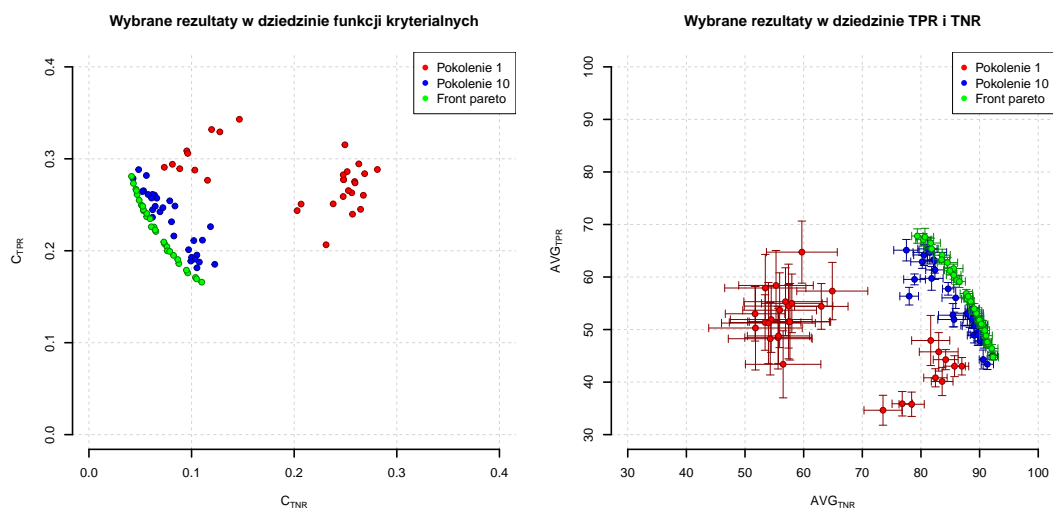
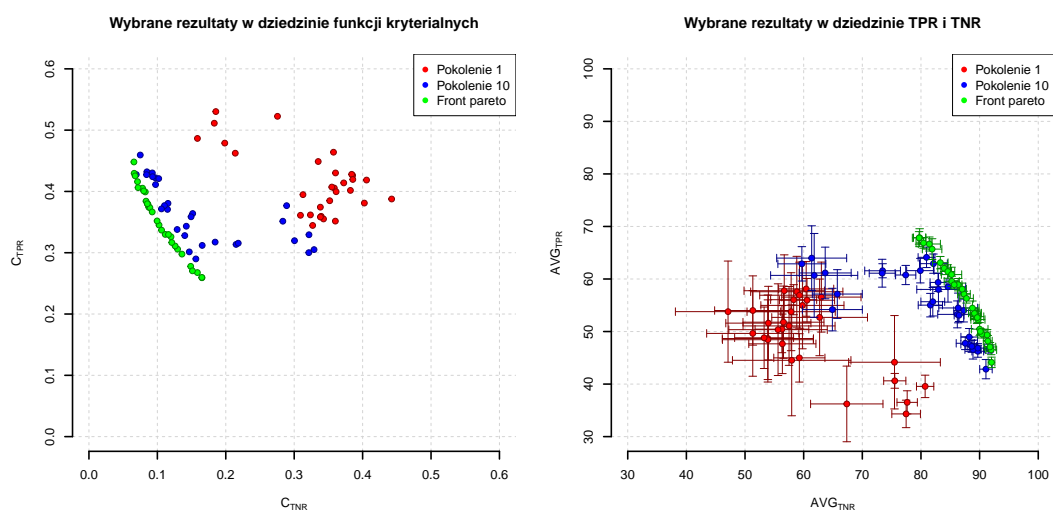
## Dodatek A

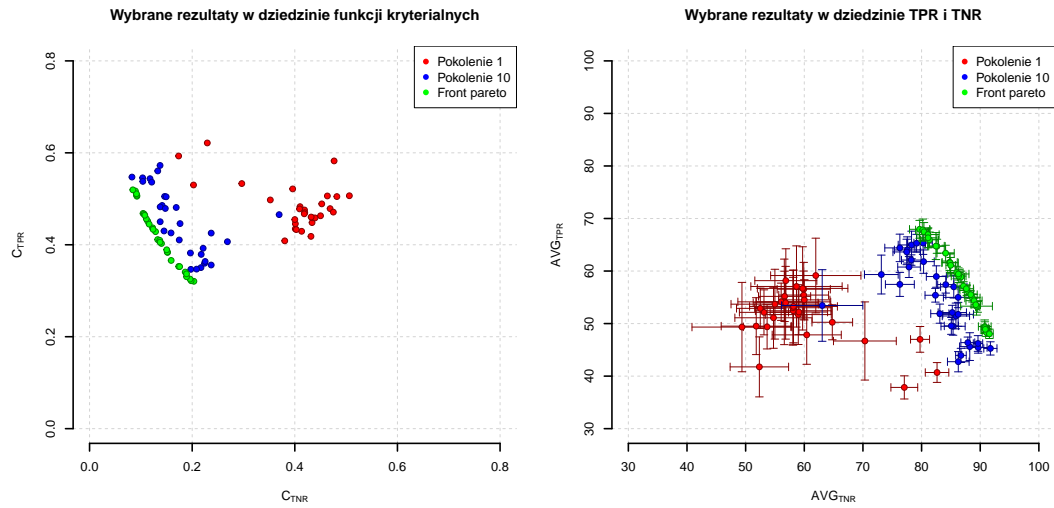
# Zbiorcze wyniki eksperymentu

Na wykresach od do zaprezentowano rozkład wybranych wyników, uzyskiwanych w rezultacie działania procesu optymalizacji parametrów treningowych klasyfikatora kontekstowego. Przedstawione wyniki dotyczą funkcji kryterialnych 4.9 i 4.10 opisanych w rozdziale 4.4.1. Rodzaj klasyfikatorów bazowych zastosowanych w klasyfikatorze kontekstowym to drzewo decyzyjne.



Rys. A.1: Wybrane zbiory rozwiązań dla parametru  $w = 0$

Rys. A.2: Wybrane zbiory rozwiązań dla parametru  $w = 0.2$ Rys. A.3: Wybrane zbiory rozwiązań dla parametru  $w = 0.5$ Rys. A.4: Wybrane zbiory rozwiązań dla parametru  $w = 0.8$



Rys. A.5: Wybrane zbiory rozwiązań dla parametru  $w = 1.0$

W kolejnych tabelach oraz na rysunkach przedstawiono wszystkie wyniki uzyskane w ramach eksperymentu na stanowisku laboratoryjnym. W tabelach od A.1 do A.12 porównano wskaźniki fałszywych alarmów dla stanu pełnej zdatności (Tab. A.1) oraz wskaźniki prawidłowych alarmów dla stanu z uszkodzeniem z rozróżnieniem poszczególnych uszkodzeń (Tab. A.2 i kolejne). Wyniki dotyczą przypadku, w którym klasyfikatory utworzone dla każdego rozwiązania optymalnego był wytrenowany i przetestowany na dokładnie tym samym zbiorze danych.

Tab. A.1: Porównanie wskaźników fałszywych alarmów wyrażonych w procentach ze względu na stany funkcjonalne  $sf_i$  dla poszczególnych rozwiązań optymalnych  $o_i$ .

<b>O</b>	$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$o_1$	2.7	2.5	2.8	2.8	0.9	0.9	1.8	0.9	10.0	1.8	1.8	1.8
$o_2$	2.7	1.4	3.9	2.8	1.8	0.0	1.8	0.9	0.9	1.8	2.7	0.0
$o_3$	2.3	2.4	2.2	2.8	0.9	0.9	0.9	2.7	10.0	0.0	1.8	1.8
$o_4$	4.6	6.5	2.6	43.5	0.9	2.7	1.8	1.8	2.7	1.8	0.0	3.7
$o_5$	2.6	3.2	1.9	0.9	0.9	2.7	0.9	3.6	10.0	1.8	4.5	3.7
$o_6$	4.1	5.3	2.9	2.8	0.9	3.6	1.8	1.8	10.0	24.5	0.0	1.8
$o_7$	5.1	7.3	2.8	43.5	0.9	0.0	0.9	2.7	10.0	0.0	5.5	2.8
$o_8$	4.9	5.1	4.7	10.2	2.7	0.0	2.7	1.8	0.9	0.9	22.7	3.7

Tab. A.2: Porównanie wskaźników prawidłowych alarmów wyrażonych w procentach dla uszkodzenia  $f_1$  ze względu na stany funkcjonalne dla poszczególnych rozwiązań optymalnych  $o_i$ .

<b>O</b>	$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$o_1$	90.0	84.4	95.6	100.0	90.0	100.0	0.0	100.0	90.0	100.0	90.0	90.0
$o_2$	80.0	76.7	83.3	100.0	100.0	90.0	20.0	100.0	90.0	100.0	0.0	90.0
$o_3$	90.6	93.3	87.8	100.0	90.0	100.0	80.0	100.0	90.0	100.0	90.0	90.0
$o_4$	96.1	96.7	95.6	90.0	100.0	100.0	100.0	100.0	100.0	100.0	90.0	90.0
$o_5$	83.9	77.8	90.0	40.0	100.0	100.0	0.0	100.0	90.0	90.0	90.0	90.0
$o_6$	91.1	86.7	95.6	100.0	100.0	100.0	100.0	100.0	90.0	100.0	0.0	90.0
$o_7$	84.4	73.3	95.6	90.0	100.0	0.0	0.0	100.0	90.0	100.0	100.0	80.0
$o_8$	90.0	86.7	93.3	10.0	100.0	90.0	100.0	100.0	90.0	90.0	100.0	100.0

Tab. A.3: Porównanie wskaźników prawidłowych alarmów wyrażonych w procentach dla uszkodzenia  $f_2$  ze względu na stany funkcjonalne dla poszczególnych rozwiązań optymalnych  $o_i$ .

<b>O</b>	$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$o_1$	98.9	98.9	98.9	100.0	90.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
$o_2$	96.6	98.9	94.3	100.0	100.0	100.0	100.0	100.0	90.9	100.0	100.0	100.0
$o_3$	98.3	97.8	98.9	100.0	90.0	100.0	100.0	100.0	100.0	90.9	100.0	100.0
$o_4$	98.9	97.8	100.0	100.0	90.0	100.0	100.0	100.0	90.9	100.0	100.0	100.0
$o_5$	98.3	98.9	97.7	100.0	90.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
$o_6$	99.4	100.0	98.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
$o_7$	92.7	86.5	98.9	100.0	100.0	0.0	100.0	100.0	100.0	90.9	100.0	85.7
$o_8$	97.2	96.6	97.7	90.0	90.0	100.0	100.0	100.0	90.9	100.0	100.0	100.0



Tab. A.4: Porównanie wskaźników prawidłowych alarmów wyrażonych w procentach dla uszkodzenia  $f_3$  ze względu na stany funkcjonalne dla poszczególnych rozwiązań optymalnych  $o_i$ .

<b>O</b>	$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$o_1$	96.1	94.4	97.8	100.0	80.0	100.0	100.0	80.0	100.0	100.0	90.0	100.0
$o_2$	97.8	96.7	98.9	100.0	90.0	100.0	100.0	80.0	100.0	100.0	100.0	100.0
$o_3$	96.7	95.6	97.8	100.0	90.0	100.0	100.0	80.0	100.0	100.0	90.0	100.0
$o_4$	96.1	94.4	97.8	100.0	80.0	100.0	100.0	80.0	100.0	100.0	90.0	100.0
$o_5$	96.7	95.6	97.8	100.0	80.0	100.0	100.0	90.0	100.0	90.0	100.0	100.0
$o_6$	97.8	96.7	98.9	100.0	100.0	100.0	100.0	80.0	100.0	100.0	90.0	100.0
$o_7$	91.1	84.4	97.8	100.0	80.0	0.0	100.0	80.0	100.0	100.0	100.0	100.0
$o_8$	96.7	98.9	94.4	100.0	90.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0

Tab. A.5: Porównanie wskaźników prawidłowych alarmów wyrażonych w procentach dla uszkodzenia  $f_4$  ze względu na stany funkcjonalne dla poszczególnych rozwiązań optymalnych  $o_i$ .

<b>O</b>	$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$o_1$	98.9	100.0	97.8	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
$o_2$	92.8	91.1	94.4	100.0	100.0	100.0	100.0	100.0	100.0	100.0	30.0	90.0
$o_3$	97.2	96.7	97.8	100.0	100.0	100.0	100.0	100.0	80.0	100.0	100.0	90.0
$o_4$	98.9	100.0	97.8	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
$o_5$	97.8	97.8	97.8	100.0	100.0	100.0	100.0	100.0	80.0	100.0	100.0	100.0
$o_6$	95.6	92.2	98.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0	30.0	100.0
$o_7$	92.2	86.7	97.8	100.0	100.0	0.0	100.0	100.0	80.0	100.0	100.0	100.0
$o_8$	96.7	100.0	93.3	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0

Tab. A.6: Porównanie wskaźników prawidłowych alarmów wyrażonych w procentach dla uszkodzenia  $f_5$  ze względu na stany funkcjonalne dla poszczególnych rozwiązań optymalnych  $o_i$ .

