



Politechnika
Śląska

POLITECHNIKA ŚLĄSKA
WYDZIAŁ AUTOMATYKI, ELEKTRONIKI I INFORMATYKI
KIERUNEK AUTOMATYKA I ROBOTYKA

Rozprawa doktorska

Przewidywanie awarii urządzeń przemysłowych w oparciu o analizę
wibrodiagnostyczną z wykorzystaniem sztucznej inteligencji

Autor: mgr inż. Paweł Fic

Promotor: prof. dr hab. inż. Adam Czornik

Gliwice, styczeń 2023

*Wszystkim tym, dzięki którym
i pomimo których
udało się niniejszą pracę napisać*

Praca doktorska realizowana w ramach Programu MEiN „Doktorat wdrożeniowy”

Streszczenie

Praca przedstawia opis wdrożeń przemysłowych, wykorzystanie dostępnych na rynku narzędzi chmurowych oraz autorskie zastosowanie algorytmów sztucznej inteligencji do predykcji awarii urządzeń przemysłowych uwzględniających dane procesowe i wibrodiagnostyczne. Wkład we wdrożenia był w pełni kompleksowy. Obejmował pozyskanie klienta, dla którego przeprowadzono wdrożenie, implementację algorytmów i zbioru programów wchodzących w skład systemu predykcji awarii, konsultacje i prezentację wyników klientowi. Porównane zostały narzędzia służące do syntezy dojrzałego systemu predykcji awarii zarówno w chmurze obliczeniowej, jak i posadowionego w infrastrukturze klienta. Wyszczególniono przypadki wdrożeń w których dostęp do internetu był pełny, ograniczony, lub nie było go wcale. Przeprowadzono opis problemów i ograniczeń wynikających z zastosowania konkretnych technologii w różnych konfiguracjach sprzętowych. Przedstawiony został autorski algorytm klasyfikacji i predykcji awarii urządzeń przemysłowych w oparciu o analizę wibrodiagnostyczną z wykorzystaniem sztucznej inteligencji. Ponadto, algorytm umożliwia prezentację wyników w sposób przejrzysty i zrozumiały dla klienta u którego wdrażany jest system przewidywania awarii. Wypracowany algorytm jest największą wartością dodaną z perspektywy badawczej, jaką niesie ze sobą niniejsza praca.

Spis treści

1	Wprowadzenie	1
1.1	Predykcyjne utrzymanie ruchu	2
1.2	Urządzenia brzegowe	2
1.3	Chmura obliczeniowa i język programowania	3
1.4	Tematyka pracy a dotychczasowy stan wiedzy	3
1.4.1	Ekstrakcja cech związanych z pomiarem wibracji i fuzja sensoryczna	4
1.4.2	Sztuczna inteligencja w wibrodiagnostyce	6
1.4.3	Aspekt wdrożeniowy	8
1.4.4	Wykazanie innowacyjności	8
1.5	Teza pracy	9
2	Architektura oprogramowania	12
2.1	Wizualizacja wyników	13
2.2	Architektura rozwiązania on-premise	13
2.2.1	Microsoft IoT Edge	15
2.2.2	Pliki wykonywalne	16
2.3	Architektura rozwiązania chmurowego	18
2.4	Porównanie zastosowanych podejść	18
2.4.1	Analiza kosztów	18

2.4.2	Kompatybilność algorytmów, ekosystemu oraz ich aktualizacja	21
2.4.3	Magazyn danych	22
2.4.4	Wymagania sprzętowe i ograniczenia związane z systemem pre- dykcji awarii	23
2.5	Opis autorskich bibliotek	24
3	Predykcja awarii - algorytmy	27
3.1	Wdrożenia produkcyjne	27
3.1.1	Detekcja anomalii	28
3.1.2	Analiza trendu	28
3.2	Badania w PONAR Wadowice S.A.	29
3.2.1	Układ pomiarowy	32
3.2.2	Klasyfikacja	33
3.2.3	Predykcja	69
4	Zakończenie	87
4.1	Przejrzystość wypracowanych hipotez	88
4.2	Jakość działania algorytmów a koszt aparatury pomiarowej	88
4.3	Przenaszalność algorytmów i dalsze badania	89
4.4	Podsumowanie	90
	Bibliografia	107

Spis rysunków

2.1	Architektura rozwiązania on-premise i chmurowego	19
2.2	Architektura Scheduler-Dispatcher-AzureFunction	21
3.1	Uszkodzone łożysko wału (górne) - wytarcie wewnętrznej bieżni wyczuwalne palcem.	30
3.2	Uszkodzona kołyska - zniekształcenie powierzchni wyczuwalne palcem.	31
3.3	Schemat układu pomiarowego do badań pompy wielotłoczkowej.	34
3.4	Porównanie wartości VRMS dla obu czujników zamontowanych na pompie z rozróżnieniem na rodzaj uszkodzenia i wartość obciążenia stałowartościowego.	36
3.5	Fragment drzewa wygenerowanego dla klasyfikatora związanego z czujnikiem zamontowanym promieniowo i obciążeniem 10%. Rzeczywista głębina drzewa jest kilkukrotnie większa, zbędna na potrzeby wizualizacji	40
3.6	Pełne drzewo wygenerowane dla klasyfikatora wytrenowanego na wszystkich dostępnych zmiennych dla obciążenia 10%	41
3.7	Charakterystyka ROC dla klasyfikatora związanego z czujnikiem zamontowanym promieniowo i obciążeniem 10%. Klasyfikator nie jest idealny, należy zdecydować się na kompromis pomiędzy TPR a FPR.	43

3.8	Charakterystyka ROC dla klasyfikatora wytrenowanego na wszystkich dostępnych zmiennych dla obciążenia 10%. Klasyfikator klasyfikuje idealnie.	44
3.9	Macierz pomyłek dla klasyfikatora typu regresja wieloraka dla obciążenia stałego równego 10%. Klasyfikator został wytrenowany na danych pochodzących wyłącznie z czujnika drgań zamontowanego promieniowo.	49
3.10	Macierz pomyłek dla klasyfikatora typu regresja wieloraka dla obciążenia stałego równego 10%. Klasyfikator został wytrenowany wszystkich dostępnych zmiennych mierzonych.	50
3.11	Spektrogram prędkości drgań dla pełnego eksperymentu pomiarowego pompy z uszkodzonymi łożyskami. Zakres częstotliwości został przycięty na potrzeby lepszej wizualizacji.	51
3.12	Wysokopoziomowy schemat przetwarzania spektrogramów w celu ekstrakcji i wizualizacji cech oraz klasyfikacji.	52
3.13	Schemat generacji i przetwarzania spektrogramów na potrzeby treningu modeli z wykorzystaniem danych syntetycznych.	54
3.14	Różnica wartości F1 pomiędzy siecią dotrenowaną w części klasyfikującej, a siecią dotrenowaną w całości. Zestawienie dla wszystkich modeli związanych ze wszystkimi dostępnymi zakresami częstotliwości.	59
3.15	Schemat przetwarzania spektrogramu na punkt na płaszczyźnie 2D.	61
3.16	Naniesione punkty po ekstrakcji cech dla obciążeń 10...100%. Wektor wejściowy do algorytmu kompresji pochodzi z sieci wytrenowanej wyłącznie na danych syntetycznych.	63
3.17	Naniesione punkty po ekstrakcji cech dla obciążenia 60%. Wektor wejściowy do algorytmu kompresji pochodzi z sieci wytrenowanej wyłącznie na danych syntetycznych.	64

3.18	Naniesione punkty po ekstrakcji cech dla obciążeń 10...100%. Wektor wejściowy do algorytmu kompresji pochodzi z sieci dotrenowanej na danych rzeczywistych.	65
3.19	Naniesione punkty po ekstrakcji cech dla obciążenia 60%. Wektor wejściowy do algorytmu kompresji pochodzi z sieci dotrenowanej na danych rzeczywistych.	66
3.20	Przetwarzanie punktu z płaszczyzny 2D na wektor pewności klasyfikacji. Schemat jest bardzo zbliżony do rysunku 3.15 natomiast z racji tego, że prezentuje transformację odwrotną, należy analizować go od prawej do lewej strony.	71
3.21	Schemat obliczania metryk służących ocenie jakości transformacji odwrotnej. Im grubsza strzałka, tym większa jest wymiarowość danych, które przezeń przechodzą.	73
3.22	Mapy konturowe wektorów pewności klasyfikacji z podziałem na klasy. Wykreślone na podstawie wartości wektorów pewności uzyskanych z sieci CNN.	75
3.23	Mapy konturowe wygenerowane z wykorzystaniem modelu regresji liniowej z podziałem na stan pompy.	77
3.24	Proces syntezy modelu regresji na potrzeby uzyskania przestrzeni 3D.	79
3.25	Przebieg przykładowej trajektorii punktu pracy na płaszczyźnie 2D. Od punktu związanego z pompą sprawną do punktu związanego z pompą z uszkodzoną kołyską. Trajektorię oznaczono kolorem czerwonym.	82
3.26	Zależność zmiany wartości RUL w miarę zmiany położenia punktu pracy na płaszczyźnie 2D poprzez sukcesywne tranzycje. Założono T równe 5 lat.	83

3.27	Proces syntezy trajektorii przemieszczania punktu pracy oraz szacowania wartości RUL z wykorzystaniem sztucznie wygenerowanych punktów pracy.	86
------	---	----

Spis tabel

3.1	Etapy eksperymentu pomiarowego	32
3.2	Porównanie drzew decyzyjnych wytrenowanych wyłącznie na estymatach drgań pochodzących z czujnika zamontowanego promieniowo	38
3.3	Porównanie drzew decyzyjnych wytrenowanych estymatach drgań pochodzących z obu czujników drgań	38
3.4	Porównanie drzew decyzyjnych wytrenowanych estymatach drgań oraz zmiennych procesowych	38
3.5	Porównanie klasyfikatorów typu regresja logistyczna wytrenowanych na danych pochodzących z czujnika drgań zamontowanego promieniowo.	39
3.6	Porównanie klasyfikatorów typu regresja logistyczna wytrenowanych estymatach drgań z obu czujników.	42
3.7	Porównanie klasyfikatorów typu regresja logistyczna wytrenowanych na estymatach drgań oraz zmiennych procesowych.	42
3.8	Porównanie drzew decyzyjnych wytrenowanych wyłącznie na estymatach drgań pochodzących z czujnika zamontowanego promieniowo dla klasyfikacji wieloklasowej.	45
3.9	Porównanie drzew decyzyjnych wytrenowanych estymatach drgań pochodzących z obu czujników drgań dla klasyfikacji wieloklasowej.	45
3.10	Porównanie drzew decyzyjnych wytrenowanych estymatach drgań oraz zmiennych procesowych dla klasyfikacji wieloklasowej.	46

3.11	Porównanie klasyfikatorów typu regresja logistyczna wytrenowanych na danych pochodzących z czujnika drgań zamontowanego promieniowo dla klasyfikacji wieloklasowej.	46
3.12	Porównanie klasyfikatorów typu regresja logistyczna wytrenowanych na estymatach drgań z obu czujników dla klasyfikacji wieloklasowej.	46
3.13	Porównanie klasyfikatorów typu regresja logistyczna wytrenowanych na estymatach drgań oraz zmiennych procesowych dla klasyfikacji wieloklasowej.	47
3.14	Architektura sieci CNN	55
3.15	Ewaluacja wybranych modeli klasyfikujących syntetyczne spektrogramy na podstawie wskazanych zakresów częstotliwości	55
3.16	Zestawienie liczebności klas spektrogramów wyekstrahowanych z eksperymentu pomiarowego dla obciążenia stałego.	56
3.17	Metryki kilku modeli klasyfikacji spektrogramów wyekstrahowanych z eksperymentu pomiarowego dla obciążenia stałego. Sieci dotrenowane w części klasyfikującej.	57
3.18	Metryki kilku modeli klasyfikacji spektrogramów wyekstrahowanych z eksperymentu pomiarowego dla obciążenia stałego. Sieci dotrenowane w całości.	58
3.19	Porównanie wartości F1 dla dotrenowanych sieci oraz różnicy pomiędzy wartościami F1 dla modeli z analogicznego zakresu częstotliwości między sobą.	58
3.20	Porównanie modeli wytrenowanych na danych z czujnika wibracji zamontowanego promieniowo	60

3.21	Porównanie jakości działania algorytmów KNN na danych rzeczywistych. Dane z warstwy flatten pochodzą z sieci wytrenowanej wyłącznie na danych syntetycznych. Wybrano kilka cechujących się najwyższą wartością metryki balanced accuracy	62
3.22	Porównanie jakości działania algorytmów KNN na danych rzeczywistych. Dane z warstwy flatten pochodzą z sieci dotrenowanej z wykorzystaniem rzeczywistych spektrogramów. Wybrano kilka cechujących się najwyższą wartością metryki balanced accuracy	62
3.23	Zestawienie liczebności zbioru danych z podziałem na charakter przeprowadzonego eksperymentu.	68
3.24	Zestawienie wartości służących ocenie jakości transformacji odwrotnej punktu z płaszczyzny 2D na wektor pewności klasyfikacji.	72
3.25	Zestawienie jakości dopasowania przestrzeni 3D dla modelu typu regresja liniowa.	76
3.26	Pula parametrów spośród których dokonano wyboru najlepszego zestawu na potrzeby dopasowania przestrzeni 3D.	78
3.27	Porównanie najlepszych modeli MLP z wyszczególnieniem ich architektury.	80
3.28	Architektura sieci MLP o najlepszej jakości dopasowania do zbioru testowego.	80

Rozdział 1

Wprowadzenie

Przemysł jest z jednej strony branżą wrażliwą na modyfikacje, z drugiej zaś poszukującą innowacji. Modernizacja infrastruktury informatycznej oraz urządzeń biorących udział w produkcji ma na celu obniżenie kosztów wytwarzania produktów, lub zwiększenie wydajności procesu technologicznego. W działania te wpisuje się koncepcja Industry 4.0, a w nią z kolei idea Internet Of Things (IoT) [45]. Wspomniane koncepcje często wykorzystywane są w połączeniu z trendem utrzymującym wysokie zainteresowanie na przestrzeni ostatnich lat - sztuczną inteligencją. Takie połączenie umożliwia efektywne wykorzystanie danych w celu wypracowywania hipotez analitycznych zapewniających przedsiębiorstwu oszczędności. Danymi wsadowymi do algorytmów i modeli przetwarzających dane mogą być wszelakie dane procesowe dostępne dzięki wykorzystaniu IoT. Mogą być to parametry elektryczne maszyn biorących udział w procesie produkcyjnym, takie jak wartości skuteczne prądów i napięć. Innym rodzajem mierzonych zmiennych mogą być wielkości charakterystyczne dla konkretnego procesu produkcyjnego. Szczególnym przypadkiem wykorzystania tego rodzaju pomiaru jest monitoring temperatury i ciśnienia cieczy płynącej w układzie hydraulicznym agregatu wysokociśnieniowego [60]. Wszystkie wymienione wielkości mogą mieć charakter diagnostyczny. Podobnie jest z pomiarem wibracji, dla którego przykładowo wartość skuteczna prędkości drgań daje wymierne informacje o tym jaki jest stan monitorowanego urządzenia. Użyteczność pomiarów wibracji w przemyśle została wskazana w [94] oraz w [59]. Fuzja sztucznej inteligencji, pomiarów procesowych oraz danych wibracyjnych umożliwia syntezę rozwiązań przynoszących przedsiębiorstwu wymierne korzyści [58].

Takie podejście cechuje możliwość stworzenia nowych obszarów badawczych, bogatych w potencjał do tworzenia i wdrażania innowacji.