<b>O</b>	$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$o_1$	98.9	98.9	98.9	100.0	100.0	100.0	100.0	100.0	90.9	100.0	100.0	100.0
$o_2$	95.6	95.6	95.6	100.0	100.0	100.0	100.0	100.0	90.9	100.0	70.0	100.0
$o_3$	98.4	97.8	98.9	100.0	100.0	100.0	100.0	100.0	90.9	90.0	100.0	100.0
$o_4$	98.4	97.8	98.9	100.0	100.0	100.0	100.0	100.0	90.9	100.0	90.0	100.0
$o_5$	97.8	97.8	97.8	100.0	100.0	100.0	100.0	100.0	90.9	100.0	90.0	100.0
$o_6$	97.3	95.6	98.9	100.0	100.0	100.0	100.0	100.0	90.9	100.0	70.0	100.0
$o_7$	92.3	85.7	98.9	100.0	100.0	0.0	100.0	100.0	90.9	90.0	100.0	90.0
$o_8$	95.1	96.7	93.4	90.0	100.0	100.0	100.0	90.0	90.9	100.0	100.0	100.0



Tab. A.10: Porównanie wskaźników prawidłowych alarmów wyrażonych w procentach dla uszkodzenia  $f_9$  ze względu na stany funkcjonalne dla poszczególnych rozwiązań optymalnych  $o_i$ .

<b>O</b>	$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$o_1$	96.0	94.3	97.7	100.0	90.0	90.0	100.0	100.0	80.0	90.0	100.0	100.0
$o_2$	94.8	93.1	96.6	100.0	100.0	90.0	100.0	100.0	80.0	90.0	80.0	100.0
$o_3$	97.1	96.6	97.7	100.0	100.0	90.0	100.0	100.0	80.0	100.0	100.0	100.0
$o_4$	97.1	96.6	97.7	100.0	90.0	90.0	100.0	100.0	100.0	90.0	100.0	100.0
$o_5$	93.7	89.7	97.7	50.0	90.0	90.0	100.0	100.0	80.0	100.0	100.0	100.0
$o_6$	96.0	94.3	97.7	100.0	90.0	100.0	100.0	100.0	80.0	100.0	80.0	100.0
$o_7$	92.0	86.2	97.7	100.0	100.0	0.0	100.0	100.0	80.0	100.0	100.0	100.0
$o_8$	89.7	85.1	94.3	0.0	100.0	90.0	100.0	100.0	80.0	100.0	100.0	100.0

Tab. A.11: Porównanie wskaźników prawidłowych alarmów wyrażonych w procentach dla uszkodzenia  $f_{10}$  ze względu na stany funkcjonalne dla poszczególnych rozwiązań optymalnych  $o_i$ .

<b>O</b>	$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$o_1$	94.5	94.5	94.4	90.0	100.0	100.0	90.0	90.0	90.0	100.0	100.0	90.0
$o_2$	82.9	85.7	80.0	90.0	100.0	100.0	90.0	90.0	90.0	100.0	10.0	100.0
$o_3$	92.8	93.4	92.2	90.0	100.0	100.0	100.0	100.0	60.0	90.0	100.0	100.0
$o_4$	95.6	96.7	94.4	100.0	100.0	100.0	100.0	100.0	100.0	90.0	90.0	90.0
$o_5$	89.5	86.8	92.2	40.0	100.0	100.0	100.0	100.0	60.0	100.0	90.0	90.0
$o_6$	91.2	86.8	95.6	90.0	100.0	100.0	100.0	100.0	90.0	100.0	10.0	90.0
$o_7$	88.4	82.4	94.4	100.0	100.0	0.0	100.0	90.0	60.0	90.0	100.0	100.0
$o_8$	87.8	87.9	87.8	10.0	100.0	100.0	100.0	100.0	90.0	100.0	100.0	90.0

Tab. A.12: Porównanie wskaźników prawidłowych alarmów wyrażonych w procentach dla uszkodzenia  $f_{11}$  ze względu na stany funkcjonalne dla poszczególnych rozwiązań optymalnych  $o_i$ .

<b>O</b>	$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$o_1$	96.9	95.1	98.8	100.0	100.0	66.7	100.0	90.0	80.0	100.0	100.0	100.0
$o_2$	90.7	91.4	90.1	100.0	100.0	100.0	100.0	90.0	80.0	100.0	60.0	100.0
$o_3$	96.9	96.3	97.5	100.0	100.0	100.0	100.0	90.0	80.0	100.0	100.0	100.0
$o_4$	98.8	98.8	98.8	100.0	100.0	100.0	100.0	90.0	100.0	100.0	100.0	100.0
$o_5$	93.2	88.9	97.5	50.0	100.0	100.0	70.0	100.0	80.0	100.0	100.0	100.0
$o_6$	95.1	91.4	98.8	100.0	100.0	100.0	100.0	90.0	80.0	100.0	60.0	100.0
$o_7$	93.8	88.9	98.8	100.0	100.0	0.0	70.0	90.0	80.0	100.0	100.0	100.0
$o_8$	88.3	88.9	87.7	37.5	90.0	100.0	100.0	100.0	80.0	100.0	90.0	100.0

W kolejnych tabelach zamieszczono wyniki walidacji uzyskane dla kolejnych rozwiązań optymalnych  $O$ . Walidacja polegała na wyodrębnieniu z całego zbioru danych fragmentu, który służył do trenowania klasyfikatora i znacznie mniejszego niezależnego zbioru walidacyjnego, który służył do wyznaczenia wskaźników prawidłowych i fałszywych alarmów. Przygotowano 5 takich zbiorów z losowo wybranymi eksperymentami dla każdego z podzbiorów. Każda kolejna tabela od A.13 do A.20 dotyczy innego rozwiązania optymalnego oraz pojedynczego klasyfikatora A.21, a wyniki w tabeli zostały podzielone ze względu na stan funkcjonalny  $sf_i$  i uszkodzenie  $f_i$ . W każdej tabeli wartości przedstawione w komórkach obrazują wartość średnią wskaźnika dla konkretnego uszkodzenia i stanu funkcjonalnego (górną wartość) i odchylenie standardowe (dolną wartość) obliczone dla każdego z walidacyjnych zbiorów danych.

Tab. A.13: Zestawienie wskaźników  $r_{fd}$  i  $r_{td}$  dla rozwiązania optymalnego  $o_1$ .

	$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$f_0$	19.1 (3.9)	8.4 (2.6)	29.8 (5.4)	14.0 (8.9)	8.0 (8.4)	22.0 (14.8)	2.0 (4.5)	0.0 (0.0)	10.0 (10.0)	8.0 (8.4)	2.0 (4.5)	10.0 (10.0)
$f_1$	71.1 (17.3)	75.6 (12.2)	66.7 (26.1)	100.0 (0.0)	100.0 (0.0)	20.0 (44.7)	0.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)
$f_2$	90.2 (9.6)	95.8 (5.8)	84.4 (14.9)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)
$f_3$	86.7 (6.3)	91.1 (9.3)	82.2 (9.9)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	60.0 (54.8)	100.0 (0.0)
$f_4$	85.6 (8.4)	88.9 (7.9)	82.2 (14.9)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	60.0 (54.8)	80.0 (44.7)
$f_5$	84.4 (7.2)	86.7 (18.3)	82.2 (16.9)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	60.0 (54.8)
$f_6$	83.3 (8.8)	91.1 (12.2)	75.6 (9.3)	60.0 (54.8)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)
$f_7$	26.7 (8.2)	31.1 (12.2)	22.2 (7.9)	40.0 (54.8)	60.0 (54.8)	20.0 (44.7)	20.0 (44.7)	20.0 (44.7)	20.0 (44.7)	0.0 (0.0)	60.0 (54.8)	40.0 (54.8)
$f_8$	95.6 (4.6)	91.1 (9.3)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	40.0 (54.8)	100.0 (0.0)
$f_9$	65.7 (9.8)	79.7 (9.0)	51.7 (26.0)	40.0 (54.8)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	60.0 (54.8)	60.0 (54.8)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)
$f_{10}$	72.2 (14.2)	84.4 (6.1)	60.0 (29.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	40.0 (54.8)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)
$f_{11}$	81.9 (10.7)	85.8 (14.9)	77.7 (10.0)	60.0 (54.8)	100.0 (0.0)	20.0 (44.7)	80.0 (44.7)	80.0 (44.7)	80.0 (44.7)	20.0 (44.7)	100.0 (0.0)	80.0 (44.7)
$f_{1-11}$	76.6 (2.7)	81.9 (1.6)	71.4 (5.6)	78.2 (10.4)	94.5 (5.0)	76.4 (5.9)	77.6 (5.8)	78.2 (18.9)	78.5 (13.9)	86.4 (9.2)	81.8 (9.1)	84.7 (8.9)

Tab. A.14: Zestawienie wskaźników  $r_{fd}$  i  $r_{td}$  dla rozwiązania optymalnego  $o_2$ .

	$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$f_0$	15.3 (3.2)	6.7 (1.1)	24.0 (6.9)	6.0 (8.9)	12.0 (8.4)	8.0 (11.0)	4.0 (8.9)	0.0 (0.0)	10.0 (10.0)	4.0 (8.9)	0.0 (0.0)	16.0 (5.5)
$f_1$	62.2 (22.7)	62.2 (16.9)	62.2 (33.9)	60.0 (54.8)	100.0 (0.0)	20.0 (44.7)	20.0 (44.7)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	0.0 (0.0)	80.0 (44.7)
$f_2$	95.6 (2.5)	97.8 (5.0)	93.3 (6.1)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)
$f_3$	94.4 (5.6)	95.6 (9.9)	93.3 (9.9)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)
$f_4$	87.8 (9.9)	88.9 (11.1)	86.7 (9.3)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	40.0 (54.8)	60.0 (54.8)
$f_5$	78.9 (14.9)	77.8 (28.3)	80.0 (9.3)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	80.0 (44.7)	80.0 (44.7)	40.0 (54.8)	60.0 (54.8)
$f_6$	80.0 (6.3)	91.1 (9.3)	68.9 (9.3)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)
$f_7$	22.2 (5.6)	20.0 (12.2)	24.4 (9.3)	20.0 (44.7)	60.0 (54.8)	20.0 (44.7)	20.0 (44.7)	0.0 (0.0)	40.0 (54.8)	0.0 (0.0)	0.0 (0.0)	20.0 (44.7)
$f_8$	95.6 (4.6)	91.1 (9.3)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	40.0 (54.8)	100.0 (0.0)
$f_9$	68.5 (8.5)	75.3 (14.2)	61.7 (21.2)	60.0 (54.8)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	60.0 (54.8)	40.0 (54.8)	80.0 (44.7)	40.0 (54.8)	100.0 (0.0)
$f_{10}$	70.0 (11.5)	71.1 (9.9)	68.9 (14.5)	40.0 (54.8)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	40.0 (54.8)	100.0 (0.0)	20.0 (44.7)	80.0 (44.7)
$f_{11}$	75.5 (9.2)	85.8 (12.0)	65.2 (20.5)	60.0 (54.8)	100.0 (0.0)	20.0 (44.7)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	20.0 (44.7)	60.0 (54.8)	100.0 (0.0)
$f_{1-11}$	75.5 (4.7)	77.8 (5.4)	73.2 (4.9)	72.7 (17.0)	96.4 (5.0)	80.2 (7.4)	79.5 (8.3)	78.2 (15.2)	80.3 (10.1)	82.4 (11.3)	47.3 (25.2)	83.2 (11.7)

Tab. A.15: Zestawienie wskaźników  $r_{fd}$  i  $r_{td}$  dla rozwiązania optymalnego  $o_3$ .

	$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$f_0$	19.8 (2.7)	12.4 (4.4)	27.1 (2.8)	18.0 (8.4)	18.0 (14.8)	24.0 (18.2)	4.0 (5.5)	8.0 (8.4)	14.0 (8.9)	8.0 (8.4)	2.0 (4.5)	16.0 (5.5)
$f_1$	74.4 (16.0)	84.4 (12.7)	64.4 (25.3)	80.0 (44.7)	100.0 (0.0)	20.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)
$f_2$	92.4 (6.0)	98.0 (4.5)	86.7 (9.3)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)
$f_3$	88.9 (5.6)	91.1 (9.3)	86.7 (9.3)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	60.0 (54.8)	100.0 (0.0)
$f_4$	85.6 (10.8)	88.9 (7.9)	82.2 (14.9)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	60.0 (54.8)	60.0 (54.8)
$f_5$	86.7 (3.0)	88.9 (13.6)	84.4 (12.7)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	60.0 (54.8)
$f_6$	82.2 (4.6)	95.6 (6.1)	68.9 (9.3)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)
$f_7$	35.6 (15.0)	40.0 (14.9)	31.1 (21.4)	20.0 (44.7)	60.0 (54.8)	20.0 (44.7)	60.0 (54.8)	40.0 (54.8)	60.0 (54.8)	20.0 (44.7)	60.0 (54.8)	20.0 (44.7)
$f_8$	96.7 (3.0)	93.3 (6.1)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	40.0 (54.8)	100.0 (0.0)
$f_9$	75.0 (9.2)	84.4 (12.7)	65.6 (14.9)	40.0 (54.8)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	100.0 (0.0)	60.0 (54.8)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)
$f_{10}$	71.1 (16.4)	82.2 (6.1)	60.0 (32.0)	60.0 (54.8)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	60.0 (54.8)	60.0 (54.8)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)
$f_{11}$	83.1 (11.6)	90.3 (10.4)	75.8 (14.6)	60.0 (54.8)	100.0 (0.0)	20.0 (44.7)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	20.0 (44.7)	100.0 (0.0)	100.0 (0.0)
$f_{1-11}$	79.2 (4.5)	85.1 (2.0)	73.2 (8.3)	74.5 (17.5)	94.5 (5.0)	80.4 (7.1)	96.2 (5.2)	85.5 (10.4)	83.9 (13.4)	86.4 (9.2)	81.8 (9.1)	83.2 (11.7)

Tab. A.16: Zestawienie wskaźników  $r_{fd}$  i  $r_{td}$  dla rozwiązania optymalnego  $o_4$ .