1.1 Predykcyjne utrzymanie ruchu

Obszarem do tworzenia innowacji w ramach niniejszej pracy jest rozwój algorytmów jednej ze strategii utrzymania ruchu. Mianowicie predykcyjnego (ang. Predictive Maintenance - PdM), które z definicji zakłada przewidywanie kiedy i jakiego rodzaju awaria wystąpi na monitorowanych elementach infrastruktury. W pracy [145] zaproponowano inne sposoby planowania utrzymania ruchu.

1.2 Urządzenia brzegowe

Implementacja rozwiązań IoT nie jest możliwa bez wykorzystania urządzeń brzegowych. Są to urządzenia niezbędne do właściwej akwizycji pomiarów, a także te zdolne do ich przetwarzania bez konieczności wysyłania ich do chmury, czy też innego scentralizowanego systemu gromadzącego dane [79]. Dane przetwarzane są niejako na "brzegu" chmury obliczeniowej. Ma to na celu między innymi ograniczenie przesyłu danych oraz poprawę szybkości działania implementowanych systemów [130]. Na potrzeby wdrożenia systemów będących przedmiotem niniejszej pracy zastosowano urządzenia takie jak:

1. Analogowe moduły pomiarowe prądów i napięć
2. Konwertery portów szeregowych
3. Moduły pomiarowe wibracji
4. Komputery brzegowe

Nie zostały wykorzystane inteligentne czujniki. Przetwarzanie danych odbywało się na komputerze brzegowym lub w chmurze. Ma to swoje biznesowe uzasadnienie, We wdrożeniach wykorzystany został autorski moduł pomiarowy o nazwie EXT/FFT służący do akwizycji wibracji. Konstrukcyjnie założono w nim bezpośredni pomiar przyspieszenia drgań oraz obliczanie szeregu wskaźników wibracyjnych. Podczas wdrożeń zawsze w pakiecie sprzedawano klientowi czujniki wibracji oraz EXT/FFT. Dzięki czemu zyskano większą elastyczność w wykorzystaniu pomiaru drgań. Szerzej urządzenie EXT/FFT zostało omówione w rozdziale 2.

1.3 Chmura obliczeniowa i język programowania

Podczas realizacji wdrożeń opisanych w niniejszej pracy w wielu miejscach posłużono się chmurą obliczeniową. Z jej wykorzystaniem przeprowadzone zostały wdrożenia oraz szereg działań mających na celu wykazanie słuszności (ang. Proof of Concept - PoC) wykorzystania mechanizmów chmurowych na potrzeby realizacji systemów predykcyjnego utrzymania ruchu. Wykorzystano chmurę Microsoft Azure [18] z tego względu, że przedsiębiorstwo w którym realizowany był doktorat korzysta z tejże chmury. Nie dokonywano porównań chmury Azure z innymi chmurami obliczeniowymi np. AWS czy GCP.

Implementacji wszelkich autorskich programów, modeli, algorytmów, bibliotek oraz wykorzystania bibliotek open-source dokonano wykorzystując język programowania Python. Do wizualizacji danych w formie wykresów wykorzystaną bibliotekę seaborn [135] oraz matplotlib [75].

1.4 Tematyka pracy a dotychczasowy stan wiedzy

Wibrodiagnostyka jest dziedziną rozwijaną i wykorzystywaną na szeroką skalę w przemyśle od lat [86]. W literaturze napotkać można publikacje związane z przewidywaniem awarii różnego rodzaju urządzeń oraz komponentów maszyn. Są to między innymi silniki [100], przekładnie zębate [85], turbiny [66], pompy [121] oraz łożyska [120]. Niniejsza praca jest poświęcona głównie klasyfikacji i predykcji awarii pompy wielotłoczkowej. Dotychczas klasyfikację i predykcję uszkodzeń pompy wielotłoczkowej w oparciu o analizę wibrodiagnostyczną z wykorzystaniem sztucznej inteligencji przeprowadzono między innymi w publikacjach [46], [125], [48]. Niniejsza praca wykorzystuje podejścia, algorytmy i przetwarzanie sygnału drgań w sposób zbliżony do opisanych w tych publikacjach. Jednakże zaproponowane w literaturze koncepcje zostały rozszerzone i zebrane w ramach jednego algorytmu, co opisano w rozdziałach 3.2.2 oraz 3.2.3.

W ramach niniejszej pracy poruszono szereg aspektów związanych z wibrodiagnostyką, które już wcześniej zostały w szerokim zakresie zbadane i opisane. Niniejsza rozprawa ten zakres poszerza i wzbogaca o nowe spojrzenie na klasyfikację i predykcję awarii pomp wielotłoczkowych. Innowacyjne elementy pracy opisano w podrozdziale 1.4.4.

1.4.1 Ekstrakcja cech związanych z pomiarem wibracji i fuzja sensoryczna

Pomiar wibracji najczęściej związany jest z bezpośrednim pomiarem przyspieszenia drgań. Następnie pośrednio poprzez całkowanie otrzymywany jest przebieg prędkości drgań [92]. Analiza wibrodiagnostyczna w literaturze przeprowadzana jest na szereg sposobów poprzez ekstrakcję różnego rodzaju informacji o badanym sygnale wykorzystując do tego rozbudowany aparat matematyczny. Między innymi w ramach badań nad ekstrakcją cech z sygnału drgań [102] wykorzystywano:

- Analizę statystyczną
- Analizę widmową
- Sztuczną inteligencję

Analiza statystyczna obejmuje obliczanie szeregu wskaźników związanych z analizowanymi szeregami czasowymi. Te wskaźniki to między innymi wartość skuteczna sygnału, wartość skuteczna w zakresie częstotliwości 10Hz-1kHz oraz wartość peak-to-peak co wyszczególniono w [92] oraz [68]. W praktyce przemysłowej najczęściej wykorzystywana jest wartość skuteczna prędkości drgań. Wskaźnik ten niesie ze sobą wiele informacji, a jego interpretacja w zależności od rozmiaru monitorowanej maszyny i warunków jej pracy została opisana między innymi w normie ISO 10816-1 [92]. Niniejsza praca również wykorzystuje wartości wymienionych wyżej wskaźników wibracyjnych jako podstawę do syntezy modeli klasyfikacji awarii pomp wielotłoczkowych. Szczegóły opisano w podrozdziale 3.2.2.

Badanie sygnału w dziedzinie częstotliwości dokonuje się często z wykorzystaniem widma częstotliwości prędkości drgań tak jak opisano w [96] [128]. Dla przykładu, w artykule [103] opisano sposób klasyfikacji konkretnych uszkodzeń przekładni planetarnej w oparciu o widmo amplitudowe prędkości drgań. Synteza i analiza spektrogramów prędkości drgań również należy do popularnych technik wykorzystywanych na potrzeby klasyfikacji uszkodzeń urządzeń drgających co ukazano w publikacjach [139] oraz [133].

Ekstrakcja bardzo złożonych cech sygnału odbywać się może z wykorzystaniem sztucznej inteligencji [113]. Sztuczna inteligencja jest pomocna w momencie, gdy posiadamy wiele wskaźników i charakterystyk opisujących badany sygnał, a ręczne wyszukiwanie najistotniejszych cech byłoby trudne i czasochłonne [143]. Kolejnym przykładem

ekstrakcji cech z wykorzystaniem sztucznej inteligencji jest badanie spektrogramów na potrzeby klasyfikacji uszkodzeń w parze z konwolucyjnymi sieciami neuronowymi [107]. W ramach niniejszej pracy również posłużono się spektrogramami na potrzeby klasyfikacji i detekcji uszkodzeń. Do ich przetwarzania również wykorzystano konwolucyjne sieci neuronowe. Natomiast oprócz podobieństw wskazać można też szereg różnic między niniejszą pracą a publikacją [107]. Inna jest architektura sieci CNN. Publikacja [107] wykorzystuje architekturę VGG16, a spektrogramy podawane na wejście sieci cechują się większym rozmiarem oraz są interpretowane jako obrazy RGB. Niniejsza rozprawa zaś korzysta z sieci CNN o architekturze mniej złożonej (o mniejszej liczbie warstw) niż VGG16, zaś spektrogramy wejściowe są interpretowane jako obrazy monochromatyczne. Sposób syntezy spektrogramów we wspomnianej publikacji przebiega na zasadzie przeprowadzenia krótkoczasowej transformaty Fouriera (STFT), znany jest przebieg sygnału drgań w horyzoncie czasu całego eksperymentu pomiarowego. Zaś w niniejszej rozprawie spektrogramy powstają z połączenia wyznaczanych co sekundę widm amplitudowych prędkości drgań na podstawie bufora 4096 próbek w urządzeniu brzegowym. Nie jest znany przebieg sygnału drgań, dostępne jest jedynie widmo powstałe z pewnego wycinka czasu. W publikacji [107] przeprowadzono analizę widmową w szerszym spektrum częstotliwości niż w niniejszej pracy. Niniejsza praca w ramach analizy spektrogramów przez sieć CNN zakłada wykorzystanie pierwszych 64 binów częstotliwości zawartych w spektrogramie. W rozprawie rozpatrzono jeden rodzaj uszkodzenia łożyska, zaś w pracy [107] rozpatrywanych było siedem typów uszkodzeń łożyska. Więcej informacji dotyczących wykorzystania sieci neuronowych typu CNN w połączeniu ze spektrogramami w ramach niniejszej pracy znajduje się w podrozdziałach 1.4.2 oraz 3.2.2.

Oprócz sygnału związanego z pomiarem drgań, dysponując innymi mierzonymi zmiennymi procesowymi, można znacząco podnieść jakość klasyfikacji uszkodzeń monitorowanego urządzenia jak ukazano w [32]. Dobrym przykładem zastosowania tego rodzaju podejścia jest publikacja [36], w której to dokonano diagnostyki turbogeneratorów małej elektrowni wodnej. W podrozdziale 3.2.2 zaprezentowano wykorzystanie uzyskanych cech sygnału wibracji oraz ich fuzji z pomiarami innych zmiennych procesowych na potrzeby klasyfikacji uszkodzeń pompy wielotłoczkowej. Ponadto wykorzystując regresję liniową oraz drzewa decyzyjne dokonano selekcji cech i wskazano te najbardziej użyteczne w kontekście klasyfikacji uszkodzeń.

1.4.2 Sztuczna inteligencja w wibrodiagnostyce

W oparciu o wskaźniki związane z sygnałem wibracji, rozwinięto w literaturze algorytmy sztucznej inteligencji [147]. Algorytmy te mają za zadanie wyciąganie wniosków z danych, które dla człowieka nie są widoczne z uwagi na ich złożoność, jak opisano w podrozdziale 1.4.1. Badania związane z klasyfikacją i detekcją uszkodzeń urządzeń przemysłowych w ramach wykorzystania sztucznej inteligencji obejmują między innymi wykorzystanie algorytmów uczenia maszynowego i uczenia głębokiego.

Uczenie maszynowe

W publikacji [33] dokonano przeglądu szeregu algorytmów uczenia maszynowego w ramach detekcji uszkodzeń silników indukcyjnych z wykorzystaniem sygnału drgań. Algorytmem znajdującym swoje zastosowanie w detekcji uszkodzeń łożysk jest maszyna wektorów nośnych [114]. Inne algorytmy z zakresu uczenia maszynowego takie jak regresja logistyczna [31], drzewa decyzyjne [34] oraz algorytm KNN [126] również są stosowane podczas klasyfikacji uszkodzeń urządzeń przemysłowych. W niniejszej pracy podrozdział 3.2.2 opisuje wykorzystanie regresji logistycznej oraz drzew decyzyjnych na potrzeby klasyfikacji uszkodzeń pompy wielotłoczkowej. Algorytm KNN został wykorzystany w pracy podczas oceny przejrzystości płaszczyzny 2D służącej do prezentacji rozmieszczenia punktów pracy pompy wielotłoczkowej z podziałem na rodzaj jej uszkodzenia.

Na potrzeby predykcji wartości obranego wskaźnika informującego o stanie monitorowanego urządzenia, zastosować można algorytmy regresji. Jednym z takich wskaźników jest RUL (ang. Remaining Useful Life). Jest to pozostały czas bezawaryjnej pracy urządzenia. Algorytmem stosowanym w predykcji wartości RUL jest między innymi regresja liniowa. Jej zastosowanie w tym kontekście zostało opisane w publikacji [61]. Rozdział 3.2.3 opisuje wykorzystanie właśnie regresji liniowej w ramach predykcji czasu pozostałego czasu bezawaryjnego użytkowania pompy wielotłoczkowej.

Uczenie głębokie

W ramach połączenia uczenia głębokiego oraz analizy widmowej wypracowane zostały algorytmy między innymi zaprezentowane w [115] i [141]. Wykorzystują one konwolucyjne sieci neuronowe (CNN) w połączeniu z cechami związanymi z sygnałem

drgań. Praca [42] jest najbardziej podobna w swej tematyce do niniejszej rozprawy pod kątem wykorzystania CNN. Rozprawa obejmuje jednak więcej rodzajów uszkodzeń niż jedynie badanie łożysk. Inny był preprocessing spektrogramów mający na celu ich przygotowanie w taki sposób, żeby uzyskać kompromis pomiędzy użytecznością spektrogramów, a ich rozmiarem. Motywacją takiego postępowania jest to, że rozmiar danych wejściowych do sieci neuronowej znacząco wpływa na czas treningu, a podczas korzystania z niej po wytrenowaniu zużywa więcej zasobów [112]. Bezpośrednio wiąże się to z architekturą sieci CNN. W ramach pracy rozważono jedną architekturę. Warto zwrócić uwagę na fakt, że liczba stopni swobody podczas projektowania sieci neuronowej jest na tyle duża, że na potrzeby dalszych badań warto byłoby zbadać wpływ architektury sieci CNN na uzyskane wyniki. Jednakże jak ukazano w [122] proces odnalezienia architektury sieci neuronowej dającej dobre wyniki jest zadaniem nietrywialnym. Klasyfikacja z wykorzystaniem CNN w niniejszej pracy była podstawą do syntezy algorytmu predykcji pozostałego czasu bezawaryjnego działania maszyny oraz prezentacji wyekstrahowanych przez sieć CNN cech na płaszczyźnie 2D. Rzutowania wielowymiarowych cech na płaszczyznę 2D dokonywano już w literaturze [144], jednak nie w kontekście spektrogramów związanych z prędkością drgań wykorzystując CNN jako ekstraktor cech.

Cechy pochodzące z płaszczyzny 2D po zastosowaniu kilku transformacji opisanych w podrozdziale 3.2.3 posłużyły do syntezy modelu sieci typu MLP. Model ten miał za zadanie jak najwierniejsze odwzorowanie wartości pewności klasyfikacji uzyskanej przez sieć CNN. Dzięki czemu cechy z płaszczyzny 2D wraz z predykcjami sieci MLP tworzą przestrzeń 3D. Przyjęto założenie, że w owej przestrzeni znajdować się mogą kolejne punkty pracy (konkretniej stany pomiędzy pompą uszkodzoną a sprawną) badanej pompy. Sieci typu MLP są szeroko wykorzystywane do zadań regresji w różnych dziedzinach [91], [137], [72]. Natomiast w literaturze nie przedstawiono procesu połączenia wartości pewności klasyfikacji sieci CNN jako zmiennej wyjściowej z sieci MLP w kontekście wykorzystania cech z płaszczyzny 2D powstałych po zmniejszeniu wymiarowości cech wyekstrahowanych przez sieć CNN.