	$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$f_0$	19.9 (2.5)	11.3 (2.9)	28.4 (4.2)	32.0 (14.8)	12.0 (8.4)	8.0 (11.0)	10.0 (7.1)	4.0 (5.5)	14.0 (13.4)	6.0 (5.5)	4.0 (5.5)	12.0 (8.4)
$f_1$	70.0 (19.5)	73.3 (16.9)	66.7 (26.1)	40.0 (54.8)	100.0 (0.0)	20.0 (44.7)	40.0 (54.8)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)
$f_2$	90.2 (9.6)	95.8 (5.8)	84.4 (14.9)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)
$f_3$	85.6 (6.3)	91.1 (9.3)	80.0 (12.2)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	60.0 (54.8)	100.0 (0.0)
$f_4$	85.6 (6.3)	88.9 (11.1)	82.2 (14.9)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	60.0 (54.8)	60.0 (54.8)
$f_5$	83.3 (6.8)	84.4 (18.6)	82.2 (16.9)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	60.0 (54.8)
$f_6$	83.3 (7.9)	93.3 (9.9)	73.3 (9.9)	60.0 (54.8)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)
$f_7$	30.0 (6.3)	37.8 (9.9)	22.2 (7.9)	60.0 (54.8)	60.0 (54.8)	20.0 (44.7)	40.0 (54.8)	20.0 (44.7)	60.0 (54.8)	20.0 (44.7)	20.0 (44.7)	40.0 (54.8)
$f_8$	95.6 (4.6)	91.1 (9.3)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	40.0 (54.8)	100.0 (0.0)
$f_9$	67.8 (13.3)	86.1 (10.0)	49.4 (24.9)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	60.0 (54.8)	100.0 (0.0)	60.0 (54.8)	100.0 (0.0)
$f_{10}$	70.0 (18.7)	84.4 (9.9)	55.6 (29.4)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	60.0 (54.8)	60.0 (54.8)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)
$f_{11}$	85.6 (9.1)	93.3 (14.9)	77.7 (10.0)	100.0 (0.0)	100.0 (0.0)	20.0 (44.7)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	20.0 (44.7)	80.0 (44.7)	80.0 (44.7)
$f_{1-11}$	76.9 (2.6)	83.5 (2.5)	70.3 (5.7)	81.8 (17.0)	94.5 (5.0)	80.2 (7.4)	86.9 (10.6)	80.0 (16.3)	85.5 (10.4)	86.4 (9.2)	72.7 (17.0)	82.9 (8.2)

Tab. A.17: Zestawienie wskaźników  $r_{fd}$  i  $r_{td}$  dla rozwiązania optymalnego  $o_5$ .

	$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$f_0$	17.6 (1.7)	8.0 (1.8)	27.1 (3.4)	12.0 (8.4)	12.0 (8.4)	8.0 (11.0)	6.0 (13.4)	0.0 (0.0)	14.0 (8.9)	8.0 (8.4)	0.0 (0.0)	12.0 (8.4)
$f_1$	68.9 (16.9)	75.6 (14.5)	62.2 (29.0)	60.0 (54.8)	100.0 (0.0)	20.0 (44.7)	20.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)
$f_2$	90.2 (4.3)	95.8 (5.8)	84.4 (9.9)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)
$f_3$	92.2 (5.0)	97.8 (5.0)	86.7 (9.3)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)
$f_4$	86.7 (10.8)	93.3 (9.9)	80.0 (14.5)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)
$f_5$	84.4 (4.6)	84.4 (18.6)	84.4 (12.7)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	60.0 (54.8)
$f_6$	80.0 (6.3)	93.3 (9.9)	66.7 (7.9)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)
$f_7$	32.2 (15.4)	33.3 (20.8)	31.1 (21.4)	0.0 (0.0)	60.0 (54.8)	20.0 (44.7)	40.0 (54.8)	20.0 (44.7)	60.0 (54.8)	20.0 (44.7)	40.0 (54.8)	40.0 (54.8)
$f_8$	95.6 (4.6)	91.1 (9.3)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	40.0 (54.8)	100.0 (0.0)
$f_9$	73.9 (10.7)	82.2 (14.9)	65.6 (14.9)	40.0 (54.8)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	60.0 (54.8)	100.0 (0.0)	60.0 (54.8)	100.0 (0.0)
$f_{10}$	71.1 (15.4)	84.4 (9.9)	57.8 (34.6)	60.0 (54.8)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	60.0 (54.8)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)
$f_{11}$	80.8 (12.3)	85.8 (12.0)	75.8 (14.6)	40.0 (54.8)	100.0 (0.0)	20.0 (44.7)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	20.0 (44.7)	100.0 (0.0)	80.0 (44.7)
$f_{1-11}$	77.7 (3.9)	83.3 (2.8)	72.2 (8.7)	69.1 (15.2)	94.5 (5.0)	80.2 (7.4)	81.3 (9.8)	85.5 (12.2)	83.9 (13.4)	88.4 (4.7)	81.8 (12.9)	84.7 (8.9)



Tab. A.18: Zestawienie wskaźników  $r_{fd}$  i  $r_{td}$  dla rozwiązania optymalnego  $o_6$ .

	$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$f_0$	20.9 (4.6)	12.2 (4.0)	29.6 (6.1)	18.0 (8.4)	14.0 (11.4)	22.0 (16.4)	12.0 (8.4)	0.0 (0.0)	10.0 (10.0)	24.0 (11.4)	0.0 (0.0)	10.0 (7.1)
$f_1$	67.8 (20.6)	73.3 (12.7)	62.2 (30.0)	100.0 (0.0)	100.0 (0.0)	20.0 (44.7)	40.0 (54.8)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	20.0 (44.7)	80.0 (44.7)
$f_2$	88.0 (6.8)	93.6 (5.9)	82.2 (12.7)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)
$f_3$	88.9 (3.9)	91.1 (9.3)	86.7 (5.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	60.0 (54.8)	100.0 (0.0)
$f_4$	84.4 (10.7)	86.7 (12.2)	82.2 (14.9)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	20.0 (44.7)	80.0 (44.7)
$f_5$	80.0 (9.3)	80.0 (21.4)	80.0 (16.5)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	80.0 (44.7)	80.0 (44.7)	60.0 (54.8)	60.0 (54.8)
$f_6$	83.3 (8.8)	88.9 (11.1)	77.8 (13.6)	60.0 (54.8)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)
$f_7$	24.4 (8.4)	24.4 (9.3)	24.4 (9.3)	20.0 (44.7)	60.0 (54.8)	0.0 (0.0)	80.0 (44.7)	0.0 (0.0)	20.0 (44.7)	0.0 (0.0)	0.0 (0.0)	40.0 (54.8)
$f_8$	92.2 (5.0)	84.4 (9.9)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	40.0 (54.8)	40.0 (54.8)	100.0 (0.0)
$f_9$	64.4 (12.8)	75.0 (9.2)	53.9 (29.1)	40.0 (54.8)	100.0 (0.0)	60.0 (54.8)	80.0 (44.7)	80.0 (44.7)	60.0 (54.8)	100.0 (0.0)	40.0 (54.8)	100.0 (0.0)
$f_{10}$	70.0 (15.0)	80.0 (9.3)	60.0 (29.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	40.0 (54.8)	100.0 (0.0)	40.0 (54.8)	80.0 (44.7)
$f_{11}$	80.5 (9.3)	83.3 (12.8)	77.7 (10.0)	60.0 (54.8)	100.0 (0.0)	20.0 (44.7)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	20.0 (44.7)	60.0 (54.8)	80.0 (44.7)
$f_{1-11}$	74.9 (3.1)	78.2 (3.7)	71.6 (6.3)	76.4 (10.4)	94.5 (5.0)	76.4 (5.9)	88.7 (8.1)	78.2 (15.2)	78.5 (13.9)	78.7 (5.0)	47.3 (23.5)	84.7 (8.9)

Tab. A.19: Zestawienie wskaźników  $r_{fd}$  i  $r_{td}$  dla rozwiązania optymalnego  $o_7$ .

	$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$f_0$	20.8 (3.8)	11.8 (3.6)	29.8 (5.4)	44.0 (11.4)	14.0 (11.4)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	10.0 (10.0)	8.0 (8.4)	8.0 (11.0)	22.0 (21.7)
$f_1$	66.7 (16.2)	66.7 (7.9)	66.7 (26.1)	100.0 (0.0)	100.0 (0.0)	0.0 (0.0)	0.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	20.0 (44.7)
$f_2$	85.8 (8.8)	87.1 (4.0)	84.4 (14.9)	100.0 (0.0)	80.0 (44.7)	0.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)
$f_3$	83.3 (5.6)	84.4 (9.9)	82.2 (9.9)	100.0 (0.0)	100.0 (0.0)	0.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)
$f_4$	80.0 (8.4)	77.8 (7.9)	82.2 (14.9)	100.0 (0.0)	100.0 (0.0)	0.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	20.0 (44.7)
$f_5$	78.9 (7.2)	75.6 (18.3)	82.2 (16.9)	80.0 (44.7)	100.0 (0.0)	0.0 (0.0)	80.0 (44.7)	80.0 (44.7)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	60.0 (54.8)
$f_6$	80.0 (3.0)	84.4 (6.1)	75.6 (9.3)	100.0 (0.0)	100.0 (0.0)	0.0 (0.0)	60.0 (54.8)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)
$f_7$	25.6 (8.4)	28.9 (9.9)	22.2 (7.9)	100.0 (0.0)	60.0 (54.8)	0.0 (0.0)	40.0 (54.8)	0.0 (0.0)	20.0 (44.7)	0.0 (0.0)	20.0 (44.7)	20.0 (44.7)
$f_8$	91.1 (3.0)	82.2 (6.1)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	0.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	60.0 (54.8)	100.0 (0.0)
$f_9$	65.7 (9.8)	79.7 (9.0)	51.7 (26.0)	100.0 (0.0)	100.0 (0.0)	0.0 (0.0)	80.0 (44.7)	80.0 (44.7)	60.0 (54.8)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)
$f_{10}$	58.9 (19.9)	57.8 (12.2)	60.0 (29.0)	100.0 (0.0)	100.0 (0.0)	0.0 (0.0)	100.0 (0.0)	40.0 (54.8)	40.0 (54.8)	80.0 (44.7)	40.0 (54.8)	20.0 (44.7)
$f_{11}$	84.5 (11.2)	91.1 (19.9)	77.7 (10.0)	100.0 (0.0)	100.0 (0.0)	0.0 (0.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	20.0 (44.7)	80.0 (44.7)	100.0 (0.0)
$f_{1-11}$	72.6 (3.1)	73.9 (1.7)	71.4 (5.6)	98.2 (4.1)	94.5 (5.0)	0.0 (0.0)	77.6 (8.7)	78.2 (8.1)	78.5 (13.9)	84.4 (13.6)	80.0 (11.9)	69.4 (11.8)

Tab. A.20: Zestawienie wskaźników  $r_{fd}$  i  $r_{td}$  dla rozwiązania optymalnego  $o_8$ .

	$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$f_0$	18.4 (4.8)	11.3 (3.5)	25.6 (6.6)	4.0 (5.5)	26.0 (18.2)	8.0 (11.0)	4.0 (8.9)	8.0 (8.4)	10.0 (10.0)	8.0 (11.0)	22.0 (17.9)	12.0 (8.4)
$f_1$	62.2 (10.7)	62.2 (16.9)	62.2 (20.2)	0.0 (0.0)	80.0 (44.7)	20.0 (44.7)	20.0 (44.7)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	60.0 (54.8)
$f_2$	94.4 (6.8)	97.8 (5.0)	91.1 (9.3)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)
$f_3$	85.6 (6.3)	95.6 (9.9)	75.6 (9.3)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)
$f_4$	90.0 (7.2)	93.3 (6.1)	86.7 (12.2)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	40.0 (54.8)
$f_5$	72.2 (18.8)	80.0 (27.7)	64.4 (16.5)	60.0 (54.8)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	60.0 (54.8)	80.0 (44.7)	80.0 (44.7)	80.0 (44.7)	80.0 (44.7)
$f_6$	81.1 (13.9)	86.7 (9.3)	75.6 (21.4)	40.0 (54.8)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)
$f_7$	28.9 (8.2)	33.3 (7.9)	24.4 (9.3)	0.0 (0.0)	80.0 (44.7)	20.0 (44.7)	20.0 (44.7)	0.0 (0.0)	40.0 (54.8)	20.0 (44.7)	80.0 (44.7)	40.0 (54.8)
$f_8$	95.6 (4.6)	91.1 (9.3)	100.0 (0.0)	100.0 (0.0)	60.0 (54.8)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	60.0 (54.8)	100.0 (0.0)
$f_9$	65.0 (13.3)	75.3 (14.2)	54.7 (15.1)	0.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	60.0 (54.8)	40.0 (54.8)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)
$f_{10}$	67.8 (16.9)	73.3 (6.1)	62.2 (30.0)	40.0 (54.8)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	60.0 (54.8)	40.0 (54.8)	100.0 (0.0)	60.0 (54.8)	80.0 (44.7)
$f_{11}$	71.8 (3.8)	81.1 (16.8)	61.6 (12.4)	40.0 (54.8)	60.0 (54.8)	20.0 (44.7)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	20.0 (44.7)	80.0 (44.7)	100.0 (0.0)
$f_{1-11}$	74.1 (3.2)	79.0 (4.1)	69.1 (3.8)	52.7 (11.9)	85.5 (10.4)	80.2 (7.4)	79.5 (8.3)	78.2 (8.1)	80.3 (10.1)	88.4 (4.7)	83.6 (17.5)	83.0 (9.9)

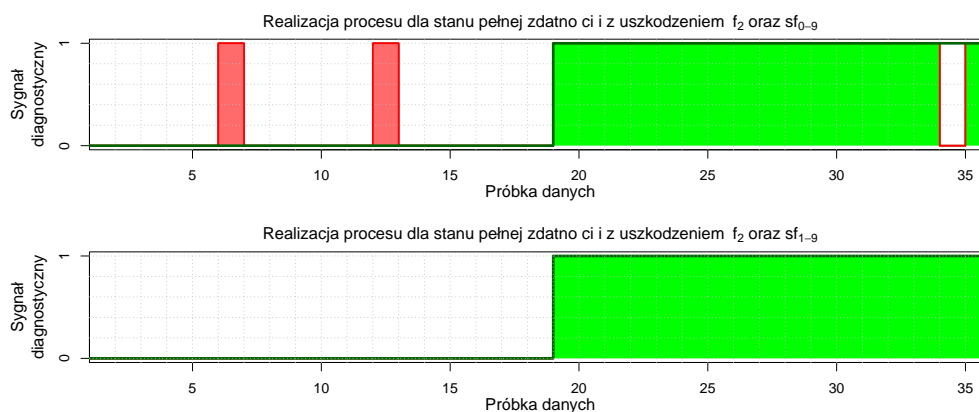
Tab. A.21: Zestawienie wskaźników  $r_{fd}$  i  $r_{td}$  dla pojedynczego klasyfikatora drzewa decyzyjnego.

	$sf_{0-9}$	$sf_{1-9}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$
$f_0$	16.9 (2.6)	13.6 (4.1)	20.2 (1.4)	18.0 (13.0)	22.0 (13.0)	10.0 (14.1)	2.0 (4.5)	20.0 (18.7)	12.0 (8.4)	8.0 (11.0)	10.0 (14.1)	20.0 (12.2)
$f_1$	60.0 (17.3)	77.8 (15.7)	42.2 (24.1)	60.0 (54.8)	80.0 (44.7)	60.0 (54.8)	20.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)
$f_2$	89.1 (7.7)	89.1 (13.6)	88.9 (7.9)	60.0 (54.8)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	100.0 (0.0)	60.0 (54.8)
$f_3$	88.9 (8.8)	93.3 (9.9)	84.4 (12.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)
$f_4$	90.0 (7.2)	88.9 (7.9)	91.1 (9.3)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	60.0 (54.8)
$f_5$	80.0 (3.0)	86.7 (18.3)	73.3 (14.9)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	80.0 (44.7)	80.0 (44.7)	60.0 (54.8)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)
$f_6$	80.0 (8.4)	88.9 (13.6)	71.1 (6.1)	80.0 (44.7)	80.0 (44.7)	60.0 (54.8)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)
$f_7$	16.7 (6.8)	11.1 (7.9)	22.2 (13.6)	20.0 (44.7)	20.0 (44.7)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	20.0 (44.7)	0.0 (0.0)	20.0 (44.7)	20.0 (44.7)
$f_8$	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)
$f_9$	75.8 (11.1)	76.9 (12.3)	74.7 (13.0)	60.0 (54.8)	80.0 (44.7)	40.0 (54.8)	80.0 (44.7)	80.0 (44.7)	80.0 (44.7)	100.0 (0.0)	60.0 (54.8)	100.0 (0.0)
$f_{10}$	63.3 (21.0)	75.6 (9.3)	51.1 (35.7)	40.0 (54.8)	100.0 (0.0)	40.0 (54.8)	100.0 (0.0)	100.0 (0.0)	80.0 (44.7)	100.0 (0.0)	40.0 (54.8)	80.0 (44.7)
$f_{11}$	92.8 (6.6)	92.8 (6.6)	92.8 (6.6)	60.0 (54.8)	100.0 (0.0)	20.0 (44.7)	80.0 (44.7)	100.0 (0.0)	100.0 (0.0)	20.0 (44.7)	100.0 (0.0)	100.0 (0.0)
$f_{1-11}$	75.9 (3.4)	80.0 (3.9)	71.8 (6.5)	70.9 (7.6)	85.5 (5.0)	68.5 (8.6)	77.8 (4.7)	87.3 (8.1)	82.0 (18.3)	82.4 (11.3)	80.0 (7.6)	84.5 (12.3)

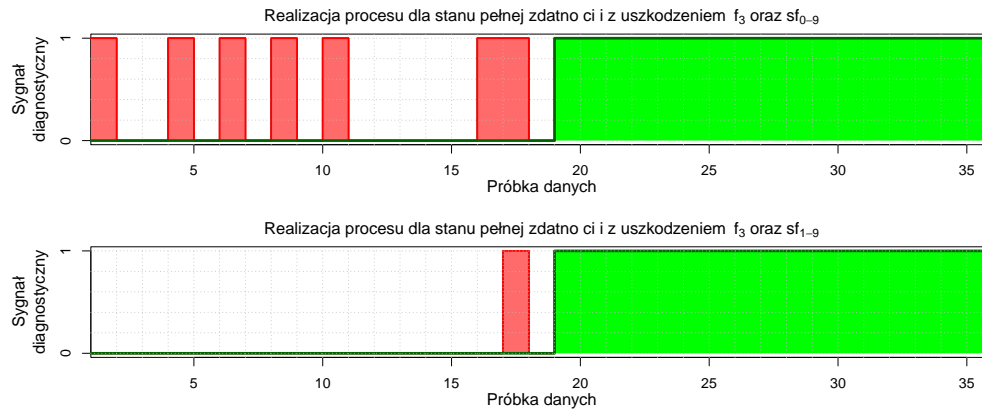
Na każdym kolejnym wykresie od A.6 do A.16 zestawiono pojedynczy przebieg dla stanu pełnej zdatości z pojedynczym przebiegiem procesu z wybranym uszkodzeniem. Dodatkowo w ramach jednej grupy wykresów porównano takie zestawienie z uwzględnieniem nieustalonego stanu funkcjonalnego  $sf_0$  (górny wykres) oraz z pominięciem nieustalonego stanu funkcjonalnego  $sf_0$  (dolny wykres).



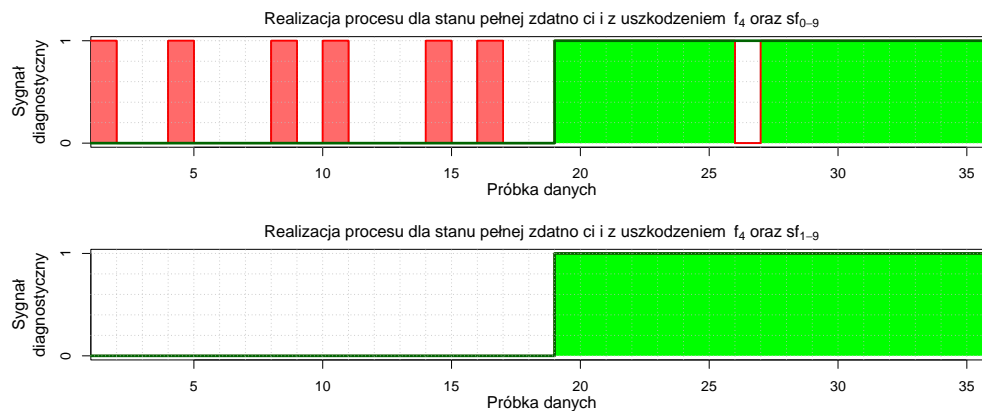
Rys. A.6: Porównanie wpływu stanu funkcjonalnego  $sf_0$  na liczbę generowanych fałszywych i prawidłowych alarmów.



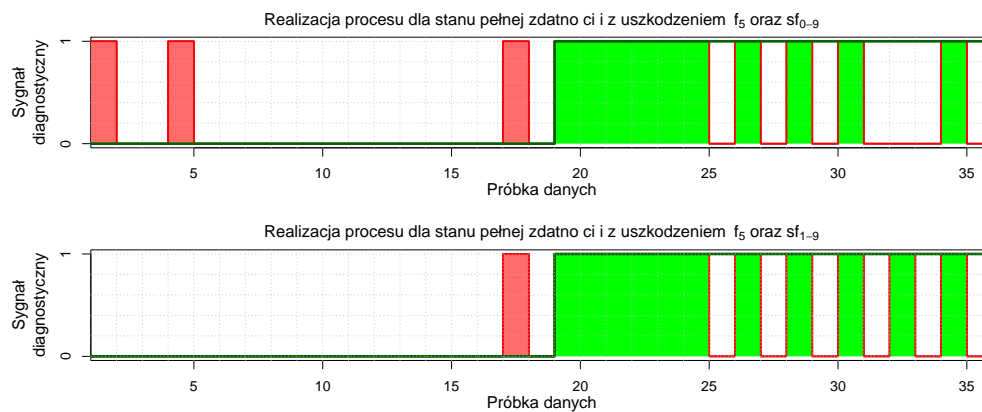
Rys. A.7: Porównanie wpływu stanu funkcjonalnego  $sf_0$  na liczbę generowanych fałszywych i prawidłowych alarmów.



Rys. A.8: Porównanie wpływu stanu funkcjonalnego  $sf_0$  na liczbę generowanych fałszywych i prawidłowych alarmów.



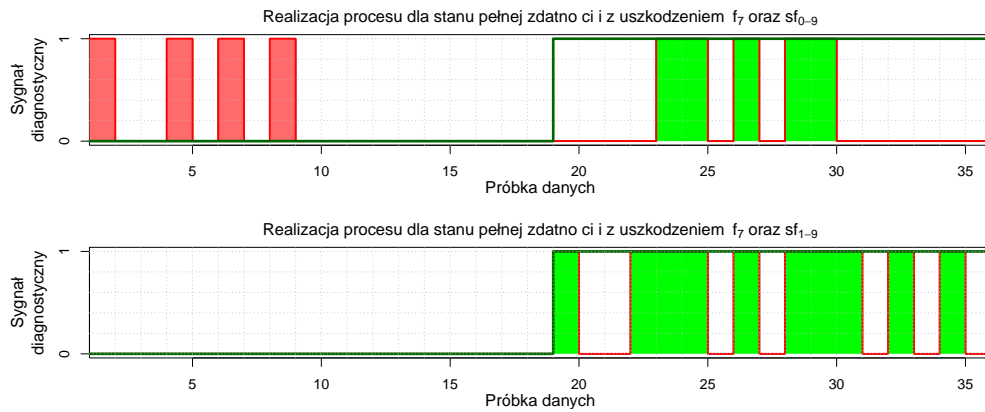
Rys. A.9: Porównanie wpływu stanu funkcjonalnego  $sf_0$  na liczbę generowanych fałszywych i prawidłowych alarmów.



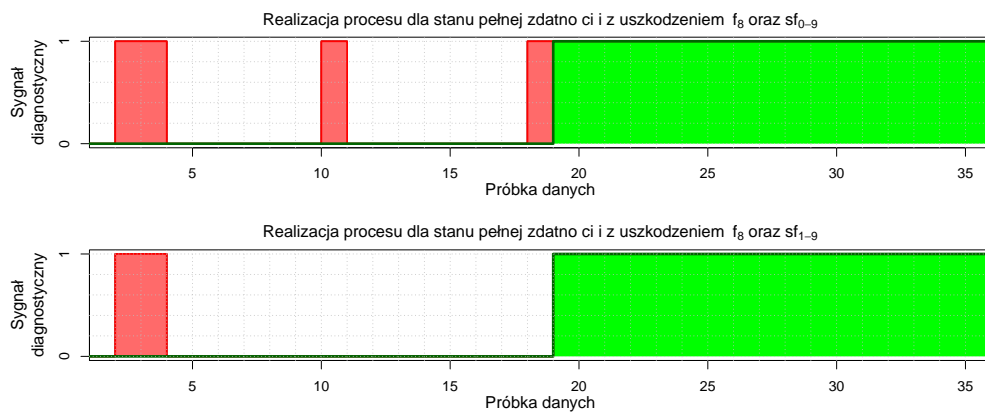
Rys. A.10: Porównanie wpływu stanu funkcjonalnego  $sf_0$  na liczbę generowanych fałszywych i prawidłowych alarmów.