Transfer wiedzy

Niniejsza praca poprzez autorski algorytm predykcji awarii korzysta z transferu wiedzy wskazanego w podrozdziale 3.2.2. Mechanizm ten szeroko opisany i wykorzystany został na potrzeby uczenia i wdrażania sieci neuronowych w różnych domenach bizne-

sowych [134] [90]. W ramach rozprawy doktorskiej dokonano syntezy spektrogramów wygenerowanych ze sztucznych przebiegów czasowych. Następnie nauczone wstępnie na syntetycznych danych sieci douczono korzystając z danych rzeczywistych. W literaturze proces generacji danych syntetycznych na potrzeby wstępnego nauczania sieci neuronowej typu CNN został szerzej i ogólniej opisany w artykule [65]. Wykorzystanie transferu wiedzy w kontekście wibrodiagnostyki przeprowadzono z kolei w publikacji [55]. Natomiast połączenie generacji syntetycznych spektrogramów, transferu wiedzy i sieci CNN w ramach niniejszej pracy jest nowatorskie.

1.4.3 Aspekt wdrożeniowy

W rozprawie położono duży nacisk na kwestię wdrożeń. Podjęto starania, by zastosowane w ramach pracy podejścia miały realną szansę być wdrożone komercyjnie. W związku z tym, w pracy przeanalizowano jaki zakres danych potrzebny jest, aby spełnić wymagania stawiane systemowi detekcji awarii. Co za tym idzie, wyciągnięto wnioski które czujniki i w jakim zakresie należy zakupić aby uzyskać pożądany efekt. W literaturze podobne analizy mające na celu kontekst związany z kosztami układu pomiarowego można znaleźć w [104].

Na rynku dostępnych jest szereg systemów służących do wibrodiagnostyki urządzeń przemysłowych. Przykładem jest MOLOS.ICM [13], system do monitoringu online firmy REDNT S.A. Do popularnych systemów należą takie, które związane są z przeprowadzaniem okresowych przeglądów elementów infrastruktury przemysłowej [95]. W wielu przypadkach jednak systemy monitoringu nie są nastawione na przejrzystość prezentacji użytkownikowi końcowemu, z uwagi na to, że przeglądu dokonuje diagnosta oceniający stan urządzenia i dostarczający rekomendacje dotyczące serwisów [81]. Z kolei niniejsza rozprawa z racji tego, że była w dużej mierze tworzona podczas pracy w komercyjnym środowisku przemysłowym, skoncentrowana jest w dużej mierze na prezentacji wyników działania algorytmów w sposób zrozumiały i intuicyjny dla odbiorcy.

1.4.4 Wykazanie innowacyjności

Wspomniane w rozdziale 1.4 publikacje oraz inne publikacje z dziedziny rozpatrywanej w ramach rozprawy, nie zakładają wizualizacji wyekstrahowanych cech związanych z wibracjami monitorowanego urządzenia w sposób przejrzysty dla użytkownika bez wiedzy domenowej z zakresu analizy danych. Nie są w nich przeprowadzane predykcje

wartości RUL z wykorzystaniem rzutowania na płaszczyznę 2D punktów pracy związanych ze spektrogramami wraz z wartością pewności klasyfikacji uszkodzenia uzyskanej z sieci klasyfikującej typu CNN.

Połączenie szeregu konceptów w jeden spójny algorytm przetwarzania spektrogramów związanych z prędkością drgań pompy wielotłoczkowej, to jest:

- Synteza sztucznych spektrogramów
- Trening sieci CNN
- Transfer wiedzy i dotrenowania CNN na spektrogramach rzeczywistych
- Rzutowanie wielowymiarowych, abstrakcyjnych cech uzyskanych za pomocą CNN na płaszczyznę 2D
- Synteza trajektorii przejścia pomiędzy punktami pracy związanymi z pompą sprawną, a pompą uszkodzoną
- Wyznaczenie związanej z trajektorią krzywej pewności klasyfikacji, którą to pewność uzyskano z sieci CNN.
- Szacowanie wartości RUL na podstawie krzywej pewności klasyfikacji

jest wartością dodaną jaką niesie ze sobą niniejsza praca.

1.5 Teza pracy

W ramach niniejszej rozprawy postawiono następującą tezę:

”Widmo prędkości drgań pozyskane z odpowiednio umiejscowionego czujnika drgań jest wystarczające, aby poprawnie klasyfikować i w przejrzysty sposób prezentować stan pompy z rozróżnieniem kilku rodzajów uszkodzeń i wartości obciążenia”.

W przemyśle, jak i w innych gałęziach biznesu, krytyczną kwestią jest sprawa kosztów, które należy ponieść w ramach prac prowadzonych podczas różnych projektów. Z punktu widzenia zarówno wykonawcy projektu jak i jego odbiorcy (klienta) istotne jest aby koszty te były jak najmniejsze. Wykonawca będzie mógł dzięki temu więcej zarobić, a odbiorca mniej zapłacić. Na potrzeby prac związanych z akwizycją i analizą danych w celu optymalizacji kosztów dążyć można do ograniczania liczby czujników i urządzeń

pomiarowych [73]. Mniejsza liczba czujników częstokroć powoduje okrojenie zestawu dostępnych zmiennych mierzonych na tyle, że prowadzenie efektywnej analizy danych staje się trudne, co ukazano w podrozdziale 3.2.2. Oprócz liczby czujników należy również zwrócić uwagę na ich rozmieszczenie. W przypadku czujników wibracji powstało szereg opracowań zawierających praktyczne wskazówki pomocne w kwestii sposobu i umiejscowienia czujników. Spośród najnowszych wyróżnić można prace [116] oraz [74]. Niniejsza praca w podrozdziałach 3.2.2 i 2.4.1 rozpatruje pod kątem kosztów szereg aspektów związanych z akwizycją i analizą danych w systemach służących do wibrodiagnostyki, włączając w to analizę związaną z doбором liczebności i rozmieszczeniem czujników wibracji.

Niniejsza praca zawiera kilka opisów wdrożeń przemysłowych. Natomiast główną jej częścią jest rozdział 3.2, w którym opisano badania laboratoryjne przeprowadzone na pompie wielotłoczkowej. Pompę tę na potrzeby badań uszkodzono na kilka sposobów oraz poddano kilku rodzajom obciążeń o charakterystyce stałowartościowej. Dane zebrane podczas badań stanowią podstawę analiz prowadzących do udowodnienia postawionej tezy. Dokonano akwizycji pomiarów związanych z wibracjami, oraz innymi wielkościami procesowymi mierzonymi podczas badań na pompie wielotłoczkowej. Szczegółowy opis stanowiska pomiarowego znajduje się w podrozdziale 3.2.1. W podrozdziale 3.2.2 przeprowadzono rozważania dotyczące wpływu liczby i rodzaju mierzonych zmiennych na jakość klasyfikacji badanej pompy.

Jednym z wielu sposobów analizy wibracji jest wyznaczanie widma prędkości drgań. Może być ono pomocne w klasyfikacji i detekcji uszkodzeń urządzeń przemysłowych [107]. Zdecydowano aby dużą część niniejszej pracy poświęcić analizie widma drgań z tego względu, że urządzenie do pomiaru wibracji opracowane w firmie w której realizowany był doktorat posiada taką właśnie funkcjonalność. Podrozdział 3.2.2 zawiera szczegółowy opis syntezy i wykorzystania widma drgań na potrzeby klasyfikacji uszkodzeń. W podrozdziale 3.2.2 dokonano analizy jakości klasyfikacji z wykorzystaniem widma drgań pozyskanego z odpowiednio umiejscowionego czujnika drgań. Wybór miejsca montażu czujnika z którego pozyskiwane jest widmo został umotywowany w podrozdziale 3.2.2.

Z praktyki przemysłowej pozyskanej w ramach prac wdrożeniowych prowadzonych podczas syntezy niniejszej rozprawy wynika, że sposób prezentacji wyników ma niebagatelne znaczenie dla pozyskiwania nowych projektów oraz pozytywnego odbioru tych już zaimplementowanych. Eksperti po stronie klienta posiadający wiedzę domenową

związaną z konkretną maszyną nie posiadają na ogół umiejętności z zakresu złożonej analizy danych. Dlatego we wdrożeniach komercyjnych istotny jest aspekt prezentacji stanu monitorowanego urządzenia. Z racji tego, że mogą one pracować pod różnym obciążeniem, wymagane jest uszczegółowienie prezentacji mając na uwadze wartość obciążenia. W niniejszej pracy podjęto próbę syntezy algorytmu umożliwiającego przejrzystą prezentację stanu pompy z rozróżnieniem kilku rodzajów uszkodzeń i wartości obciążenia.

Jakość klasyfikacji uszkodzeń definiować można na wiele sposobów. W niniejszej pracy określono ją poprzez poprawność klasyfikacji mierzoną z wykorzystaniem stosowanych metryk, co opisano w podrozdziale 3.2.2. Dokonano również analizy zasadności zastosowanych metryk z punktu widzenia klienta w zależności od komercyjnego przeznaczenia monitorowanego urządzenia, co zaprezentowano w podrozdziale 3.2.2. Na całokształt jakości ukazanego w niniejszej pracy podejścia do klasyfikacji uszkodzeń składa się również jakość prezentacji stanu pompy z podziałem na uszkodzenia. Do jej oceny wykorzystano stosowną metrykę, co opisano w podrozdziale 3.2.2.

Biorąc pod uwagę powyższe rozważania można podjąć próbę opracowania sposobu akwizycji i analizy danych prowadzącego do redukcji kosztów aparatury pomiarowej przy jednoczesnym zapewnieniu wystarczająco dobrego poziomu poprawności i przejrzystości rozwiązania z punktu widzenia klienta.

Rozdział 2

Architektura oprogramowania

W zależności od zastosowania, systemy PdM zostały wdrożone w chmurze lub on-premise. Podyktowane jest to tym, że nie zawsze klient dla którego wdrażane jest rozwiązanie godzi się na przechowywanie danych w chmurze. Co więcej, nie zawsze serwer na którym realizowany jest system predykcji awarii ma dostęp do internetu co znacząco wpływa na aspekt aktualizacji oprogramowania. Zrealizowano również wdrożenie hybrydowe. Urządzenie brzegowe umieszczono wtedy w infrastrukturze klienta, ale miało ono dostęp do internetu. Natomiast wypracowanie części hipotez analitycznych odbywało się na urządzeniu brzegowym.

Koncepcyjnie, niezależnie od wymagań projektowych w ramach wdrożeń systemu predykcji awarii wskazać można następujące komponenty funkcjonalne:

1. Wizualizacja wyników
2. Kod udostępniający hipotezy
3. Kod wypracowujący hipotezy
4. Kod pobierający dane z magazynu danych
5. Magazyn danych

Wszystkie one składają się na swoisty ekosystem systemu predykcji awarii. W zależności od warunków wdrożenia poszczególne komponenty będą miały nieco inną formę i możliwości.

Podczas wdrożeń dołożono starań, aby niezależnie od warunków wdrożenia zapewnić spójność algorytmów predykcji awarii po stronie chmury obliczeniowej oraz

w rozwiązaniach on-premise. Większa spójność algorytmów oznacza łatwiejsze rozwiązywanie problemów związanych z wypracowanym oprogramowaniem, gdyż nie ma konieczności utrzymywania dwóch niezależnych wersji algorytmów. Dyskusja związana ze spójnością wersji algorytmów pomiędzy wdrożeniami znajduje się w rozdziale 2.4.2.

2.1 Wizualizacja wyników

Najistotniejsze z punktu widzenia klienta końcowego jest to, aby mógł korzystać z wypracowanych hipotez analitycznych. W tym celu należy zapewnić intuicyjną i przejrzystą wizualizację - system SCADA. Tu również zależnie od potrzeb użytkownika końcowego możemy wyróżnić systemy posadowione w chmurze oraz on-premise w infrastrukturze klienta.

Realizacja on-premise zakłada wykorzystanie dostępnych narzędzi firm trzecich służących do syntezy systemów SCADA. Przykładami takich narzędzi są ASIX, iFIX, TIAPortal i inne [14]. Interfejsem komunikacyjnym pomiędzy programem do wypracowywania hipotez, a wizualizacją jest protokół Modbus.

Wersja chmurowa wymaga internetowej wersji systemu SCADA. W przypadku implementowanych systemów jest to aplikacja MOLOS.CLOUD [13]. Poprawną komunikację na linii algorytm predykcji awarii - SCADA zapewnia standard komunikacyjny wypracowany w ramach MOLOS.CLOUD. Oparty jest on na przesyłaniu komunikatów w formacie JSON.

2.2 Architektura rozwiązania on-premise

Zrealizowane zostały dwa podejścia realizacji systemu predykcji awarii posadowione na urządzeniu brzegowym. Jedno zakłada dostęp urządzenia brzegowego do internetu. W drugim zaś należy się liczyć z tym, że instalacja, konfiguracja oraz aktualizacja oprogramowania do predykcji awarii odbywać się może jedynie na obiekcie. Co więcej liczba wizyt na obiekcie jest ograniczona do minimum.

Komponenty funkcjonalne wskazane w liście 2 dla rozwiązania on-premise ulegają konkretyzacji poprzez wykorzystanie następujących programów.

1. Wizualizacja wyników - SCADA (Asix, TIAPortal)

2. Kod udostępniający hipotezy - ModbusSink (autorska aplikacja)
3. Kod wypracowujący hipotezy - RULonPrem (autorska aplikacja)
4. Kod pobierający dane z magazynu danych - ModbusArchiver (autorska aplikacja)
5. Magazyn danych (estymaty wibracji) - InfluxDB [3]
6. Magazyn danych, widma drgań - SpectraArchiver (autorska aplikacja)

Wyżej wymienione autorskie aplikacje oraz magazyn danych zostaną poniżej szczegółowo opisane.

ModbusArchiver to usługa korzystająca z protokołu Modbus [53], działająca jako Modbus Master. Cyklicznie odpytuje ona wskazanych Modbus Slave'ów o dane wibracyjne i procesowe. Następnie zapisuje pobrane wartości do bazy danych.

Archiwum, w którym przechowywane są dane jest darmowa baza danych InfluxDB. Jest to baza NoSQL, której przeznaczeniem jest przechowywanie pomiarów typu klucz:wartość. Baza bardzo dobrze nadaje się do przechowywania danych z szeregów czasowych [21].

Oprócz magazynowania danych, składowane są również widma drgań. Archiwizacja realizowana jest poprzez narzędzie netcat [10], przechwytyjący strumień bajtów z urządzenia REDNT EXT/FFT realizującego wyznaczanie widma. Przechwycony strumień bajtów jest następnie kompresowany przy pomocy programu 7z [9]. Harmonogramowanie akwizycji i kompresji widma zaimplementowano jako usługę. Skompresowane widma trafiają do chmury z wykorzystaniem autorskiej biblioteki *blob-downloader*. W ramach systemu Linux wykorzystano systemd [12], natomiast w wersji na system Windows wykorzystano nssm [11]. Na schemacie 2.1 opisany komponent opisany jest jako SpectraArchiver.

Najważniejszym z punktu widzenia algorytmiki jest usługa RULonPrem. Korzysta ona z autorskiej biblioteki *rul* przechowywanej na prywatnym repozytorium paczek Pythona - MOLOS.ANALYTICS. Usługa umożliwia pobieranie danych wibracyjnych z magazynu danych, przetwarzanie danych, wypracowywanie hipotez oraz udostępnianie ich na potrzeby wizualizacji wyników na przykład do systemu SCADA.

ModbusSink jest serwerem TCP nasłuchującym na połączenia przychodzące i przyjmującym poprzez socket ramki w formacie JSON. Owe ramki obudowane są w struktury

umożliwiająca swoistą walidację klientów TCP wysyłających dane. Takie zabezpieczenie jest konieczne z uwagi na to, że inne aplikacje lub usługi mogłyby podłączyć się do aplikacji ModbusSink jako klient. Przychodzące ramki JSON są przetwarzane w taki sposób, aby trafiły w odpowiednie miejsca mapy Modbus udostępnianej przez aplikację. Stamtąd mogą być odczytywane na przykład przez systemy SCADA, w których odbywa się wizualizacja hipotez.

Warte uwagi jest zastosowanie w architekturze sprzężenia zwrotnego. ModbusArchiver oprócz urządzeń udostępniających dane procesowe odpytuje również usługę ModbusSink. Z niej to pozyskuje wskaźniki analityczne związane z wypracowanymi hipotezami analitycznymi. Dzięki czemu wyniki działania algorytmów predykcji awarii zostają zarchiwizowane. Archiwum może zostać poddane analizie na potrzeby dalszych badań.

Wszystkie zaimplementowane narzędzia opisane w niniejszej sekcji w celu zapewnienia stabilności działania ekosystemu RULonPrem jako całości zostały uruchomione jako usługi.

Schemat ekosystemu on-premise przedstawiono na rysunku 2.1.

2.2.1 Microsoft IoT Edge

Microsoft Azure IoT Edge to framework oparty na kontenerach Dockera [19]. Umożliwia proste zarządzanie kontenerami, a co za tym idzie łatwą aktualizację oprogramowania na urządzeniu brzegowym. Budowa kontenerów odbywać się może na dowolnym komputerze klasy PC. Korzystanie z kontenerów na różnych urządzeniach wymaga ich składowania w ogólnodostępnym repozytorium kontenerów. Przykładem takich repozytoriów kontenerów są DockerHub [16] i Azure Container Registry [15].

Często zachodzi konieczność uruchamiania aplikacji na różnych systemach operacyjnych, a nawet na różnych architekturach sprzętowych. Konteneryzacja sprawia, że rozwój aplikacji można uniezależnić od systemu operacyjnego. Zatem ten sam kod niewielkim kosztem można uruchomić na komputerze klasy PC z systemem Windows lub Linux, czy też na komputerze brzegowym w architekturze armhf (np. komputer AVTSE-RPI-IIOTG [1] lub MOXA UC8100 [2]) z systemem Linux lub Windows. Rozwój oprogramowania przeprowadza się niemal bez obaw o problemy jakie można napotkać przy przenoszeniu aplikacji pomiędzy systemami operacyjnymi lub architekturami sprzętowymi. Szczegółowe wymagania sprzętowe związane z frameworkiem IoT Edge wskazane są w [19] oraz w sekcji 2.4.4.

W kontenerze Dockera zawrzeć można niemal dowolny fragment kodu programu w dowolnie wybranym przez siebie języku programowania. Kontenery mogą również przechowywać pliki. Wykorzystując tę funkcjonalność na potrzeby systemu predykcji awarii w kontenerze zawrzeć można plik z modelem realizującym funkcjonalność predykcyjną. Mogą być to pliki binarne wypracowane z wykorzystaniem popularnych bibliotek do uczenia maszynowego i uczenia głębokiego. W szczególności tego rodzaju plikami są:

1. Pliki *.joblib, utworzone z wykorzystaniem biblioteki scikit-learn [41]
2. Pliki *.h5 lub *.ckpt stworzone wykorzystując bibliotekę Tensorflow [29].

W kodzie programu kontenera należy zapewnić poprawne przyjęcie i dostosowanie danych wejściowych do modelu, tak aby mógł on wypracować predykcję. Dane wejściowe muszą zostać odczytane z modułów pomiarowych, a następnie poddane standaryzacji kompatybilnej z potokiem przetwarzania wypracowanym podczas syntezy modeli, które z tych danych będą korzystać. Ponadto, wypracowane hipotezy należy w odpowiedni sposób udostępnić do kolejnego elementu ekosystemu - wizualizacji.

Więcej szczegółów dotyczących technologii IoT Edge, zalet i wad płynących z jej wdrożenia oraz możliwości implementacji systemów do akwizycji danych przedstawiono w [57] oraz [56].

Technologia mimo dużego potencjału została odrzucona w kolejnych wdrożeniach komercyjnych. Powodem były problemy ze stabilnością oprogramowania na tanich komputerach brzegowych o słabych parametrach sprzętowych. Z biznesowego punktu widzenia nieopłacalna była elastyczność, którą daje IoT Edge w konfrontacji ze środkami jakie należało wydać na komputer brzegowy o odpowiednich parametrach.

2.2.2 Pliki wykonywalne

Wymaganą funkcjonalność można realizować bez konieczności wykorzystania kontenerów i frameworku IoT Edge. Co więcej, jak zostało wyżej wspomniane, niektóre projekty komercyjne z racji swojej specyfiki związanej z dostępem do internetu i chmury obliczeniowej muszą być realizowane bez technologii chmurowych.

Poszczególne elementy funkcjonalne architektury nie różnią się od tych, które działają w ramach tych opisanych w rozdziale 2.2.1. Tym razem jednak, zamiast zamykać kod w kontenery Dockera, które opisano w podrozdziale 2.2.1, wykorzystano pliki wyko-

nywalne. Język programowania Python to język interpretowany w którym zaimplementowane zostały autorskie komponenty systemu. Działanie skryptów wymagałoby zatem interpretera zainstalowanego na komputerze, na którym są one uruchamiane. Sam interpreter nie wystarcza. Komponenty napisane w języku Python wymagają szeregu bibliotek-paczek, które pobierane są z repozytoriów paczek. Publicznego (PyPI [23]) oraz prywatnego (MOLOS.ANALYTICS) będącego częścią strzeżonego know-how. Pobieranie paczek wymaga dostępu do internetu. Jego brak oznacza konieczność wykorzystania mechanizmu, który zawierać w sobie będzie zarówno interpreter Pythona jak i wszystkie potrzebne biblioteki-paczki. Rozwiązaniem jest PyInstaller [17], który do jednego pliku wykonywalnego (*.exe dla systemu Windows, *.bin dla systemu Linux) jest w stanie zawrzeć interpreter Pythona, skrypt do wykonania oraz wszystkie potrzebne biblioteki. Zatem przy konieczności aktualizacji oprogramowania wystarczy podmienić plik wykonywalny. Bez obaw o to, że wystąpi problem z instalacją biblioteki czy interpreterem jako takim.

2.3 Architektura rozwiązania chmurowego

System predykcji awarii posadowiony w chmurze również cechuje się swoistym ekosystemem w ramach którego funkcjonują algorytmy.

Elementy ekosystemu predykcji awarii dla rozwiązania chmurowego są następujące:

1. Wizualizacja wyników - MOLOS.CLOUD
2. Kod udostępniający hipotezy - Azure Function + Azure IoT Hub
3. Kod wypracowujący hipotezy - Azure Function + autorska biblioteka *rul*
4. Kod pobierający dane z magazynu danych - Azure Function + autorska biblioteka *telemetry-extractor*
5. Magazyn danych - Azure Blob Storage [27]

Aby poprawnie wizualizować efekty działania systemu predykcji awarii należy wypracowane hipotezy w odpowiedni sposób i w odpowiednim formacie danych przesłać do ekosystemu MOLOS.CLOUD. Wykorzystane zostały biblioteki języka Python dostępne na licencji MIT na repozytorium paczek PyPI. Są to oficjalne biblioteki Microsoft służące do komunikacji dwustronnej z chmurą [26].

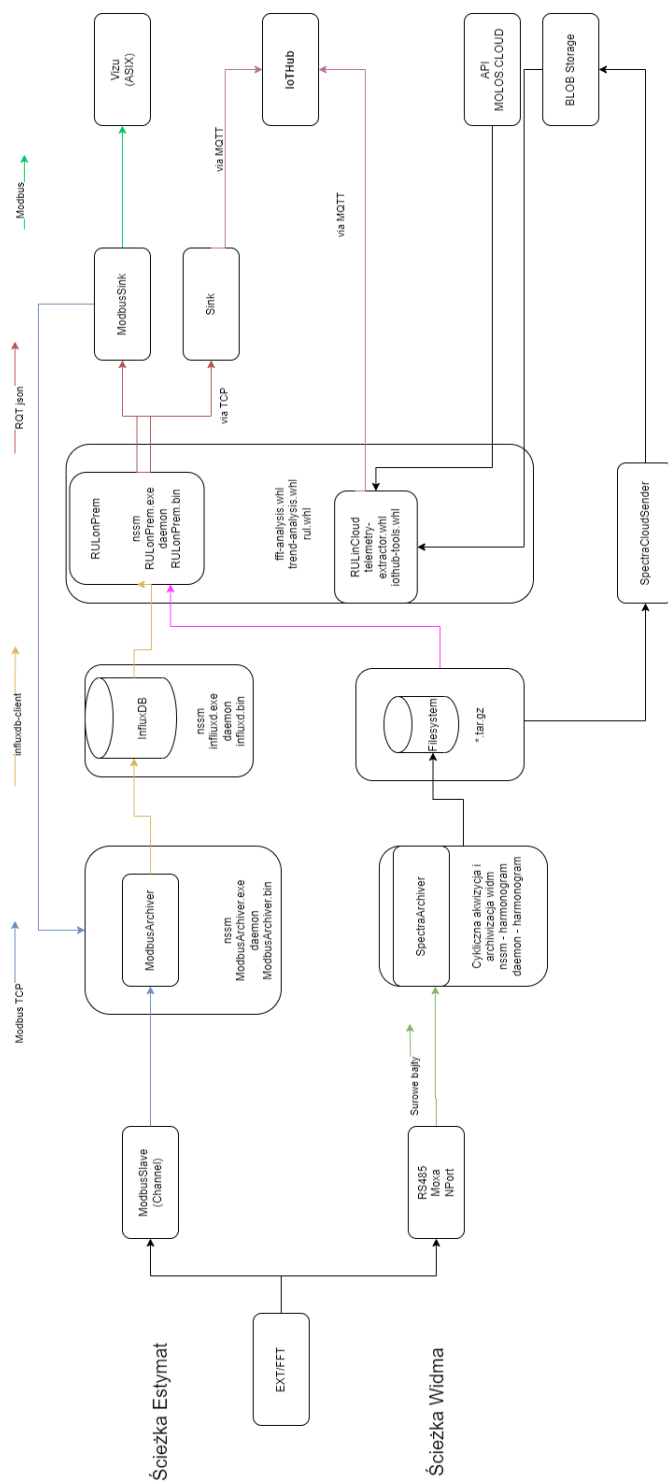
Architektura rozwiązania chmurowego schematycznie ukazana jest na rysunku 2.1. Konceptyjnie punkt centralny rozwiązania, autorskie biblioteki analityczne, pokrywają się z wersją on-premise.

2.4 Porównanie zastosowanych podejść

Zaprezentowane sposoby wdrożenia systemów różnią się między sobą. Zarówno na poziomie kalkulacji biznesowej oraz w warstwie implementacji.

2.4.1 Analiza kosztów

Rozwiązanie on-premise jest bardziej korzystne cenowo w ramach wykonywanych dotychczas projektów. Tańsze w kontekście infrastruktury sprzętowej z tego względu, że w ramach wszystkich implementowanych projektów i tak należy przewidzieć w kosztorysie serwery bazodanowe dla systemów SCADA oraz serwery na który zainstalowane



Rysunek 2.1: Architektura rozwiązania on-premise i chmurowego

mają być wizualizacje SCADA. Powodem takiego stanu rzeczy jest skala implementowanych rozwiązań. Dotychczas w ramach projektu monitorowanych było co najwyżej pięć urządzeń. W takim wypadku dedykowany serwer bazodanowy na potrzeby SCADA w zupełności wystarczy aby umieścić na nim system predykcji awarii. Więcej informacji związanych z wymaganiami sprzętowymi dla systemu predykcji awarii on-premise znajduje się w sekcji 2.4.4.

Implementacja chmurowa rozwiązania cechuje się większą skalowalnością i znacznie prostszym utrzymaniem. Narzędzia, które oferuje platforma Microsoft Azure oferują szybkie i w miarę intuicyjne mechanizmy pozwalające na zwiększenie liczby instancji wymaganych do analizy dużej liczby urządzeń. Wykorzystanie Function App oraz Azure Functions [25] podczas implementacji na wiele instancji odbywa się z poziomu przetwarzania serwerless autorskiego kodu napisanego w języku Python. Ów kod sekwencyjnie analizuje dane z monitorowanych urządzeń korzystając z autorskich bibliotek *rul*, *telemetry-extractor*, *fft-analysis* oraz *blob-downloader*. W swoim działaniu program jest tożsamy z wersją on-premise systemu predykcji awarii. Inne jest jedynie źródło danych będące przedmiotem analizy. Przetwarzanie oraz przygotowanie danych wraz z wypracowywaniem hipotez jest takie samo jak w wersji on-premise. Kwestię kompatybilności, różnic i podobieństw między programami i algorytmami systemu predykcji zostały omówione w rozdziale 2.4.2.

Azure Functions mają swoje ograniczenia w postaci czasu wykonywania kodu oraz zasobów pamięci RAM, pamięci dyskowej oraz CPU. Nie stanowią one natomiast przeszkody w dotychczasowych implementacjach. Natomiast w przypadku, w którym doszłoby do sytuacji wymagającej analizy takiej liczby urządzeń, że ograniczenia związane z Azure Functions okazałyby się istotne można zastosować podejście opisane w rozdziale 2.4.1.

Analiza dużej liczby urządzeń w chmurze

Zapewnienie stabilności działania mechanizmu umożliwiającego analizę urządzeń na dużą skalę wymaga zastosowania szeregu usług dostępnych z poziomu platformy Azure. W ramach wypracowanej architektury usługi te to:

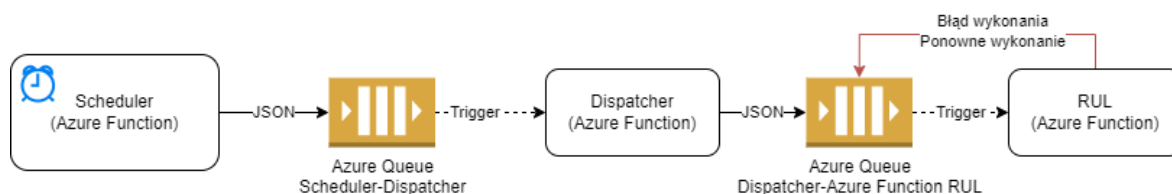
- Wyzwalana czasowo co godzinę Azure Function odpowiedzialna za sprawdzenie dla których urządzeń należy przeprowadzić analizę. Następnie Azure Function wysyła na Azure Queue wiadomość w formacie JSON, zawierającą informację

o tym które urządzenia wymagają wykonania analizy. Ten komponent to tzw. Scheduler.

- Wyzwalana z wykorzystaniem Azure Queue, Azure Function odpowiedzialna za przekazanie zlecenia analizy. Komponent ten to tzw. Dispatcher.
- Wyzwalana z wykorzystaniem Azure Queue odpowiedzialna za przeprowadzenie analizy dla konkretnego urządzenia. Opisany komponent to tzw. Azure Function RUL.

Pośredni komponent Dispatcher może wydawać się niepotrzebny. Scheduler również mógłby przekazywać stosowną wiadomość JSON na kolejkę, którą wyzwalana jest Azure Function RUL. Jednakże zdarza się, że Azure Function RUL zakończy się niepowodzeniem. Niekoniecznie w związku z błędną implementacją. Niestety wykorzystywane Azure Function App nie gwarantuje, że uda się uruchomić Azure Function RUL. Problem ten jest strictly właściwością komponentu Azure Function App. Nie jest on związany z wypracowanym kodem czy architekturą. Nowsze i droższe wersje Azure Function App cechują się większą stabilnością. Natomiast przedstawiona architektura niejako zabezpiecza przed niestabilnością Azure Function App.