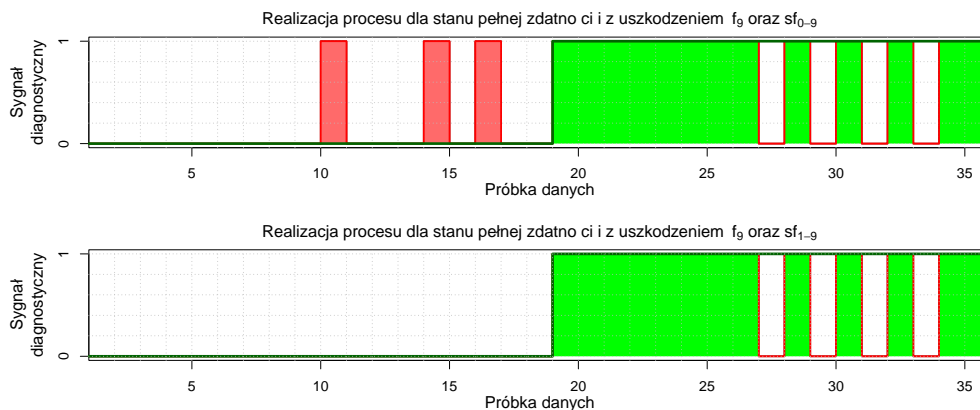
Rys. A.11: Porównanie wpływu stanu funkcjonalnego  $sf_0$  na liczbę generowanych fałszywych i prawidłowych alarmów.



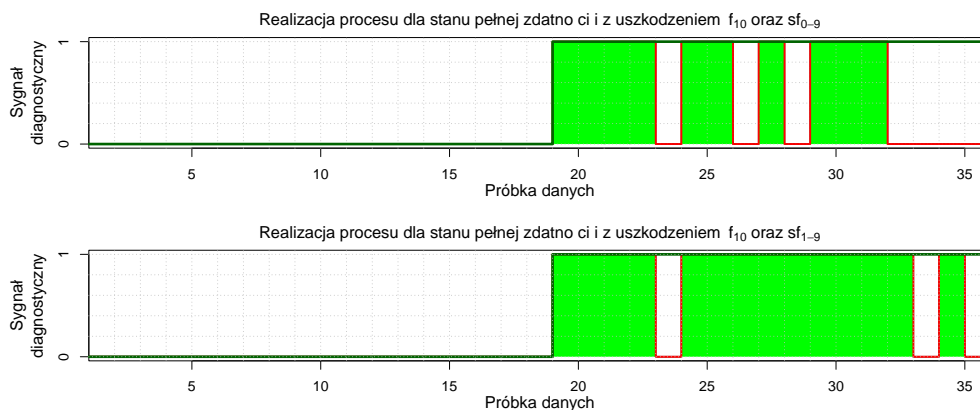
Rys. A.12: Porównanie wpływu stanu funkcjonalnego  $sf_0$  na liczbę generowanych fałszywych i prawidłowych alarmów.



Rys. A.13: Porównanie wpływu stanu funkcjonalnego  $sf_0$  na liczbę generowanych fałszywych i prawidłowych alarmów.



Rys. A.14: Porównanie wpływu stanu funkcjonalnego  $sf_0$  na liczbę generowanych fałszywych i prawdziwych alarmów.



Rys. A.15: Porównanie wpływu stanu funkcjonalnego  $sf_0$  na liczbę generowanych fałszywych i prawdziwych alarmów.



Rys. A.16: Porównanie wpływu stanu funkcjonalnego  $sf_0$  na liczbę generowanych fałszywych i prawdziwych alarmów.



W dalszej części porównano złożoności bazowych klasyfikatorów klasyfikatora kontekstowego dla wszystkich rozwiązań optymalnych  $\mathbf{O}$  z podziałem na poszczególne stany funkcjonalne  $sf_i$ . Każda kolejna tabela od A.22 do A.26 prezentuje wyniki dla innego zbioru treningowego  $Et$ .

Tab. A.22: Porównanie poziomów złożoności uzyskanych klasyfikatorów kontekstowych dla zbioru danych treningowych  $Et_1$  ze względu na rozwiązanie optymalne  $o_i$  z podziałem na stany funkcjonalne  $sf_i$ .

$\mathbf{O}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$	$sf_{1-9}$
$o_1$	350	32	14	38	4	8	16	16	28	32	156
$o_2$	410	34	26	34	4	6	30	6	2	32	142
$o_3$	402	32	14	38	22	16	16	20	28	30	186
$o_4$	362	48	14	30	18	20	30	18	20	28	198
$o_5$	410	32	14	30	4	16	4	6	16	28	122
$o_6$	354	32	22	38	20	22	16	6	6	30	162
$o_7$	350	8	28	0	4	6	16	20	22	32	104
$o_8$	456	2	38	34	4	16	30	14	8	26	146

Tab. A.23: Porównanie poziomów złożoności uzyskanych klasyfikatorów kontekstowych dla zbioru danych treningowych  $Et_2$  ze względu na rozwiązanie optymalne  $o_i$  z podziałem na stany funkcjonalne  $sf_i$ .

$\mathbf{O}$	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$	$sf_{1-9}$
$o_1$	346	32	12	36	4	10	14	12	26	32	146
$o_2$	458	34	22	38	4	10	14	6	2	32	130
$o_3$	370	32	22	16	22	20	4	22	26	32	164
$o_4$	358	50	12	34	20	16	28	16	22	32	198
$o_5$	384	32	14	34	4	26	4	8	20	32	142
$o_6$	344	32	16	36	26	16	14	6	6	32	152
$o_7$	346	12	16	0	4	4	14	22	26	40	98
$o_8$	464	2	36	38	4	24	14	6	12	24	136

Tab. A.24: Porównanie poziomów złożoności uzyskanych klasyfikatorów kontekstowych dla zbioru danych treningowych  $Et_3$  ze względu na rozwiązanie optymalne  $o_i$  z podziałem na stany funkcjonalne  $sf_i$ .

<b>O</b>	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$	$sf_{1-9}$
$o_1$	334	28	22	42	4	10	16	12	26	32	160
$o_2$	494	10	18	30	4	10	20	12	2	32	106
$o_3$	402	26	28	34	26	14	4	22	26	30	180
$o_4$	346	34	20	28	20	14	26	20	22	28	184
$o_5$	408	10	14	28	4	10	4	12	18	28	100
$o_6$	342	26	14	34	20	14	16	4	6	30	134
$o_7$	334	6	20	0	4	6	16	13	26	40	92
$o_8$	470	2	34	30	4	30	20	14	12	28	146

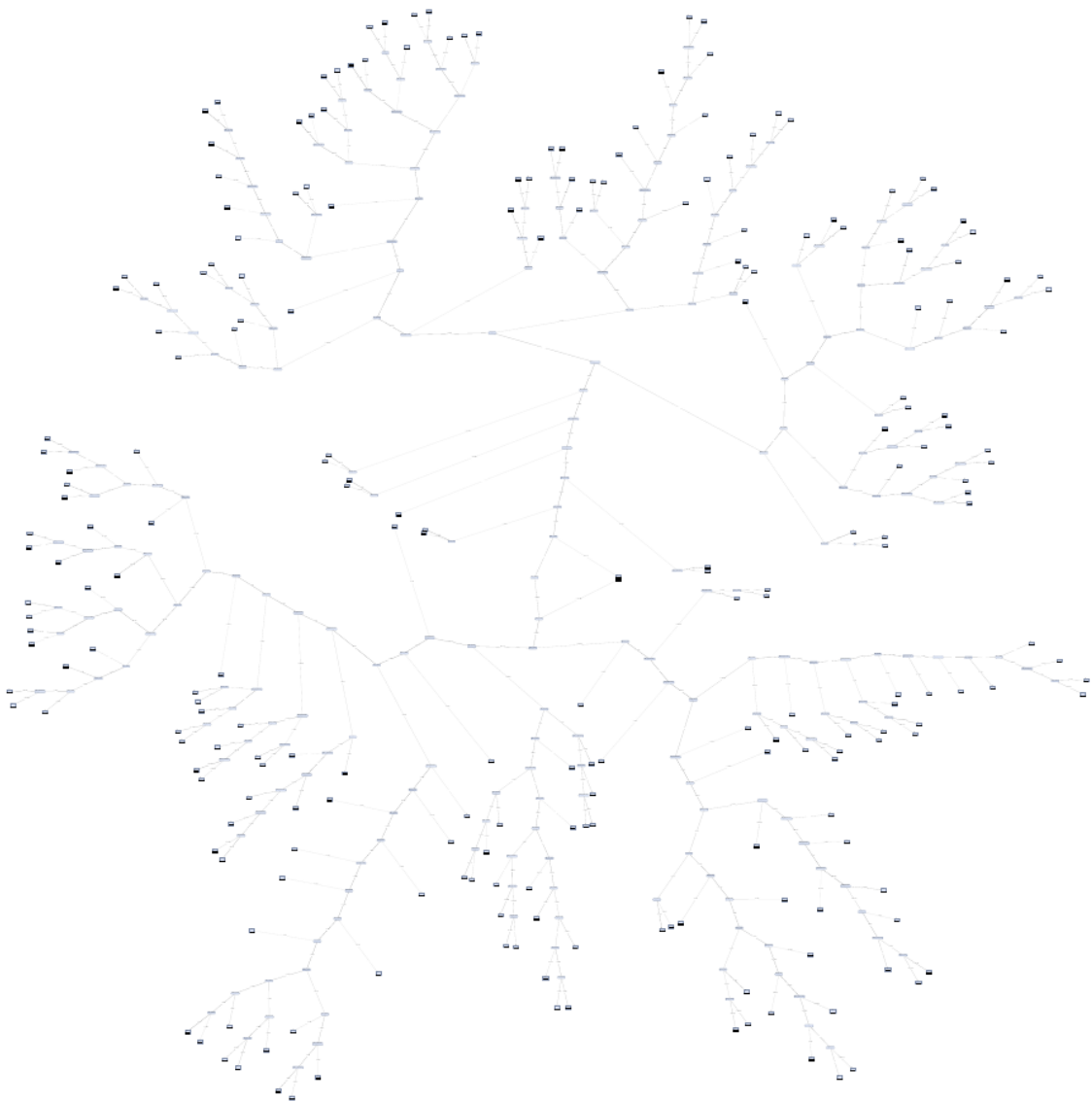
Tab. A.25: Porównanie poziomów złożoności uzyskanych klasyfikatorów kontekstowych dla zbioru danych treningowych  $Et_4$  ze względu na rozwiązanie optymalne  $o_i$  z podziałem na stany funkcjonalne  $sf_i$ .

<b>O</b>	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$	$sf_{1-9}$
$o_1$	340	32	14	36	4	14	20	18	28	32	166
$o_2$	470	36	28	34	4	14	20	16	2	36	154
$o_3$	424	32	34	34	22	20	8	18	28	28	196
$o_4$	256	8	16	32	20	20	32	18	20	22	166
$o_5$	426	32	16	32	4	12	4	8	24	24	132
$o_6$	340	32	18	32	20	14	20	6	16	24	158
$o_7$	340	8	20	0	4	6	20	18	18	34	94
$o_8$	480	2	42	34	4	24	20	26	8	30	160

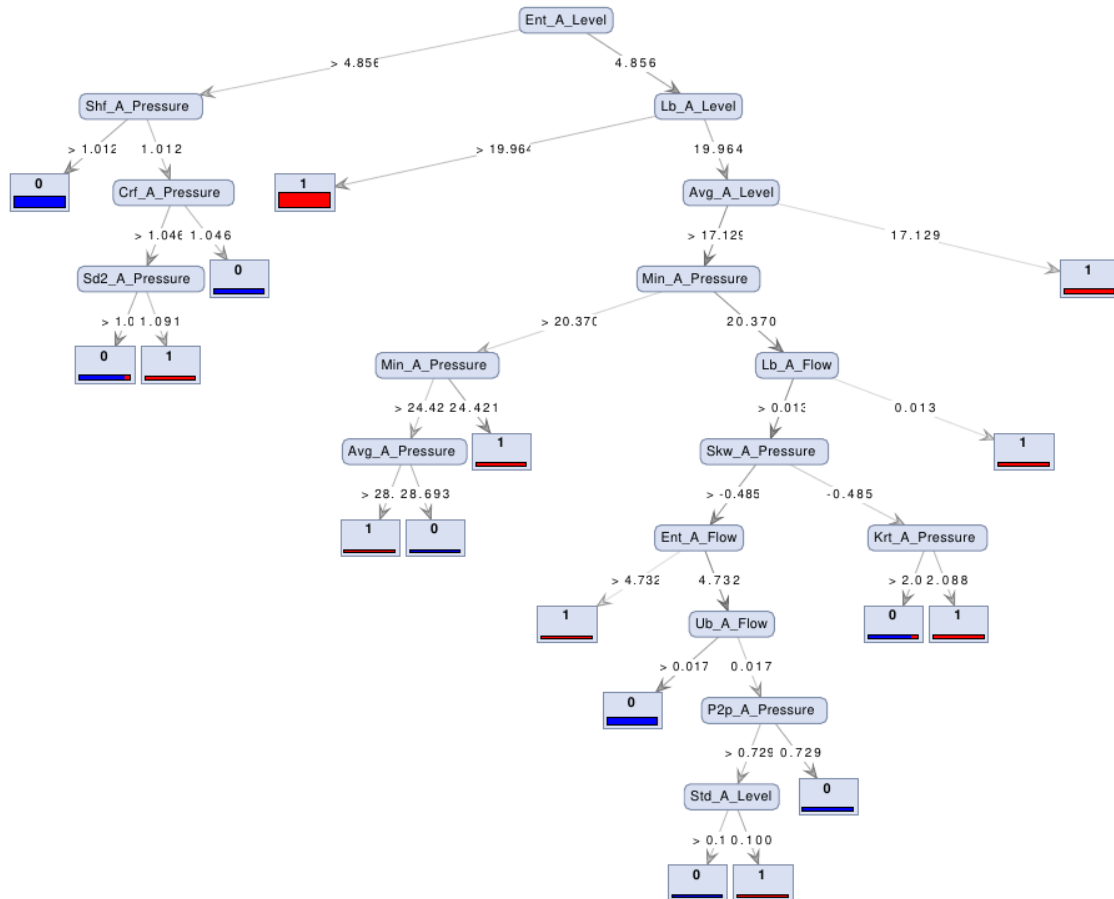
Tab. A.26: Porównanie poziomów złożoności uzyskanych klasyfikatorów kontekstowych dla zbioru danych treningowych  $Et_5$  ze względu na rozwiązanie optymalne  $o_i$  z podziałem na stany funkcjonalne  $sf_i$ .