Architektura nie została dotychczas przetestowana w ramach systemu predykcji awarii, natomiast w REDNT od lat działa stabilnie w ramach systemu generowania niemal tysiąca raportów produkcyjnych dla różnych urządzeń. Schematycznie została ona zaprezentowana na rysunku 2.2.



Rysunek 2.2: Architektura Scheduler-Dispatcher-AzureFunction

2.4.2 Kompatybilność algorytmów, ekosystemu oraz ich aktualizacja

Wdrożenia on-premise w wersji bez dostępu do internetu nie są w pełni kompatybilne w kwestii algorytmów z wdrożeniami hybrydowymi i chmurowymi. Niekompatybilne nie tylko porównując rozwiązanie on-premise z rozwiązaniem chmurowym. Brak

zgodności oprogramowania występuje również pomiędzy różnymi instancjami rozwiązań on-premise. Niespójność wynika z tego, że dostęp do infrastruktury klienta na której wdrożono rozwiązanie był mocno ograniczony lub wręcz niemożliwy. Czy to z uwagi na fizyczne położenie serwerów (na przykład pod ziemią w przypadku kopalń), czy też z uwagi na odległość do miejsca wdrożenia i konieczność długich wyjazdów służbowych. Aktualizacja oprogramowania na infrastrukturze klienta w większości przypadków był niemożliwa. Zatem nie można było przeprowadzić wzbogacenia rozwiązania o nowe funkcjonalności.

Inaczej kwestia spójności algorytmów prezentuje się w chmurze. Wykorzystanie zaimplementowanych w ramach doktoratu mechanizmów ciągłej integracji i ciągłego wdrażania (ang. CI/CD) przy użyciu Azure DevOps [20] zapewnia, że na wszystkich instancjach, na których wdrożony został system predykcji awarii działa ten sam kod. Naprawianie błędów i rozwijanie nowych funkcjonalności jest nieporównywalnie łatwiejsze i wygodniejsze niż w przypadku pracy w ramach systemu on-premise.

2.4.3 Magazyn danych

Zgromadzone dane są podstawą do wypracowania hipotez analitycznych. Należy zatem zadbać o to, aby ich przechowanie oraz tworzenie kopii zapasowych było stabilne i pewne. Nie bez znaczenia jest utrzymanie magazynu danych, które w dłuższym horyzoncie nie powinien być potrzebny, a jakiegokolwiek prace w trakcie wdrożenia powinny zajmować jak najmniej czasu. Dostęp do danych powinien być wystarczająco szybki i łatwy w implementacji.

Analizując stabilność działania oraz czas potrzebny na utrzymanie baz InfluxDB oraz Azure Storage można wysnuć wniosek, że magazyn chmurowy wypada pod tym względem korzystniej. Nie jest konieczne utrzymywanie infrastruktury serwerów, jej monitorowanie i dbanie o bezpieczeństwo. Niewielkim kosztem czasu można zapewnić redundancję danych. Nawet w takim wariacie, aby dane ulokowane były na serwerach w różnych miejscach świata co zapobiec może utracie danych w momencie fizycznego uszkodzenia jednej z serwerowni chmury Azure. Baza InfluxDB, jak każda baza on-premise, wymaga utrzymania pod kątem sieciowym oraz monitoringu jej stanu. Należy kontrolować tworzenie kopii zapasowych, przynajmniej przez pewien czas od wdrożenia. Trudne, a wręcz niemożliwe, jest zapewnienie redundancji danych w wielu lokalizacjach. Jak pokazuje praktyka wdrożeniowa, infrastruktura klienta jest hermetyczna. Nie ma

możliwości wykorzystania serwerów klienta w celu przechowywania kopii zapasowych magazynu danych. Z kolei z punktu widzenia kosztów ponoszonych przez wykonawcę projektu nie opłaca się zakupywać kolejnego serwera, a dodatkowo lokować go w innym miejscu niż pozostałe dotychczas postawione.

Warstwa dostępu do danych zarówno w wariancie chmurowym jak i on-premise jest wygodna w użytkowaniu. Oba rozwiązania posiadają stabilnie działające, na bieżąco rozwijane biblioteki języka Python na licencji MIT za pomocą których można manipulować danymi. Jednakże warto zauważyć, że wariant chmurowy systemu predykcji awarii jest częścią ekosystemu MOLOS.CLOUD. Dostęp do danych telemetrycznych nie odbywa się bezpośrednio z poziomu magazynu danych, a poprzez API. Z kolei dostęp do widm drgań zgromadzonych w BLOB Storage odbywa się bezpośrednio z magazynu. Nie ma pośrednich warstw dostępowych.

Dostęp do danych w obu analizowanych wersjach jest wystarczająco szybki. W ramach implementowanego systemu predykcji awarii nie ma konieczności dostępu do danych w ciągu kilku sekund. Aczkolwiek przedstawione magazyny danych umożliwiają tak szybki dostęp. Zupełnie wystarczające jest natomiast gdy dane zostaną pobrane w ciągu kilku minut.

2.4.4 Wymagania sprzętowe i ograniczenia związane z systemem predykcji awarii

Opisane wymagania sprzętowe zostały przedstawione z punktu widzenia wyizolowanego systemu predykcji awarii. Nie zostały uwzględnione zasoby potrzebne w celu wizualizacji wyników. Na dotychczas przeprowadzonych wdrożeniach średnie zużycie procesora w szczytowym momencie wyniosło 3% na jednym rdzeniu CPU o częstotliwości taktowania 4GHz. Z kolei wykorzystanie pamięci RAM wyniosło w szczytowym momencie średnio 60MB. Przestrzeń dyskowa zajmowana przez rozwiązanie on-premise obejmuje dane umiejscowione w bazie InfluxDB oraz archiwa z widmami drgań. Akwizycja danych na produkcyjnej instalacji z dwóch monitorowanych urządzeń przez 6 miesięcy skutkowałą zajęciem około 100MB przestrzeni dyskowej.

Można zatem stwierdzić, że zużycie zasobów jest niewielkie. Implementacja on-premise systemu predykcji awarii możliwa jest nawet na urządzeniach brzegowych o możliwościach takich jak MOXA-UC8100 (1GHz CPU, 512 MB RAM, karta SD 32GB). Jest ona również możliwa na serwerach na których posadowiona jest baza danych za-

wierająca pomiary z monitorowanych urządzeń lub innych serwerach w infrastrukturze klienta.

Rozwiązanie chmurowe również posiada pewne ograniczenia w ramach założeń przyjętych podczas implementacji. Podstawowym założeniem było racjonalne dysponowanie środkami finansowymi związanymi z utrzymaniem usług w ramach systemu. Dopiero w następnej kolejności należało rozważyć prostotę implementacji oraz utrzymania kodu źródłowego. Implikuje to konieczność wykorzystania najtańszych komponentów dostępnych z poziomu platformy Azure. Co z kolei pociąga za sobą szereg ograniczeń. Niekoniecznie natomiast są to istotne ograniczenia. Function App i Azure Function będące niejako w centrum całego ekosystemu narzucają ograniczenia w postaci czasu wykonywania. Wykonywany kod może działać najwyżej 10min i zużywać 1GB pamięci operacyjnej RAM. Magazyn chmurowy BLOB Storage przechowujący widma drgań w najtańszej opcji nie zapewnia dostępu do danych w nielimitowanej formie hot archive, a jedynie cold archive [28]. Natomiast z punktu widzenia logiki biznesowej predykcja awarii nie wymaga tak szybkiego dostępu do danych z uwagi na długi czas w którym w monitorowanym urządzeniu występuje poszerzające się uszkodzenie.

2.5 Opis autorskich bibliotek

Oprócz modeli klasyfikacji i predykcji awarii oraz w ramach pracy powstał zestaw autorskich bibliotek wspierających proces analizy danych. Biblioteki te, są częścią architektury oprogramowania opisanej w rozdziale 2. Zgodnie ze schematem 2.1 w ramach systemu predykcji awarii wyróżnić można szereg bibliotek.

- *telemetry-extractor*
- *blob-downloader*
- *iothub-tools*
- *fft-analysis*
- *trend-analysis*
- *rul*

telemetry-extractor jest biblioteką umożliwiającą pobieranie danych z magazynu Azure Table Storage poprzez asynchroniczne zapytania do MOLOS.CLOUD API. Jest

one częścią ekosystemu MOLOS.CLOUD do wizualizacji wyników związanych z analizą urządzeń. Ekosystem ten został szerzej opisany w rozdziale 2.1. Często wolumen danych wejściowych do modeli jest na tyle obszerny, że pobieranie danych z API wymaga zastosowania asynchroniczności, by umożliwić płynne działanie systemu predykcji awarii. Ponadto, w ramach wdrożenia niemal zawsze monitorowane jest więcej niż jedno urządzenie, co również uzasadnia konieczność stosowania równoległego przetwarzania. Implementacji asynchroniczności dokonano z wykorzystaniem biblioteki *asyncio* [22]. Dzięki temu możliwe jest tworzenie i równoległe wykonywanie potoków przetwarzania [140], w których skład wchodzi przygotowanie danych wejściowych do wykorzystywanych modeli oraz przesyłanie wypracowanych hipotez do dalszych części ekosystemu MOLOS.CLOUD. *telemetry-extractor* wykorzystywany jest jedynie w chmurowej implementacji systemu predykcji awarii.

Jak opisano w rozdziale 2.2 biblioteka *blob-downloader* wykorzystywana jest do wysyłania zgromadzonych widm drgań w postaci archiwów *.tar.gz do chmury na Azure BLOB Storage. Z kolei na potrzeby wypracowania hipotez związanych z widmem drgań biblioteka *blob-downloader* umożliwia pobieranie widma w postaci plików *.tar.gz z magazynu chmurowego. Udostępnia ona także przystępny interfejs programistyczny, dzięki któremu możliwe jest zarządzanie pobranymi widmami i ułatwienie dzięki temu wypracowywania hipotez diagnozujących uszkodzenia monitorowanych maszyn. Biblioteka stosowana jest podczas wdrożeń chmurowych i op-premise.

W celu wizualizacji wypracowanych w chmurze hipotez, konieczne jest ich przesłanie co chmury Azure do Azure IoT Hub. Stamtąd owe hipotezy trafiają do ekosystemu MOLOS CLOUD, gdzie są częścią wizualizacji zamówionej przez klienta. Możliwe jest to dzięki bibliotece *iothub-tools*, która zawiera asynchroniczną implementację wysyłania danych w odpowiednim formacie do Azure IoT Hub. Biblioteka korzysta z oficjalnej biblioteki służącej do interakcji z Azure IoT Hub. Utrzymywana jest ona przez firmę Microsoft [26]. *iothub-tools* wykorzystywana jest wyłącznie we wdrożeniach chmurowych.

Biblioteka *trend-analysis* służy jak sama nazwa wskazuje do analizy trendu wskazanej jednej lub wielu zmiennych procesowych, wskaźników oraz zmiennych związanych z wibracjami. Implementacja korzysta z regresji liniowej z biblioteki *scikit-learn* [41], która umożliwia obliczanie współczynników związanych z poszczególnymi zmiennymi po dopasowaniu krzywej regresji. W przypadku najczęściej wdrażanym, czyli regresji jednej zmiennej jako funkcja liniowa, szczególnie użyteczne jest śledzenie współczynni-

ka kierunkowego prostej regresji informującego o dynamice zmian badanego wskaźnika. Bibliotekę wykorzystano podczas wdrożeń, w których śledzono zmiany wartości VRMS w celu wypracowania wskaźnika RUL poprzez ekstrapolację trendu do wartości krytycznej, podobnie jak opisano to w publikacji [132]. Wdrożenia wykorzystujące *trend-analysis* opisano w rozdziale 3.1. Biblioteka jest częścią wdrożeń zarówno chmurowych jak i on-premise.

Analiza widma przeprowadzana jest w ramach implementacji biblioteki *fft-analysis*. Korzysta ona z opisanej powyżej biblioteki *blob-downloader*. Pliki z widmami przezeń wyekstrahowane mają postać surowych bajtów przechwyconych z portu szeregowego urządzenia EXT/FFT. Wymagają one przetworzenia aby móc je zinterpretować jako widma drgań. Plik z widmem zawiera szereg ramek, w których zawarte są widma drgań. Ponadto, każda ramka zawierająca widmo cechuje się określoną postacią sekwencji bajtów rozpoczynających i kończących ramkę. Obróbka ramek oraz ich konwersja na przystępne w wykorzystaniu obiekty języka Pythona odbywa się właśnie w bibliotece *fft-analysis*. Oprócz samego przygotowania widm wykorzystując pliki binarne biblioteka umożliwia również tworzenie wykresów widma, powstałych z widm spektrogramów oraz wyciągania podstawowych statystyk związanych z każdym z dostępnych widm. Przykładem takiej statystyki są wiodące prążki widma. Ponadto, biblioteka wykorzystując algorytm K-Means [35] pozwala na klasteryzację widm, co jest użyteczne w kontekście wykrywania reżimu pracy badanego urządzenia.

Punktem centralnym każdego wdrożonego systemu predykcji awarii jest biblioteka *rul*. Jest ona biblioteką nadrzędna względem wszystkich opisanych powyżej. Nadzoruje wykonanie i wywołanie metod zaimplementowanych w bibliotekach podrzędnych.

Wszystkie opisane biblioteki znajdują się na repozytorium paczek Pythona będącym własnością firmy, w której realizowany jest doktorat. Automatyczne budowanie, testowanie i publikacja paczek została przeprowadzona z wykorzystaniem Azure DevOps [20].

Rozdział 3

Predykcja awarii - algorytmy

We wdrożeniach produkcyjnych nie zaobserwowano uszkodzenia monitorowanych urządzeń. Nie zawsze można było dokonać oceny jakości predykcji i klasyfikacji awarii w sposób w pełni. Jednakże wypracowano i wdrożono produkcyjnie system predykcji i detekcji awarii korzystający z zaimplementowanych algorytmów opartych na założeniach dotyczących pracy urządzenia monitorowanego.

Przeprowadzono natomiast wdrożenie laboratoryjne we współpracy z firmą PONAR Wadowice S.A. dotyczące pomp wielotłoczkowych. Uzyskano dostęp do infrastruktury badawczej umożliwiającej w pierwszej kolejności wytrenowanie modeli detekcji awarii, a następnie ich ewaluację. Posłużono się narzędziami służącymi do klasyfikacji, aby wesprzeć proces predykcji uszkodzeń. W celu wypracowania wyników posłużono się zarówno pomiarami wibracji jak i pomiarami innych dostępnych zmiennych procesowych.

Wykorzystanie i trenowanie modeli na potrzeby konkretnej instalacji w ramach wdrożenia wymaga uwzględnienia różnych ograniczeń. Związane one są głównie z tym, jakiego rodzaju i ile czujników można zamontować na infrastrukturze klienta. Zależnie bowiem od projektu i przeznaczenia, monitorowane urządzenia mogą posiadać zupełnie inny zestaw dostępnych zmiennych pomiarowych.

3.1 Wdrożenia produkcyjne

Podczas wdrożeń produkcyjnych nie zaobserwowano degradacji ani uszkodzenia monitorowanych urządzeń. Brak możliwości precyzyjnego określenia rodzaju oraz momentu wystąpienia uszkodzenia bazując na danych historycznych implikuje konieczność

korzystania z narzędzi innych niż algorytmy uczenia nadzorowanego w celu ustalenia stanu urządzenia i pozostałego czasu jego poprawnej pracy. W ramach wdrożeń wykorzystano dwa podejścia dające klientowi końcowemu wartość w kontekście oceny stanu maszyn wchodzących w skład jego infrastruktury. Pierwsze związane jest z detekcją anomalii, drugie zaś z analizą trendu wskaźników związanych z dostępnymi pomiarami.

Architektura, implementacja i sposób posadowienia systemu detekcji i predykcji awarii zostały szczegółowo opisane w rozdziale 2

3.1.1 Detekcja anomalii

Algorytmy detekcji anomalii umożliwiają analizę stanu maszyny w sposób wielowymiarowy. Bardziej kompleksowy niż za pomocą śledzenia prostych, na przykład dwuwymiarowych, charakterystyk związanych z pracą urządzenia, choć i takie okazują się być pomocne [60]. W ramach jednego z przeprowadzonych wdrożeń zaimplementowano system detekcji anomalii wykorzystujący dane wibracyjne oraz inne pomiary procesowe.

Badane urządzenia będące częścią wdrożenia były fabrycznie nowe. Przyjęto założenie, że pomiary zebrane w ciągu pierwszych tygodni wdrożenia stanowią stan nominalnej pracy. Nauczony na takich danych model będzie w stanie orzec, czy dane procesowe zbierane z urządzenia wskazują na stan normalnej lub anomalnej pracy. Podczas wdrożenia wykorzystano model One-Class-SVM [111] zasilony zestawem cech pochodnych względem mierzonych zmiennych. Cechy te zostały wyznaczone z wykorzystaniem biblioteki *ts-fresh* [51]. Wyznaczony za pomocą modelu obszar normalnej pracy w wielowymiarowej przestrzeni został zrzutowany na płaszczyznę 2D z wykorzystaniem algorytmu PCA [127]. One-Class-SVM swoje działanie opiera na funkcji decyzyjnej. Dzięki czemu możliwa jest kontrola nad czułością algorytmu detekcji anomalii.

3.1.2 Analiza trendu

W pewnych przypadkach analiza wielowymiarowa może okazać nieadekwatna, wręcz nadmiarowa, względem potrzeb klienta. Budżet przeznaczony na wdrożenie systemu predykcji awarii może być wystarczający na przeprowadzenie prostej analizy polegają-

cej na śledzeniu trendu pewnych wskaźników i ich ekstrapolacji w czasie. Wtedy wyznaczony pozostały czas poprawnej pracy maszyny obliczany jest na bazie krytycznej wartości śledzonego wskaźnika. Ów wskaźnik może być pojedynczą zmienną procesową lub też być obliczany z wykorzystaniem większej liczby zmiennych. W literaturze związanej z wibrodiagnostyką napotkać można wskaźniki takie jak współczynnik szczytu oraz kurtoza przebiegu prędkości skutecznej drgań [101]. Wskaźniki te mogą być użyteczne w kontekście przewidywania uszkodzeń łożysk [136] i przekładni zębatych [138]. Istotą takiego podejścia jest prostota implementacji oraz to, że wskaźnik będący podstawą analizy jest łatwy w interpretacji. Dzięki temu nie tylko osoba odpowiedzialna za implementację, ale również klient będzie w stanie zrozumieć istotę działania systemu.

Wdrożenia w ramach których zbierane były dane wyłącznie wibracyjne w ramach predykcji awarii zakładały ekstrapolację trendu jednej zmiennej procesowej. Mianowicie prędkości skutecznej drgań (tzw. VRMS). Urządzenia monitorowane podczas wdrożeń omawianych w niniejszej rozdziale nie były nowe. Przyjęto natomiast założenie, że monitorowane urządzenia są w stanie dobrym. Wypracowano je w oparciu o wywiad przeprowadzony z obsługą, która posiadała wieloletnie doświadczenie w pracy z maszyną będącą przedmiotem monitoringu. Wartość VRMS dla działającego pod stałym obciążeniem urządzenia wynosiła $1.2 \frac{mm}{s}$. Zgodnie z założeniami opisanymi w normie ISO-10816 jako wartość krytyczną VRMS przyjęto dziesięciokrotnie większą [83] aniżeli zaobserwowana podczas pracy monitorowanej maszyny. Wartość krytyczna dla rozważnego wdrożenia wynosiła zatem $12 \frac{mm}{s}$. Zbierane przez kilka miesięcy pomiary podczas ciągłej pracy urządzenia poddane zostały analizie trendu. Prosta regresja liniowa najlepiej oddała charakter przebiegu czasowego VRMS. Wartość współczynnika kierunkowego prostej regresji wykazała, że trend wartości VRMS jest niezauważalnie wzrostowy. Co za tym idzie, obliczana wartość pozostałego czasu poprawnego użytkowania maszyny jest bardzo duża i przekracza kilka lat.

3.2 Badania w PONAR Wadowice S.A.

W warunkach laboratoryjnych korzystając z hali R&D firmy PONAR Wadowice S.A. przeprowadzono szereg badań mających na celu dostarczenie danych będących podstawą do wypracowania modeli klasyfikacji i predykcji uszkodzeń pomp wielotłoczkowych. Współpraca z ekspertami domenowymi ze strony firmy PONAR pozwoliła przeprowadzić badania w taki sposób, aby warunki pracy pompy jak najwierniej od-



Rysunek 3.1: Uszkodzone łożysko wału (górne) - wytarcie wewnętrznej bieżni wyczuwalne palcem.

wzorowywały to, co może mieć miejsce na rzeczywistej instalacji podczas wdrożenia komercyjnego. Pomiary przeprowadzane były na pompie wielotłoczkowej HSP-10V firmy Hydraul [7]. Została ona poddana następującym uszkodzeniom:

- Uszkodzone łożysko wału (górne)
- Uszkodzona kołyska
- Uszkodzona tarcza rozrządu
- Luz poosiowy



Rysunek 3.2: Uszkodzona kołyska - zniekształcenie powierzchni wyczuwalne palcem.

Scenariusz badań zawierał wiele rodzajów obciążeń. Został zaprojektowany tak, aby pompa poddana była takim obciążeniom jakim może być poddana podczas działania u klienta. Tabela 3.1 ukazuje kolejne etapy każdego przeprowadzonego eksperymentu pomiarowego.

Etap	Obciążenie	Okres	Czas trwania
Obciążenie stałe	10, 20...100%	x	3min każde
Trapez	30%-90%	180s	30min
Sinus	75%-95%	60s	15min
Sinus	10%-100%	30s	10min

Tabela 3.1: Etapy eksperymentu pomiarowego

Wartością zadawaną w czasie badań było obciążenie pompy momentem obrotowym silnika hamującego.

Przeprowadzono jeden eksperyment dla każdego rodzaju uszkodzenia pompy. Ponadto, dokonano trzech serii pomiarowych pompy sprawnej. Serie pomiarowe związane z reassemblingiem miały na celu zbadanie jak bardzo wpływa on na charakterystykę działania pompy. Modele wypracowane w rozdziale 3.2.2 związane były z jednego rodzaju obciążeniem. Oznacza to, że przykładowo nie należy stosować modelu wytrenowanego na danych pochodzących z obciążenia stałowartościowego o wartości 70% podczas pracy pompy z obciążeniem stałowartościowym o wartości 30%. W pracy w podrozdziale 3.2.2 dokonano natomiast syntezy modeli, które nie są dedykowane konkretnym wartością obciążenia stałego.

3.2.1 Układ pomiarowy

Na potrzeby badań zaprojektowano układ pomiarowy z myślą o tym, by był on możliwy do implementacji podczas wdrożeń produkcyjnych. Centralnym punktem akwizycji pomiarów był komputer brzegowy AVTSE-RPI-IIOTG [1]. Zaimplementowano na nim szereg programów autorstwa REDNT S.A. służących do odczytu danych z urządzeń pomiarowych oraz ich wysyłki do chmury Microsoft Azure. Do pomiaru wibracji posłużył moduł pomiarowy EXT/FFT firmy REDNT S.A. Urządzenie zaopatrzone jest w wejścia dedykowane do podłączenia czujników wibracji w standardzie IEPE. Na potrzeby badań, jak i na potrzeby dotychczasowych wdrożeń, wykorzystano dwa czujniki HS-150 firmy Hansford [4]. Jeden z czujników zamontowany był w osi obrotu pompy. Z kolei drugi czujnik zamontowany był promieniowo do osi obrotu pompy na łożysku

łączący pompę z napędzającym ją silnikiem. Akwizycja danych związanych z ciśnieniem i temperaturą medium tłoczonego przez pompę odbywała się z wykorzystaniem urządzenia EXT/UI autorstwa REDNT S.A. Do pomiaru tych zmiennych posłużyły czujniki firmy IFM [6], [5].

Schematycznie układ pomiarowy zaprezentowany jest na rysunku 3.3.

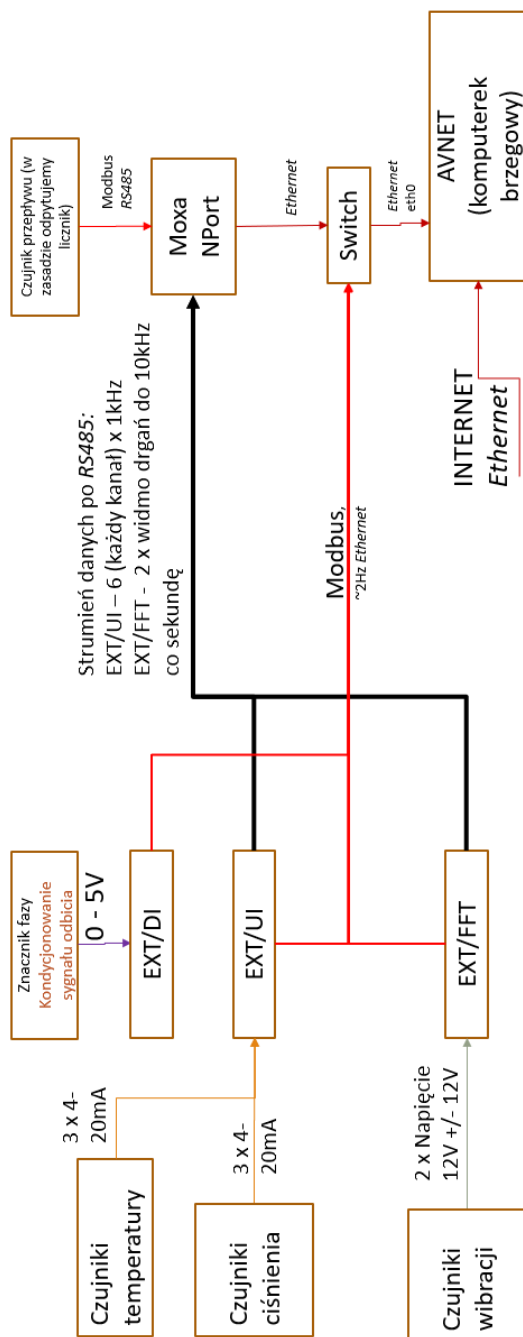
Po przeprowadzeniu pomiarów dane zostały poddane analizie w celu wypracowania modeli, których zadaniem jest klasyfikacja i predykcja uszkodzeń.

3.2.2 Klasyfikacja

Pozyskane dane były precyzyjnie opisane. Dokładnie wiadomo, która seria pomiarowa związana jest z konkretnym rodzajem uszkodzenia i etapem eksperymentu. W celu syntezy modeli klasyfikacji zastosowano dwa podejścia. Pierwsze z nich zakłada wykorzystanie wszystkich dostępnych mierzonych zmiennych oprócz widma drgań. Drugie zaś bazuje wyłącznie na widmie drgań. Takie rozróżnienie umotywowane jest tym, że znacząco odmienny jest stopień skomplikowania modeli w zależności od tego, jakiego rodzaju dane zastosowane są do ich treningu. Modele oparte o dane nieuwzględniające widma są bardziej intuicyjne i zrozumiałe dla odbiorcy. Rolę jaką odgrywają wartości temperatury, ciśnienia, a nawet prędkości skutecznej drgań w działaniu modeli prościej jest zrozumieć aniżeli to jakiego rodzaju informacje niesie ze sobą widmo drgań. Z drugiej strony, widmo drgań jest szeroko wykorzystywane w diagnostyce urządzeń wirujących co pokazano w [93] oraz [110]. Wypracowane modele i algorytmy oprócz jak najlepszej jakości klasyfikacji miały na celu również umożliwienie ukazania klientowi, które zmienne mają największy wpływ na wynik klasyfikacji. Stąd konieczność zbadania tego, w jaki sposób zakres analizowanych danych wpływa na jakość działania modeli.

Zmienne procesowe i estymaty wibracji

Niejednokrotnie w praktyce przemysłowej podczas wdrożeń montaż dodatkowych czujników wiąże się z problemami. Każdy dodatkowy czujnik to dodatkowy koszt. Co więcej, usytuowanie monitorowanej maszyny lub specyfika jej pracy uniemożliwia montaż konkretnego czujnika. Czujniki wibracji, zwłaszcza montowane na magnes, są łatwe i wygodne w montażu. Jednakże nie zawsze dane dotyczące wibracji są wystarczające aby wykonać poprawną klasyfikację. Dokonano analizy mającej na celu odpowiedź na



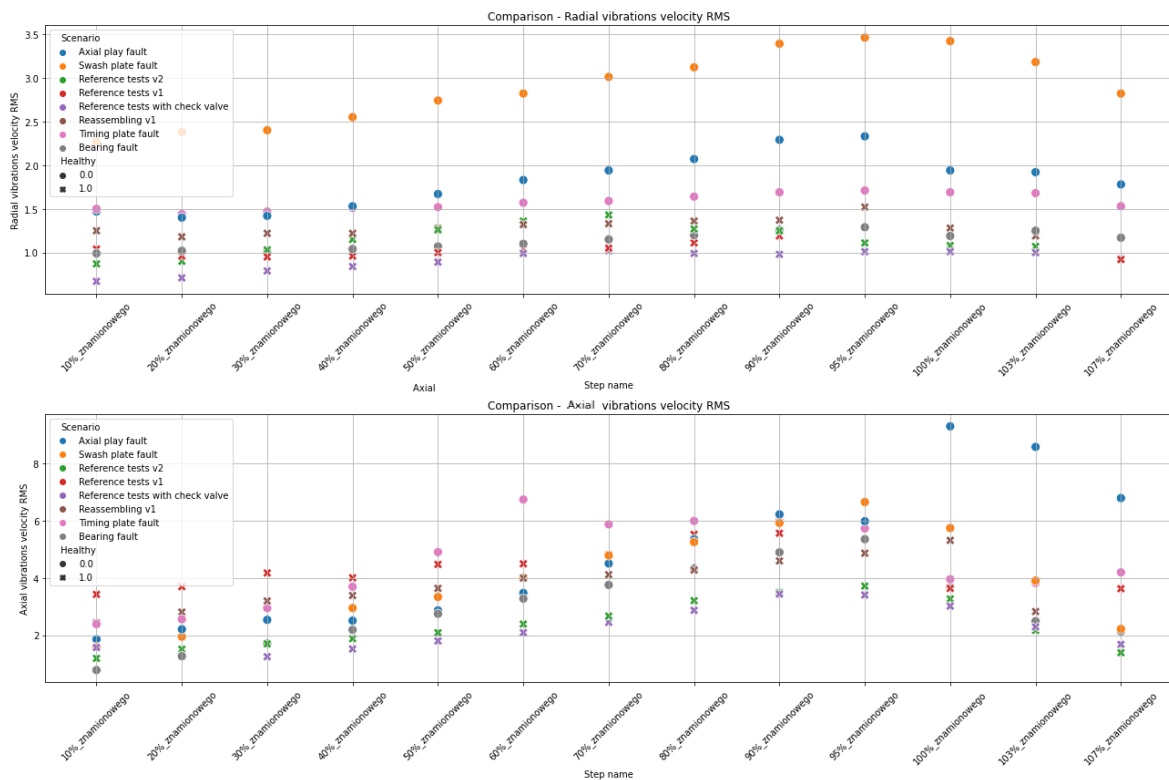
Rysunek 3.3: Schemat układu pomiarowego do badań pompy wielotłoczkowej.

pytanie ile i jakiego rodzaju czujniki są potrzebne aby zapewnić wystarczająco dobrą jakość klasyfikacji.