<b>O</b>	$sf_0$	$sf_1$	$sf_2$	$sf_3$	$sf_4$	$sf_5$	$sf_6$	$sf_7$	$sf_8$	$sf_9$	$sf_{1-9}$
$o_1$	322	36	10	36	4	18	4	18	28	30	154
$o_2$	434	14	24	30	20	16	14	12	2	30	132
$o_3$	400	30	32	32	22	28	4	20	28	30	196
$o_4$	356	44	10	26	22	16	26	20	16	30	180
$o_5$	410	30	14	26	10	16	4	12	22	30	134
$o_6$	342	30	16	30	20	16	4	6	6	30	128
$o_7$	322	6	16	0	4	6	4	14	26	28	76
$o_8$	488	2	32	30	18	20	14	6	12	30	134

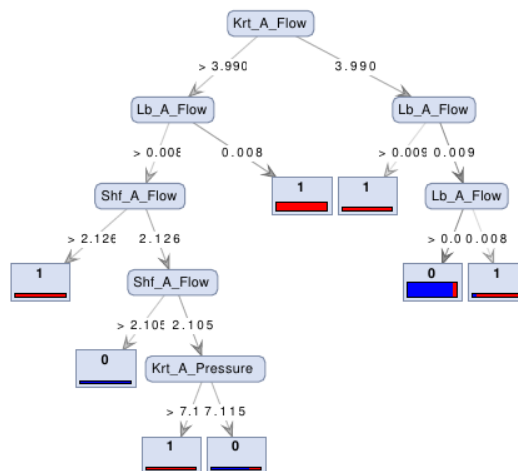
Ostatnia część rozdziału również dotyczy złożoności klasyfikatorów, która została zaprezentowana w postaci grafu drzewa decyzyjnego. Na rysunku A.17 pokazano kształt drzewa decyzyjnego dla pojedynczego drzewa decyzyjnego wytrenowanego przy pomocy danych  $Et_1$ . Kolejne rysunki od A.18 do A.26 przedstawiają struktury drzew decyzyjnych utworzonych dla klasyfikatora kontekstowego wytrenowanego za pomocą danych  $Et_1$  z uwzględnieniem parametrów algorytmów otrzymanych w rozwiązaniu optymalnym  $o_6$ .



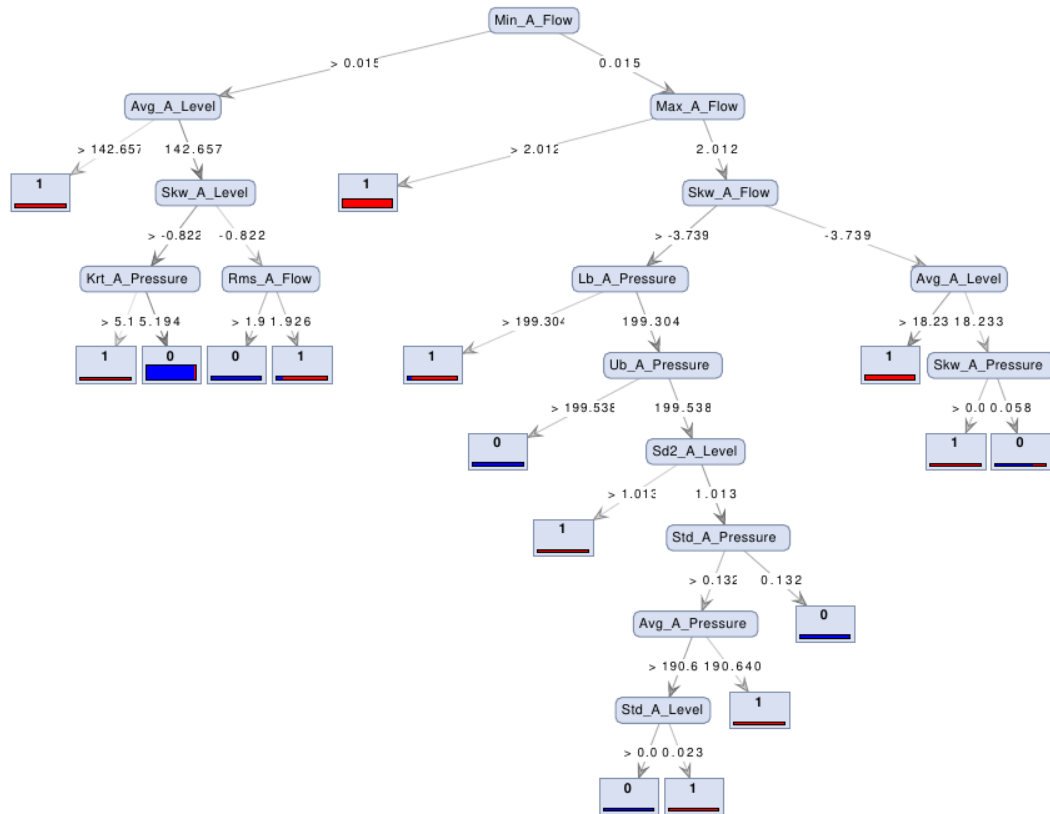
Rys. A.17: Wizualizacja struktury drzewa decyzyjnego uzyskanego dla zbioru danych treningowych  $Et_1$



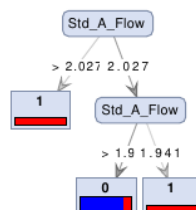
Rys. A.18: Wizualizacja struktury drzewa decyzyjnego, utworzonego w ramach klasyfikatora kontekstowego dla stanu funkcjonalnego  $sf_1$



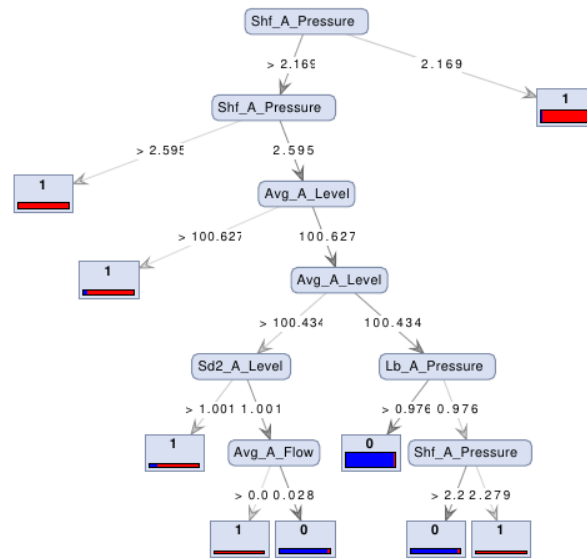
Rys. A.19: Wizualizacja struktury drzewa decyzyjnego, utworzonego w ramach klasyfikatora kontekstowego dla stanu funkcjonalnego  $sf_2$



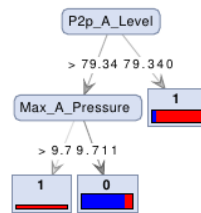
Rys. A.20: Wizualizacja struktury drzewa decyzyjnego, utworzonego w ramach klasyfikatora kontekstowego dla stanu funkcjonalnego  $sf_3$



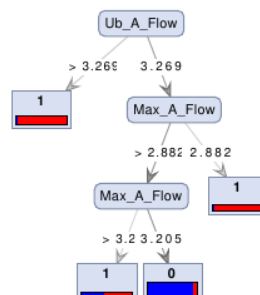
Rys. A.21: Wizualizacja struktury drzewa decyzyjnego, utworzonego w ramach klasyfikatora kontekstowego dla stanu funkcjonalnego  $sf_4$



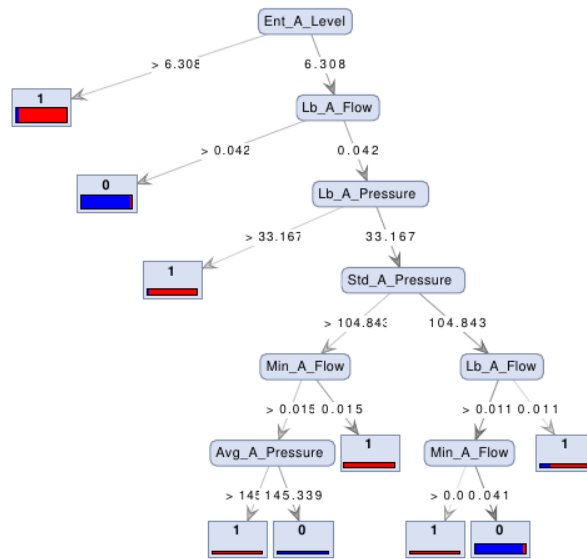
Rys. A.22: Wizualizacja struktury drzewa decyzyjnego, utworzonego w ramach klasyfikatora kontekstowego dla stanu funkcjonalnego  $sf_5$



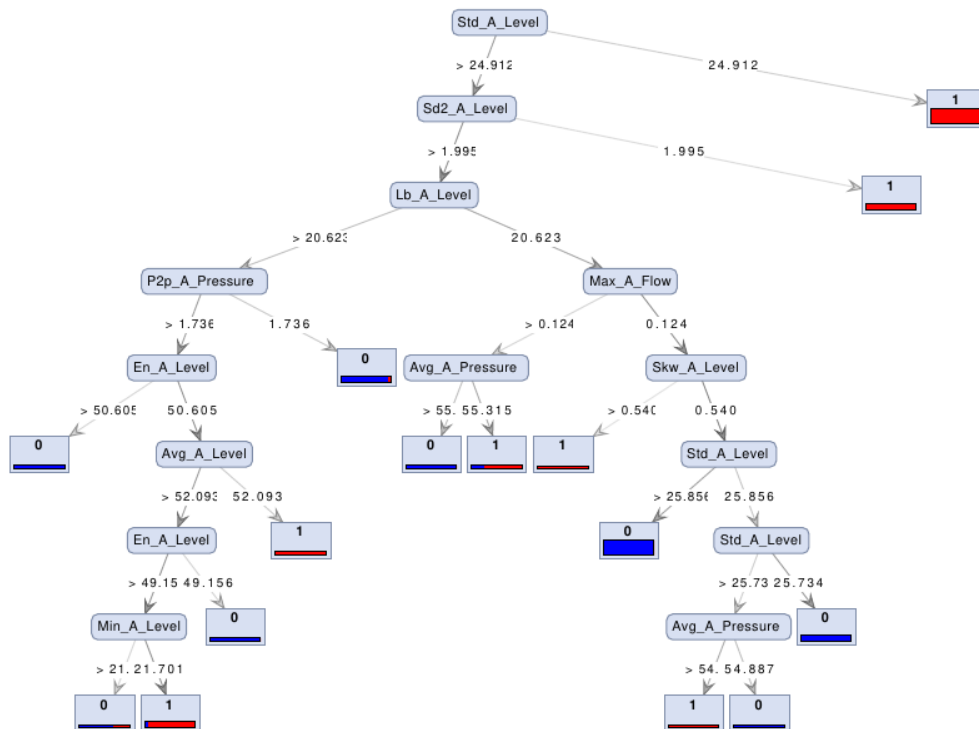
Rys. A.23: Wizualizacja struktury drzewa decyzyjnego, utworzonego w ramach klasyfikatora kontekstowego dla stanu funkcjonalnego  $sf_6$



Rys. A.24: Wizualizacja struktury drzewa decyzyjnego, utworzonego w ramach klasyfikatora kontekstowego dla stanu funkcjonalnego  $sf_7$



Rys. A.25: Wizualizacja struktury drzewa decyzyjnego, utworzonego w ramach klasyfikatora kontekstowego dla stanu funkcjonalnego  $sf_8$



Rys. A.26: Wizualizacja struktury drzewa decyzyjnego, utworzonego w ramach klasyfikatora kontekstowego dla stanu funkcjonalnego  $sf_9$