Analiza obciążenia stałego

Scenariusz zaprezentowany w tabeli 3.1 zakładał szereg obciążeń stałowartościowych zmienianych skokowo. Aby wyeliminować efekty związane z przebiegami niestabilnymi dokonano analizy tych części przebiegów, które występują po osiągnięciu stanu ustalonego. Jak pokazuje wykres 3.4 same wartości estymat VRMS mogą być wystarczające do stosunkowo łatwej diagnozy niektórych uszkodzeń. Szczególnie dobrze widać to w przypadku pomiarów z czujnika zamontowanego promieniowo dla uszkodzeń typu uszkodzona kołyska pompy, uszkodzona tarcza rozrządu oraz luz poosiowy. Jednakże oprócz wartości wskaźników dla uszkodzonej kołyski nie dla wszystkich wartości obciążenia łatwo jest na pierwszy rzut oka rozróżnić uszkodzenia oraz nawet to, czy pompa jest sprawna czy uszkodzona. W przypadku pomiarów pochodzących z czujnika zamontowanego osiowo wartość VRMS jest dużo mniej użytecznym wskaźnikiem pozwalającym rozróżnić uszkodzenia oraz odróżnić pompę sprawną od uszkodzonej. Niemal dla każdej wartości obciążenia niemożliwym jest określenie jaka wartość lub zakres VRMS związany jest z pompą sprawną, a jaki z pompą uszkodzoną. Warty uwagi jest również fakt, że reassembling pompy w przypadku czujnika zamontowanego osiowo tak istotnie wpływa na wartość wskaźnika VRMS, że rozrzut jego wartości uniemożliwia łatwą i przejrzystą analizę. W przypadku czujnika zamontowanego promieniowo wartości VRMS związane z reassemblingiem w zasadzie pokrywają się ze sobą. Na podstawie tych informacji można wysnuć wniosek, że bardziej użyteczne są pomiary z czujnika zamontowanego promieniowo nie tylko ze względu na łatwiejszą analizę wartości VRMS, ale również dlatego, że jego wskazania są mniej czułe na reassembling. Mogłoby się bowiem okazać, że bazując jedynie na wartości VRMS pochodzącej z czujnika zamontowanego osiowo może one być błędnie interpretowana jako uszkodzenie.

Rysunek 3.4 został oceniony przez klienta jako przejrzysty i łatwy w interpretacji. Jednakże informacje tam zawarte nie są w pełni wystarczające aby precyzyjnie i z dużą dozą pewności wskazać uszkodzenie pompy. W tym celu wytrenowano dwa rodzaje klasyfikatorów zdolnych wskazać pomiary mające największy wpływ na jakość klasyfikacji. Mianowicie drzewa decyzyjne oraz modele regresji logistycznej. Jako pierwsze wytrenowano modele wyłącznie na pomiarach pochodzących z czujnika zamontowanego osiowo, rozróżniający pompę sprawną od uszkodzonej. Jeden model



Rysunek 3.4: Porównanie wartości VRMS dla obu czujników zamontowanych na pompie z rozróżnieniem na rodzaj uszkodzenia i wartość obciążenia stałowartościowego.

per wartość obciążenia. Wpierw wytrenowano modele dokonujące klasyfikacji binarnej (pompa sprawna / uszkodzona). Następnie wytrenowano modele dokonujące klasyfikacji z podziałem na uszkodzenia. Porównano podejścia wykorzystujące wyłącznie dane wibracyjne oraz wykorzystujące fuzję danych wibracyjnych i procesowych. Modele wytrenowane na zmiennych procesowych oraz estymatach wibracji korzystały z danych próbkowanych co sekundę. Każdy z przykładów w zbiorze danych to niejako wektor danych opisujący sekundę pracy pompy. Wszystkie wytrenowane modele to de facto potoki przetwarzające dane. Zawierają one wstępne przygotowanie danych poprzez ich standaryzację. Warto zaznaczyć, że to w jaki sposób przebiega standaryzacja opracowywane jest na zbiorze treningowym. Nie ma zatem przecieku informacji związanego ze wstępnym przygotowaniem danych. Zbiór danych został podzielony na treningowy i testowy w proporcjach 75:25. Podział odbywał się w taki sposób, że dane były losowane w proporcjach odpowiadających liczebności poszczególnych klas w zbiorze.

W pierwszej kolejności zostaną omówione drzewa decyzyjne [39]. Jest to kompromis pomiędzy jakością klasyfikacji, a prezentacją hipotez wypracowanych przez model w sposób zrozumiały dla klienta. Zależnie od wartości obciążenia oraz rodzaju danych wejściowych struktura drzewa była bardziej lub mniej rozbudowana. Porównanie wytrenowanych modeli na bazie danych wibracyjnych z czujnika zamontowanego promieniowo przedstawia tabela 3.2. Wyniki z tabeli 3.3 Z kolei porównanie wytrenowanych modeli na bazie danych wibracyjnych oraz procesowych przedstawia tabela 3.4. Przyjęto metrykę F1 [129] w celu ewaluacji modeli. Jej wartość obliczana jest na podstawie zarówno czułości jak i precyzji klasyfikatora. Dzięki temu za pomocą jednej wartości można oszacować skłonność klasyfikatora do klasyfikacji false positives i false negatives [123]. Dyskusja dotycząca wyboru wskaźników jakości znajduje się w rozdziale 3.2.2.

Porównując dane z tabeli można wysnuć wniosek, że im większą liczbą zmiennych dysponujemy, tym lepsza jest jakość klasyfikacji. Największy zysk objawia się w momencie, gdy zamiast korzystać z jednego czujnika drgań wykorzystamy oba. W takim przypadku otrzymujemy bardzo dobrą jakość klasyfikacji. Natomiast w momencie gdy wykorzystamy pełne spektrum dostępnych pomiarów dla niemal każdej wartości badanego obciążenia otrzymujemy idealną jakość klasyfikacji. Jedynym wyjątkiem jest obciążenie 80%, dla którego jakość klasyfikacji nieco odbiega od idealnej. Kolejnym wnioskiem płynącym z przedstawionych wyników jest zmiana głębokości drzewa decyzyjnego. Im większą liczbą zmiennych dysponujemy, tym mniejsza jest głębokość drzewa. Związane jest to z tym, że drzewo decyzyjne dokonuje klasyfikacji poczynają-

Obciążenie	F1 zb. testowy	Głębokość	Najistotniejsze cechy
10%	0.84	23	VRMS promieniowo, VP2P promieniowo
20%	0.86	17	VRMS promieniowo, VP2P promieniowo
30%	0.88	12	VRMS promieniowo, VP2P promieniowo
40%	0.93	13	VRMS promieniowo, VP2P promieniowo
50%	0.91	12	VRMS promieniowo, VP2P promieniowo
60%	0.86	13	VRMS promieniowo, VP2P promieniowo
70%	0.9	17	VRMS promieniowo, VP2P promieniowo
80%	0.89	12	VRMS promieniowo, VP2P promieniowo
90%	0.93	17	VRMS promieniowo, VP2P promieniowo
100%	0.9	21	VRMS promieniowo, VP2P promieniowo

Tabela 3.2: Porównanie drzew decyzyjnych wytrenowanych wyłącznie na estymatach drgań pochodzących z czujnika zamontowanego promieniowo

Obciążenie	F1 zb. testowy	Głębokość	Najistotniejsze cechy
10%	0.97	6	VRMS promieniowo, VP2P osiowo
20%	0.99	6	VRMS promieniowo, VRMS osiowo
30%	0.98	6	VRMS promieniowo, VRMS osiowo
40%	0.99	5	VRMS promieniowo, VRMS osiowo
50%	0.98	4	VRMS promieniowo, VRMS osiowo
60%	1.0	6	VRMS promieniowo, VRMS osiowo
70%	0.98	6	VRMS promieniowo, VRMS osiowo
80%	0.95	11	VRMS promieniowo, VRMS osiowo
90%	0.97	12	VRMS promieniowo, VP2P promieniowo
100%	0.97	14	VRMS promieniowo, VRMS osiowo

Tabela 3.3: Porównanie drzew decyzyjnych wytrenowanych estymatach drgań pochodzących z obu czujników drgań

Obciążenie	F1 zb. testowy	Głębokość	Najistotniejsze cechy
10%	1.0	2	Temp. na przeciekach, Temp. wejściowa
20%	1.0	3	VRMS promieniowo, Temp. na wejściu
30%	1.0	3	VRMS promieniowo, Temp. na wejściu
40%	1.0	2	VRMS promieniowo, Temp. na wejściu
50%	1.0	3	VRMS promieniowo, Temp. na wejściu
60%	1.0	3	VRMS promieniowo, Temp. na wejściu
70%	1.0	3	VRMS promieniowo, Temp. na wejściu
80%	0.98	6	VRMS promieniowo, Temp. na wejściu
90%	1.0	4	VRMS promieniowo, Temp. na wejściu
100%	1.0	5	VRMS promieniowo, Ciśn. na wyjściu

Tabela 3.4: Porównanie drzew decyzyjnych wytrenowanych estymatach drgań oraz zmiennych procesowych

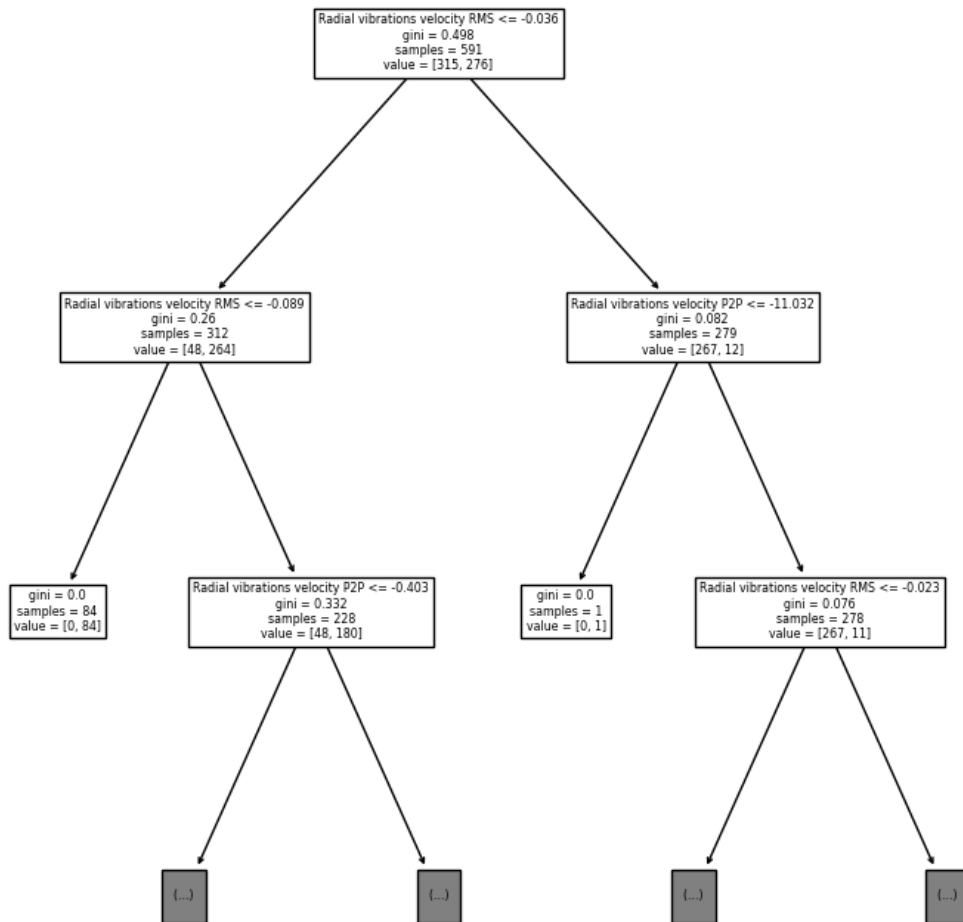
Obciążenie	F1 zb. testowy	AUC	Najistotniejsze cechy
10%	0.81	0.87	VRMS promieniowo, VP2P promieniowo
20%	0.87	0.96	VP2P promieniowo, VRMS promieniowo
30%	0.86	0.94	VP2P promieniowo, VRMS promieniowo
40%	0.86	0.95	VP2P promieniowo, VRMS promieniowo
50%	0.85	0.94	VP2P promieniowo, VRMS promieniowo
60%	0.87	0.93	VP2P promieniowo, VRMS promieniowo
70%	0.84	0.92	VP2P promieniowo, VRMS promieniowo
80%	0.9	0.96	VP2P promieniowo, VRMS promieniowo
90%	0.86	0.9	VRMS promieniowo, VP2P promieniowo
100%	0.89	0.97	VP2P promieniowo, VRMS promieniowo

Tabela 3.5: Porównanie klasyfikatorów typu regresja logistyczna wytrenowanych na danych pochodzących z czujnika drgań zamontowanego promieniowo.

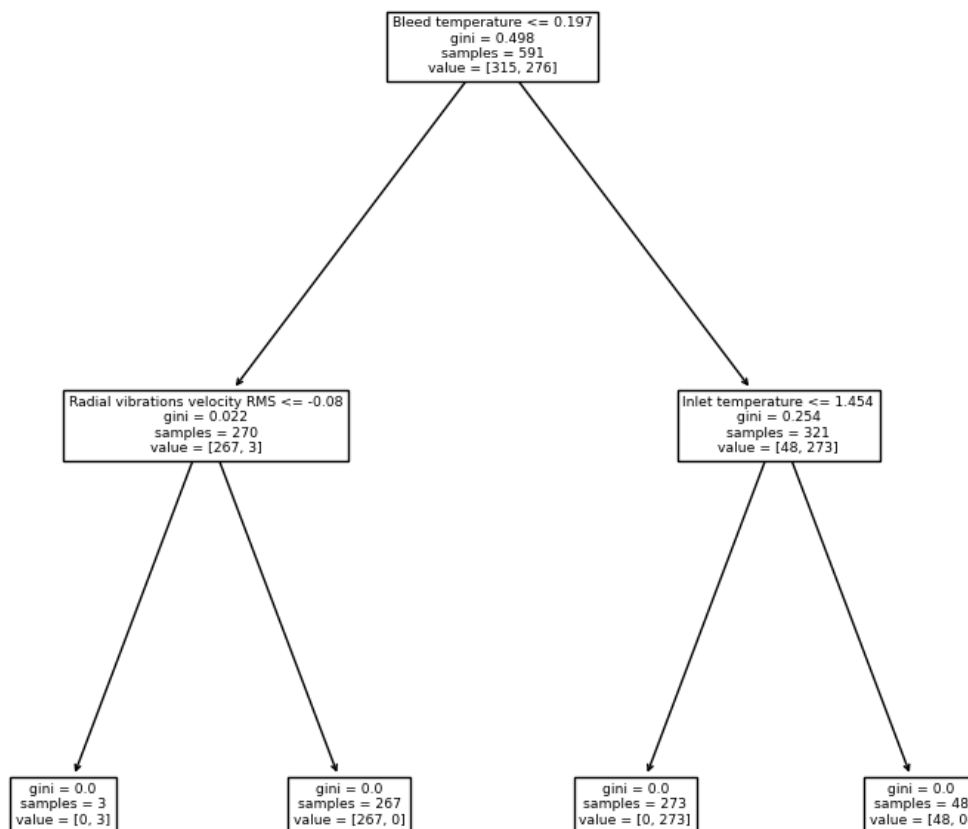
jąc od najbardziej znaczących zmiennych w kwestii separacji klas. W większej liczbie zmiennych drzewo decyzyjne zdolne jest znaleźć najistotniejsze. Przykładowe wizualizacje drzew dzięki którym klient będzie w stanie zrozumieć które zmienne najbardziej wpływają na wyniki klasyfikacji przedstawiono na rysunkach 3.5 oraz 3.6.