# Bibliografia

- Adair W., Buchan N., Chen X.-P. i Liu D. (2015): *A model of communication context and measure of context dependence*. *Academy of Management Discoveries* **2**.
- Afzaal M., Ahmad D., Imran M. i Xiangtao D. (2020): *Artificial intelligence, context, and meaning making in language: A rationalization approach*. *International Journal of Future Generation Communication and Networking* : pp115–122.
- Akman V. i Surav M. (1996): *Steps toward formalizing context*. *AI Magazine* **17**(3): pp. 55.
- Alegre U., Augusto J. C. i Clark T. (2016): *Engineering context-aware systems and applications: A survey*. *Journal of Systems and Software* **117**: pp. 55–83.
- Alippi C., Ntalampiras S. i Roveri M. (2017): *Model-free fault detection and isolation in large-scale cyber-physical systems*. *IEEE Transactions on Emerging Topics in Computational Intelligence* **1**(1): pp. 61–71.
- Amarowicz M. (2015): *Designing of the diagnostic systems based on the sets of requirements*. *Machine Dynamics Research* **39**: pp. 125–134.
- Asadi Majd A., Samet H. i Ghanbari T. (2017): *k-nn based fault detection and classification methods for power transmission systems*. *Protection and Control of Modern Power Systems* **2**.
- Azad M. i Moshkov M. (2015): *Classification and optimization of decision trees for inconsistent decision tables represented as mvd tables*. *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*: pp. 31–38.
- Baldauf M., Dustdar S. i Rosenberg F. (2007): *A survey on context-aware systems*. *International journal of ad hoc and ubiquitous computing* **2**(4): pp. 263–277.
- Bartyś M., Patton R., Syfert M., de las Heras S. i Quevedo J. (2006): *Introduction to the damadics actuator fdi benchmark study*. *Control Engineering Practice* **14**(6): pp. 577–596. *A Benchmark Study of Fault Diagnosis for an Industrial Actuator*.
- Benkercha R. i Moulahoum S. (2018): *Fault detection and diagnosis based on c4.5 decision tree algorithm for grid connected pv system*. *Solar Energy* **173**: pp. 610–634.
- Blanke M., Kinnaert M., Lunze J. i Staroświecki M. (2006): *Diagnosis and fault-tolerant control*. Springer-Verlag Berlin Heidelberg.

- Bloch I. (1996): *Information combination operators for data fusion: a comparative review with classification*. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans **26**(1): pp. 52–67.
- Brézillon P. (1999): *Context in artificial intelligence: I. a survey of the literature*. Computers and Artificial Intelligence **18**.
- Caccavale F. i Villani L. (2003): *Fault diagnosis and fault tolerance for mechatronic systems: Recent advances*. Springer Tracts in Advanced Robotics. Springer Berlin/Heidelberg.
- Cenacewicz K. i Katunin A. (2016): Modeling and simulation of longwall scraper conveyor considering operational faults.
- Chandrashekar G. i Sahin F. (2014): *A survey on feature selection methods*. Computers and Electrical Engineering **40**(1): pp. 16–28.
- Chen, Jie and Patton R. J. (1998): *Robust model-based fault diagnosis for dynamic systems*. Springer.
- Chen K., Huang C. i He J. (2016): *Fault detection, classification and location for transmission lines and distribution systems: a review on the methods*. High Voltage **1**(1): pp. 25–33.
- Chen M., Zheng A. X., Lloyd J., Jordan M. I. i Brewer E. (2004): *Failure diagnosis using decision trees*. International Conference on Autonomic Computing, 2004. Proceedings: pp. 36–43.
- Cholewa W. (2010): *Szkieletowy system doradczy MMNET*. Katedra Podstaw Konstrukcji Maszyn. Wydział Mechaniczny Technologiczny. Politechnika Śląska.
- Cholewa W. i Kaźmierczak J. (1996): *Diagnostyka techniczna maszyn: przetwarzanie cech sygnałów*. Wydawnictwo Politechniki Śląskiej.
- Cholewa W., Kosmowski K. i Radkowski S. (2008): *Modele systemów oceny ryzyka i diagnostyki technicznej*. Katedra Podstaw Konstrukcji Maszyn, Politechnika Śląska.
- Cholewa W. i Pedrycz W. (1987): *Systemy doradcze*. Wydawnictwo Politechniki Śląskiej.
- Cholewa W. i Skupnik D. (2011): *Badania porównawcze wielomodalnych sieci stwierdzeń*. Pomiary, Automatyka, Kontrola **57**(11): pp. 1388–1393.
- Cichosz P. (2000): *Systemy uczące się*. Warszawa: WNT.
- Deb K., Pratap A., Agarwal S. i Meyarivan T. (2002): *A fast and elitist multiobjective genetic algorithm: Nsga-ii*. IEEE Transactions on Evolutionary Computation **6**(2): pp. 182–197.
- Deza M. M. i Deza E. (2009): *Encyclopedia of distances*. Springer Berlin Heidelberg. Berlin, Heidelberg.
- Dhibi K., Mansouri M., Bouzrara K., Nounou H. i Nounou M. (2021): *An enhanced ensemble learning-based fault detection and diagnosis for grid-connected pv systems*. IEEE Access **1**: pp. 1–10.

- Drabik L., Kubiak-Sokół A. i Sobol E. (2021): *Słownik języka polskiego PWN*. Wydawnictwo Naukowe PWN. Warszawa.
- Esteban J., Starr A., Willetts R., Hannah P. i Bryanston-cross P. (2005): *A review of data fusion models and architectures: towards engineering guidelines*. *Neural Computing and Applications* **14**: pp. 273–281.
- Fidali M. (n.d.): *Metody łącznej analizy sygnałów w wielokanałowych układach diagnozujących maszyny wirnikowych*.
- Han J., Kamber M. i Pei J. (2012): *Data mining concepts and techniques, third edition*. Morgan Kaufmann Publishers.
- Hańderek W. (2020): *System detekcji cyberataków na przemysłowe systemy sterowania*.
- Hou P., Yi X. J. i Chen Y. F. (2017): *Fault diagnosis of rolling element bearing using naïve bayes classifier*. *Vibroengineering PROCEDIA* **14**: pp. 64–69.
- Isermann R. (1993): *Fault diagnosis of machines via parameter estimation and knowledge processing*. *Automatica* **29**(4): pp. 815–835.
- Isermann R. (1997): *Supervision, fault-detection and fault-diagnosis methods — an introduction*. *Control Engineering Practice* **5**(5): pp. 639–652.
- Isermann R. (2005): *Model-based fault-detection and diagnosis status and application*. *Annual Reviews in Control* **29**(1): pp. 71–85.
- Isermann R. (2006): *Fault-diagnosis systems. An introduction from fault detection to fault tolerance*. Springer.
- Isermann R. i Balle P. (1997): *Trends in the application of model-based fault detection and diagnosis of technical processes*. *Control Engineering Practice* **5**(5): pp. 709–719.
- Jain A. K., Duin R. P. W. i Mao J. (2000): *Statistical pattern recognition: a review*. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(1): pp. 4–37.
- Jamil M., Sharma S. i Singh R. (2015): *Fault detection and classification in electrical power transmission system using artificial neural network*. *SpringerPlus* **4**: pp. 334.
- Jamrozik W., Fidali M., Bzymek A. i Timofiejczuk A. (2011): *Zastosowanie fuzji obrazów wizyjnych i termowizyjnych do monitorowania i diagnozowania procesu spawania*. *Przegląd Spawalnictwa - Welding Technology Review* **83**.
- Jankowski N. (2014): *Meta-uczenie w inteligencji obliczeniowej*. Akademicka Oficyna Wydawnicza EXIT Andrzej Lang.
- Jensen F. V. i Nielsen T. D. (2007): *Bayesian networks and decision graphs*. 2nd edn. Springer. New York, NY.
- Ji D., Yao X., Li S., Tang Y. i Tian Y. (2021): *Model-free fault diagnosis for autonomous underwater vehicles using sequence convolutional neural network*. *Ocean Engineering* **232**.
- Jović A., Brkić K. i Bogunović N. (2015): *A review of feature selection methods with applications*. 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO): pp. 1200–1205.

- Kalisch M. (2014): *Modelowanie testów diagnostycznych za pomocą metod uczenia maszynowego*. Modelowanie Inżynierskie **22**: pp. 70–77.
- Kalisch M. (2015): *Supervised context classification methods for an industrial machinery*. 2015 Federated Conference on Computer Science and Information Systems (FedCSIS): pp. 1667–1672.
- Kalisch M. (2018): *Application of context-based meta-learning schemes for an industrial device*. Springer: pp. 487–497.
- Kalisch M., Przysańska P. i Timofiejczuk A. (2015): *Actuator fault diagnosis using single and meta-classification strategies*. W: Artificial Intelligence for Knowledge Management, E. Mercier-Laurent, M. L. Owoc i D. Boulanger (Red.). Springer International Publishing. Cham: pp. 132–149.
- Kalisch M., Timofiejczuk A. i Przysańska P. (2020): *Multi-objective optimization of meta-learning scheme for context-based fault detection*. IFSA Publishing: pp. 138–142.
- Kapucu C. i Cubukcu M. (2021): *A supervised ensemble learning method for fault diagnosis in photovoltaic strings*. Energy **227**.
- Kiernan P. (n.d.): *What if everybody in canada flushed at once?*. Pat's Picks . [https://www.patspicks.com/blog/item/what\\_if\\_everybody\\_flushed\\_at\\_once\\_Edmonton\\_water\\_gold\\_medal\\_hockey\\_game/](https://www.patspicks.com/blog/item/what_if_everybody_flushed_at_once_Edmonton_water_gold_medal_hockey_game/), ostatni dostęp: 26.02.2023.
- Korbicz J. (1997): *Neural networks and their application in fault detection and diagnosis*. IFAC Proceedings Volumes **30**(18): pp. 367–372. IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS 97), Kingston upon Hull, UK, 26-28 August 1997.
- Korbicz J., Kościelny J. M., Kowalczyk Z. i Cholewa W. (2004): *Fault diagnosis. Models, artificial intelligence, applications*. Springer Berlin / Heidelberg.
- Korbicz J., Obuchowicz A. i Uciński D. (1994): *Sztuczne sieci neuronowe: podstawy i zastosowania*. Problemy Współczesnej Nauki: Informatyka. Akademicka Oficyna Wydawnicza PLJ.
- Kotelly J. C. (1970): *Context logic. i. fundamental concepts, notations, and derived notions*. Notre Dame Journal of Formal Logic **11**(4): pp. 431 – 446.
- Kościelny J. M. (2001): *Diagnostyka zautomatyzowanych procesów przemysłowych*. Akademicka Oficyna Wydawnicza EXIT. Warszawa.
- Krzyśko M., Wołyński W., Górecki T. i Skorzybut M. (2008): *Systemy uczące się. Rozpoznawanie wzorców, analiza skupień i redukcja wymiarowości*. Wydawnictwa Naukowo-Techniczne, Warszawa.
- Kuncheva L. I. (2004): *Combining pattern classifier: Methods and algorithms*. Wiley-Interscience. New Jersey.
- Kuncheva L. I. (2008): *Classifier ensembles for detecting concept change in streaming data: Overview and perspectives*. Proc. Eur. Conf. Artif. Intell. .

- Lipinski P., Brzychczy E. i Zimroz R. (2020): *Decision tree-based classification for planetary gearboxes' condition monitoring with the use of vibration data in multidimensional symptom space*. *Sensors* **20**(21).
- Mahamdi Y., Boubakeur A., Mekhaldi A. i Benmahamed Y. (2022): *Power transformer fault prediction using naive bayes and decision tree based on dissolved gas analysis*. *ENP Engineering Science Journal* **2**: pp. 1–5.
- Marler R. T. i Arora J. (2004): *Survey of multi-objective optimization methods for engineering*. *Structural and Multidisciplinary Optimization* **26**: pp. 369–395.
- McCarthy J. (1987): *Generality in artificial intelligence*. *Commun. ACM* **30**(12): pp. 1030–1035.
- McCarthy J. (1993): *Notes on formalizing context*. *Proceedings of the 13th International Joint Conference on Artificial Intelligence - Volume 1. IJCAI'93*. Morgan Kaufmann Publishers Inc.. San Francisco, CA, USA: pp. 555–560.
- McCarthy J. (1996): *A logical approach to context*.
- McCarthy J. i Buvac S. (1998): *Formalizing context (expanded notes)*. *CSLI Lecture Notes* **81**: pp. 13–50.
- Moczulski W. (2002): *Diagnostyka techniczna. Metody pozyskiwania wiedzy*. Wydawnictwo Politechniki Śląskiej. Gliwice.
- Moczulski W. (2005): *Methodology of heuristic modelling of dynamic objects and processes for diagnostics and control*. *Recent Developments in Artificial Intelligence Methods*: pp. 123–126.
- Moczulski W., Ciupke K., Przysańska P., Tomasik P., Wachla D., Wiglenda R. i Wyczółkowski R. (2011): *Metodyka budowy systemu monitorowania wycieków w sieciach wodociągowych - przykład zastosowania*. *Proceedings of X Conference Diagnostyka Procesów i Systemów* : pp421–432.
- MOEA Framework (2023): <http://moeaframework.org>, ostatni dostęp: 26.02.2023.
- Muralidharan V. i Sugumaran V. (2012): *A comparative study of naive bayes classifier and bayes net classifier for fault diagnosis of monoblock centrifugal pump using wavelet analysis*. *Applied Soft Computing* **12**(8): pp. 2023–2029.
- Mync P. (2017): *Zastosowanie systemu ekspertowego do diagnozowania wybranego układu automatyki*.
- Nakamura E. F., Loureiro A. A. F. i Frery A. C. (2007): *Information fusion for wireless sensor networks: Methods, models, and classifications*. *ACM Comput. Surv.* **39**(3): pp. 9–es.
- Nascimento N., Alencar P., Lucena C. i Cowan D. (2018): *A context-aware machine learning-based approach*. *Proceedings of the 28th Annual International Conference on Computer Science and Software Engineering. CASCON '18*. IBM Corp.. USA: pp. 40–47.

- Niederliński A. (2000): *Regułowe systemy ekspertowe*. Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego.
- Odgaard P. F., Stoustrup J. i Kinnaert M. (2009): *Fault tolerant control of wind turbines - a benchmark model*. IFAC Proceedings Volumes **42**(8): pp. 155–160. 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes.
- Oowski S. (2006): *Sieci neuronowe do przetwarzania informacji*. Oficyna Wydawnicza PW.
- Oowski S. (2013): *Metody i narzędzia eksploracji danych*. Wydawnictwo BTC.
- Patel J., Patel V. i Patel A. (2013): *Fault diagnostics of rolling bearing based on improve time and frequency domain features using artificial neural networks*. International Journal for Scientific Research and Development **1**(4): pp. 781–788.
- Patton R. J., Frank P. M. i Clark R. N. (2000): *Issues of fault diagnosis for dynamic systems*. Springer-Verlag Berlin and Heidelberg.
- Przystałka P. (2011): *Metodyka modelowania neuronowego w diagnostyce procesów z uwzględnieniem elementów teorii chaosu*. Tom 141 z Zeszyty. Politechnika Śląska, Katedra Podstaw Konstrukcji Maszyn. Gliwice.
- Przystałka P., Kalisch M. i Timofiejczuk A. (2018): *Genetic optimization of meta-learning schemes for context-based fault detection*. W: Advances in Technical Diagnostics, A. Timofiejczuk, B. E. Łazarz, F. Chaari i R. Burdzik (Red.). Springer International Publishing. Cham: pp. 287–297.
- Przystałka P. i Sikora M. (2017): *Zintegrowany, szkieletowy system wspomagania decyzji dla systemów monitorowania procesów, urządzeń i zagrożeń*. Monograficzna seria wydawnictwa Instytutu technik Innowacyjnych EMAG. Instytut Technik Innowacyjnych EMAG.
- Rutkowski L. (2005): *Metody i techniki sztucznej inteligencji: inteligencja obliczeniowa*. Informatyka - Zastosowania. Wydawnictwo Naukowe PWN.
- Samantaray S., Kamwa I. i Joos G. (2011): *Decision tree based fault detection and classification in distance relaying*. International Journal of Engineering Intelligent Systems for Electrical Engineering and Communications **19**: pp. 139.
- Senthamarai Kannan S. i Ramaraj N. (2010): *A novel hybrid feature selection via symmetrical uncertainty ranking based local memetic search algorithm*. Knowledge-Based Systems **23**(6): pp. 580–585.
- Sheriff M. Z., Botre C., Mansouri M., Nounou H., Nounou M. i Karim M. N. (2017): *Process monitoring using data-based fault detection techniques: Comparative studies*. W: Fault Diagnosis and Detection, M. Demetgul i M. Unal (Red.). IntechOpen. Rijeka. chapter 10.
- Singh S., Vajirkar P. i Lee Y. (2003): *Context-based data mining using ontologies*. W: Conceptual Modeling - ER 2003, I.-Y. Song, S. W. Liddle, T.-W. Ling i P. Scheuermann (Red.). Springer Berlin Heidelberg. Berlin, Heidelberg: pp. 405–418.

- Skupnik D. (2009): *Wieloaspektowe modele diagnostyczne*. Tom 140 z *Zeszyty*. Politechnika Śląska, Katedra Podstaw Konstrukcji Maszyn. Gliwice.
- Stachurski A. i Wierzbiński A. (1999): *Podstawy optymalizacji*. Oficyna Wydawnicza Politechniki Warszawskiej.
- Takeuchi H., Ogino S., Watanabe H. i Shirata Y. (2008): *Context-based text mining for insights in long documents*. W: *Practical Aspects of Knowledge Management*, T. Yamaguchi (Red.). Springer Berlin Heidelberg. Berlin, Heidelberg: pp. 123–134.
- Timofiejczuk A. (2008): *Identification of diagnostic rules with the application of an evolutionary algorithm*. *Eksploatacja i Niezawodność - Maintenance and Reliability* **37**: pp. 11–16.
- Timofiejczuk A. (2011): *Metodyka wnioskowania z uwzględnieniem kontekstu w diagnostyce maszyn*. online.
- Turney P. (2002a): *Exploiting context when learning to classify*. CoRR **cs.LG/0212035**.
- Turney P. (2002b): *The management of context-sensitive features: A review of strategies*. CoRR **cs.LG/0212037**.
- Turney P. (2002c): *Robust classification with context-sensitive features*. CoRR **cs.LG/0212041**.
- Vahdat-Nejad H., Ramazani A., Mohammadi T. i Mansoor W. (2016): *A survey on context-aware vehicular network applications*. *Vehicular Communications* **3**: pp. 43–57.
- Verdier G. i Ferreira A. (2011): *Adaptive mahalanobis distance and k-nearest neighbor rule for fault detection in semiconductor manufacturing*. *IEEE Transactions on Semiconductor Manufacturing* **24**(1): pp. 59–68.
- Wachla D. (2018): *A mathematical model of a longwall shearer cutting system with selected faults*. W: *Advances in Technical Diagnostics*, A. Timofiejczuk, B. E. Łazarz, F. Chaari i R. Burdzik (Red.). Springer International Publishing. Cham: pp. 225–237.
- Wachla D., Przyszałka P., Kalisch M., Moczulski W. i Timofiejczuk A. (2015): *Development of expert system shell with context-based reasoning*. Springer International Publishing: pp. 369–382.
- Weiser M. (1991): *The computer for the 21st century*. *Scientific American* **265**(3): pp. 94–104.
- Widmer G. (1997): *Tracking context changes through meta-learning*. *Machine Learning* **27**: pp. 259–286.
- Widmer G. i Kubat M. (1994): *Learning in the presence of concept drift and hidden contexts*. *Machine Learning* **23**.
- Wojtusik J. (2006): *Wielomodele diagnostyczne maszyn wirnikowych*. Katedra Podstaw Konstrukcji Maszyn. Wydział Mechaniczny Technologiczny. Politechnika Śląska. Gliwice.

- Woźniak M. (2006): *Metody fuzji informacji dla komputerowych systemów rozpoznawania*. Oficyna Wydawnicza Politechniki Wrocławskiej. Wrocław.
- Yang N.-C. i Ismail H. (2022): *Voting-based ensemble learning algorithm for fault detection in photovoltaic systems under different weather conditions*. Mathematics **10**(2).
- Yao P., Liu Z., Wang Z. i Bu S. (2012): *Fault signal classification using adaptive boosting algorithm*. Elektronika ir Elektrotechnika **18**(8).
- Ziamba P. (2012): *Redukcja wymiarowości i selekcja cech w zadaniach klasyfikacji i regresji z wykorzystaniem uczenia maszynowego*. Zeszyty Naukowe. Studia Informatica/Uniwersytet Szczeciński .



# Metoda detekcji uszkodzeń w diagnostyce procesów z uwzględnieniem kontekstu

Praca doktorska - Streszczenie

Autor: mgr inż. Mateusz Kalisch

Promotor: dr hab. inż. Timofiejczuk Anna, prof. PŚ

Promotor pomocniczy: dr hab. inż. Przystałka Piotr, prof. PŚ

Politechnika Śląska, Wydział Mechaniczny Technologiczny

Niniejsza praca dotyczy detekcji uszkodzeń z uwzględnieniem kontekstu. Diagnostyka techniczna odgrywa kluczową rolę w efektywnym zarządzaniu procesami przemysłowymi. Głównym celem badań było opracowanie i weryfikacja metody detekcji uszkodzeń bazującej na klasyfikatorze kontekstowym, którego parametry były automatycznie dostrajane za pomocą algorytmu optymalizacji wielokryterialnej NSGA-II. Uwzględnienie kontekstu oraz optymalizacji było istotne z punktu widzenia poszukiwania rozwiązania, które ograniczy złożoność stosowanych klasyfikatorów oraz będzie optymalne w kontekście maksymalizacji detekcji stanu z uszkodzeniem oraz bez uszkodzenia z zachowaniem wysokiej powtarzalności wyników.

W literaturze można znaleźć wiele publikacji związanych z pojęciem kontekstu, jednak definicja i sposób wykorzystania kontekstu różni się od obszaru badań. W przypadku diagnostyki procesów przemysłowych pojęcie kontekstu również jest obecne, ale nie znajduje tak powszechnego zastosowania jak w przypadku innych dziedzin.

W niniejszej rozprawie zaproponowano metodę detekcji uszkodzeń, bazującą na komitecie klasyfikatorów uwzględniających kontekst. Kontekst pozwolił na wyspecjalizowanie bazowych klasyfikatorów do działania w określonym wariacie kontekstu, czyli w ograniczonym zbiorze danych i cech. Dodatkowe wykorzystanie metody optymalizacji wielokryterialnej NSGA-II pozwoliło na automatyczne znalezienie optymalnych wartości parametrów klasyfikatorów, w celu uzyskania zbioru rozwiązań spełniających kryteria minimalizacji wartości wskaźnika fałszywych alarmów oraz maksymalizacji wskaźników prawdziwych alarmów z zachowaniem wysokiej powtarzalności uzyskiwanych wyników.

Weryfikację zaproponowanej metody przeprowadzono dla danych uzyskanych podczas czynnego eksperymentu diagnostycznego zrealizowanego na stanowisku laboratoryjnym FESTO S7 EduTrainer. Stanowisko składa się z elementów instalacji hydraulicznych, takich jak pompy, zawory czy zbiorniki. Podczas eksperymentu symulowano różnego typu uszkodzenia, mogące wystąpić podczas działania procesu przemysłowego. Wyniki uzyskane podczas badań weryfikacyjnych potwierdziły użyteczność zaproponowanej metody w kontekście zdefiniowanego problemu badawczego. Metoda detekcji uszkodzeń bazująca na kontekście pozwoliła na poprawę wskaźników opisujących skuteczność detekcji uszkodzeń oraz znacząco uprościć struktury modeli klasyfikatorów.

**Słowa kluczowe:** diagnostyka procesów technicznych, detekcja uszkodzeń, kontekst, uczenie maszynowe, optymalizacja wielokryterialna.

# Fault detection method in process diagnostics using context-based approach

PhD thesis - Summary

Author: Mateusz Kalisch, MSc, Eng.

Supervisor: Prof. Anna Timofiejczuk, PhD, DSc, Eng.

Assistant supervisor: Prof. Piotr Przystałka, PhD, DSc, Eng.

Silesian University of Technology at Gliwice, Faculty of Mechanical Engineering

The PhD thesis deals with fault detection using context-based approach. It is well-known that fault detection is an absolute must for any practical systems. The main aim of the thesis was to elaborate and verify the method of fault detection using context-based approach whose parameters were automatically tuned using the NSGA-II multi-criteria optimization algorithm. Taking into account the context and the optimization algorithm in the developed method was crucial in finding an optimal solution. The solution should be able to reach as high detection rates as possible for fault and faultless states detection and keep high repeatability of the results. The another goal of the elaborated method was to reduce complexity of the obtained classifier.

In the literature, it is possible to find many publications related to the concept of context, but the definition and use of context differs from the area of research. In the case of diagnostics of industrial processes, the concept of context is also present, but it is not as widely used as in other fields.

This dissertation proposes a fault detection method using context based ensemble classifiers. The context allowed the base classifiers to be specialized to operate in a specific variant of the context, i.e. in a limited set of data and features. The additional use of the NSGA-II multi-criteria optimization method allowed for automatic finding of optimal values of classifier parameters. Each solution found by the multi-criteria optimization method is optimal in the domain of criteria functions. In the case of the elaborated method, optimization algorithm searches for solutions, which are repeatable, which false detection rate of fault is minimal and true detection rate is maximal. The set of found solutions allows to choose one, which is best for specific implementation.

The verification of the proposed method was carried out for data obtained during an experiment carried out on a laboratory stand FESTO S7 EduTrainer. This object exemplifies a physical miniaturization of an industrial installation used, e.g. in a chemical industry or a food industry and makes possible diagnostics of the continuous processes. The object consists of elements of hydraulic systems, such as pumps, valves and tanks. During the experiment, various types of faults that could occur during the operation of an industrial process were simulated. The results obtained during the verification tests confirmed the usefulness of the proposed method in the context of the defined research problem. The method of context based fault detection allowed to improve indicators describing the effectiveness of fault detection and to significantly simplify the structure of classifier models.

**Key words:** fault diagnosis, fault detection, context based reasoning, machine learning, multi-criteria optimization.