Regresja logistyczna jest algorytmem, który podczas klasyfikacji określa wartość pewności związanej z każdą dostępną klasą [44]. W odróżnieniu od drzewa decyzyjnego, które to każdemu przypadkowi w zbiorze danych przyporządkowuje tylko jedną klasę. Jednakże, regresja logistyczna również jest właściwym mechanizmem do tego, by dokonać ekstrakcji cech najbardziej użytecznych z punktu widzenia klasyfikacji. Koniecznym jest zastosowanie regularyzacji typu L1 [67]. Oprócz metryki F1 obliczono również wskaźnik AUC, który daje pogląd na to, jak kształtuje się charakterystyka true positives rate (TPR) vs false positives rate (FPR) dla klasyfikatora [64]. Dyskusja dotycząca wskaźników jakości znajduje się w rozdziale 3.2.2.

Tendencja związana z jakością klasyfikacji dla regresji logistycznej jest podobna jak w przypadku wcześniej omawianych drzew decyzyjnych. Im większą liczbę zmiennych bierzemy pod uwagę, tym lepsza jakość klasyfikacji. W przypadku gdy dane pochodzą wyłącznie z czujnika zamontowanego promieniowo, oraz w przypadku gdy do dyspozycji są pełne dane wibracyjne oraz dane procesowe jakość klasyfikacji dla obu podejść jest podobna. Jednakże w zestawieniu obejmującym dane z dwóch czujników wibracji, jakość klasyfikacji wypada gorzej. Należy zaznaczyć, że przyjęto formę regularyzacji, która służy uwypukleniu najważniejszych cech, czasami kosztem lepszej jakości klasyfikacji [67]. Jednak w przypadku zastosowania innego rodzaju składnika kary w funkcji



Rysunek 3.5: Fragment drzewa wygenerowanego dla klasyfikatora związanego z czujnikiem zamontowanym promieniowo i obciążeniem 10%. Rzeczywista głębokość drzewa jest kilkukrotnie większa, zbędna na potrzeby wizualizacji



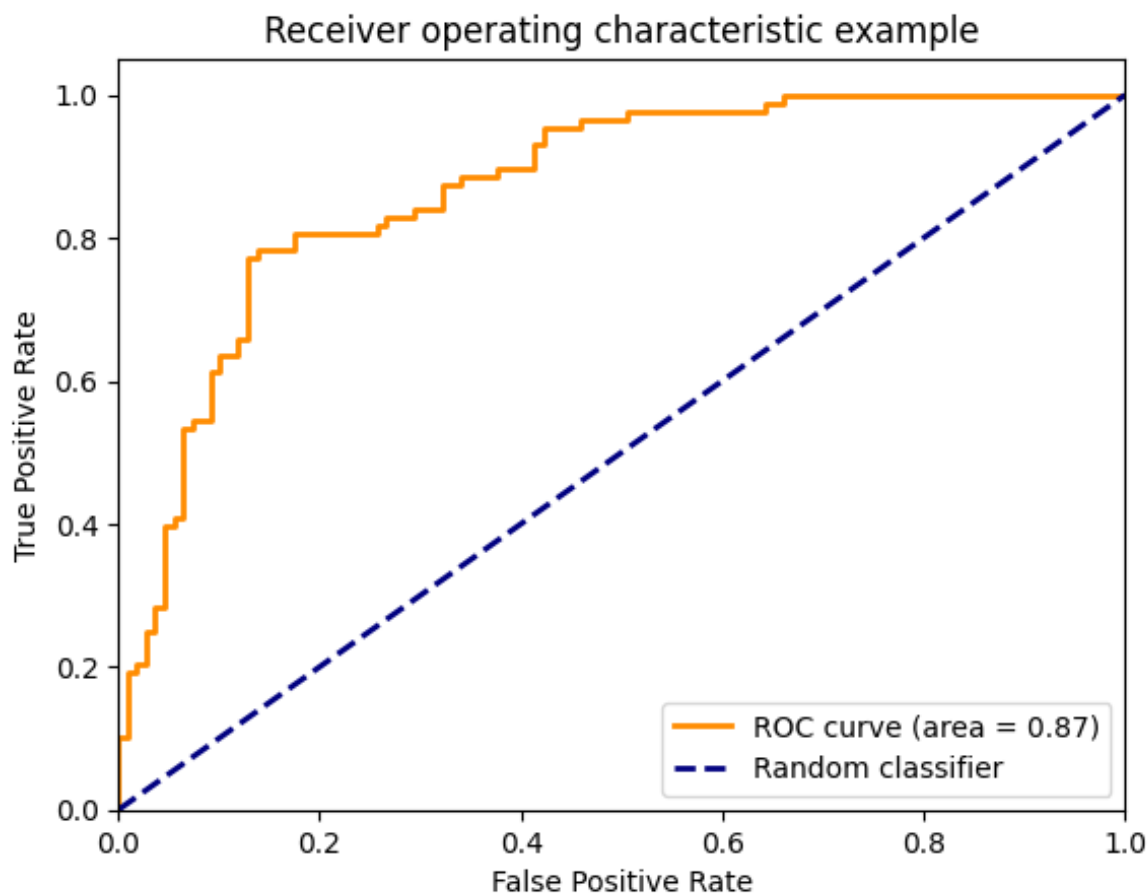
Rysunek 3.6: Pełne drzewo wygenerowane dla klasyfikatora wytrenowanego na wszystkich dostępnych zmiennych dla obciążenia 10%

Obciążenie	F1 zb. testowy	AUC	Najistotniejsze cechy
10%	0.9	0.98	VP2P osiowo, VRMS promieniowo
20%	0.99	0.99	VP2P osiowo, VP2P promieniowo
30%	0.91	0.98	VP2P osiowo, VRMS osiowo
40%	0.9	0.98	VP2P osiowo, VRMS osiowo
50%	0.89	0.96	VRMS osiowo, VP2P osiowo
60%	0.91	0.95	VRMS osiowo, VP2P promieniowo
70%	0.9	0.92	VRMS osiowo, VP2P osiowo
80%	0.89	0.95	VRMS osiowo, VP2P promieniowo
90%	0.84	0.93	VRMS promieniowo, VP2P promieniowo
100%	0.91	0.98	VP2P osiowo, VRMS osiowo

Tabela 3.6: Porównanie klasyfikatorów typu regresja logistyczna wytrenowanych estymatach drgań z obu czujników.

Obciążenie	F1 zb. testowy	AUC	Najistotniejsze cechy
10%	1.0	1.0	Ciśn. na przeciekach, Temp. na przeciekach
20%	1.0	0.99	Ciśn. na przeciekach, Temp. na przeciekach
30%	1.0	1.0	VRMS osiowo, Ciśn. na przeciekach
40%	1.0	1.0	VRMS osiowo, Ciśn. na przeciekach
50%	1.0	0.99	VRMS osiowo, Ciśn. na przeciekach
60%	1.0	1.0	VRMS osiowo, Ciśn. na przeciekach
70%	1.0	1.0	VRMS osiowo, Ciśn. na przeciekach
80%	1.0	1.0	Ciśn. na przeciekach, Temp. na przeciekach
90%	1.0	1.0	Ciśn. na przeciekach, Ciśn. na wyjściu
100%	1.0	1.0	VP2P promieniowo, VP2P osiowo

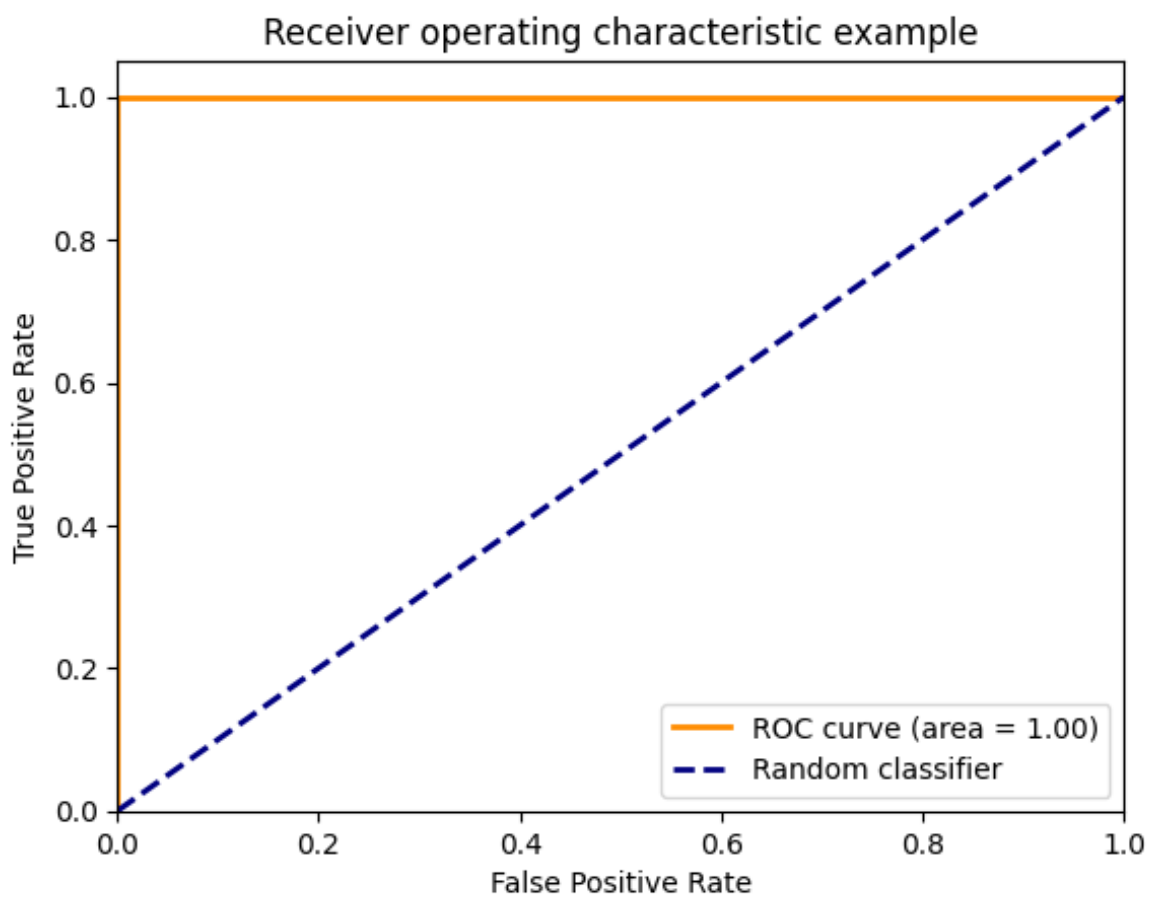
Tabela 3.7: Porównanie klasyfikatorów typu regresja logistyczna wytrenowanych na estymatach drgań oraz zmiennych procesowych.



Rysunek 3.7: Charakterystyka ROC dla klasyfikatora związanego z czujnikiem zamontowanym promieniowo i obciążeniem 10%. Klasyfikator nie jest idealny, należy zdecydować się na kompromis pomiędzy TPR a FPR.

kosztu (np. związanego z regularyzacją L2 [67]), jakość klasyfikacji jest niewiele lepsza. Wartość obliczonego wskaźnika AUC powiązana jest w pewnym stopniu z metryką F1. Jednakże AUC jest wskaźnikiem o tyle bardziej informatywnym, że analizując przebieg krzywej ROC można dostroić klasyfikator typu regresja logistyczna w sposób odpowiadający wymaganiom dotyczącym kompromisu pomiędzy TPR a FPR. Na rysunkach 3.7 oraz 3.8 zaprezentowano przykładowe krzywe ROC [38]. Zmienne procesowe wytypowane przez klasyfikatory typu regresja logistyczna oraz przez drzewa decyzyjne jako najistotniejsze, są rozbieżne. W przypadku gdy modele mają do dyspozycji jedynie dane wibracyjne oba rodzaje modeli wskazują ten sam zestaw cech jako najistotniejsze z punktu widzenia klasyfikacji.

Udało się osiągnąć idealną jakość klasyfikacji binarnej analizowanej pompy dla niemal wszystkich rodzajów obciążenia stałego. Jest to dobry prognostyk przed przystą-



Rysunek 3.8: Charakterystyka ROC dla klasyfikatora wytrenowanego na wszystkich dostępnych zmiennych dla obciążenia 10%. Klasyfikator klasyfikuje idealnie.

Obciążenie	F1 zb. testowy	Głębokość	Najistotniejsze cechy
10%	0.74	23	VRMS promieniowo, VP2P promieniowo
20%	0.7	28	VRMS promieniowo, VP2P promieniowo
30%	0.71	24	VRMS promieniowo, VP2P promieniowo
40%	0.72	22	VRMS promieniowo, VP2P promieniowo
50%	0.86	20	VRMS promieniowo, VP2P promieniowo
60%	0.84	13	VRMS promieniowo, VP2P promieniowo
70%	0.87	17	VRMS promieniowo, VP2P promieniowo
80%	0.85	13	VRMS promieniowo, VP2P promieniowo
90%	0.92	16	VRMS promieniowo, VP2P promieniowo
100%	0.85	21	VRMS promieniowo, VP2P promieniowo

Tabela 3.8: Porównanie drzew decyzyjnych wytrenowanych wyłącznie na estymatach drgań pochodzących z czujnika zamontowanego promieniowo dla klasyfikacji wieloklasowej.

Obciążenie	F1 zb. testowy	Głębokość	Najistotniejsze cechy
10%	0.96	8	VRMS promieniowo, VRMS osiowo
20%	0.99	6	VRMS promieniowo, VRMS osiowo
30%	0.97	7	VRMS promieniowo, VRMS osiowo
40%	0.98	9	VRMS promieniowo, VRMS osiowo
50%	0.97	8	VRMS promieniowo, VRMS osiowo
60%	0.99	7	VRMS promieniowo, VP2P osiowo
70%	0.97	11	VRMS promieniowo, VRMS osiowo
80%	0.93	11	VRMS promieniowo, VRMS osiowo
90%	0.92	14	VRMS promieniowo, VP2P promieniowo
100%	0.96	14	VRMS promieniowo, VP2P osiowo

Tabela 3.9: Porównanie drzew decyzyjnych wytrenowanych estymatach drgań pochodzących z obu czujników drgań dla klasyfikacji wieloklasowej.

pieniem do klasyfikacji wieloklasowej. Wytrenowane modele mają w tym przypadku na celu wskazanie precyzyjnie jakiego rodzaju usterka występuje w badanej pompie. Przeprowadzono analogiczną analizę jak w przypadku klasyfikacji binarnej. Wytrenowano drzewa decyzyjne i lasy losowe. Wyznaczono najważniejsze cechy z punktu widzenia klasyfikacji. Obliczono wskaźnik F1 pomocny w kontekście porównywania modeli między sobą.

Tabele od 3.2 do 3.13 ukazują wyniki otrzymane po wytrenowaniu modeli. Dotyczą one zarówno drzew decyzyjnych jak i regresji wielorakiej. Porównując analogiczne modele dla klasyfikacji binarnej i wielorakiej można wysnuć szereg wniosków. Modele dla klasyfikacji wieloklasowej cechuje się generalnie gorszą jakością (w kontekście wartości

Obciążenie	F1 zb. testowy	Głębokość	Najistotniejsze cechy
10%	1.0	4	Temp. na przeciekach, Ciśn. na przeciekach
20%	1.0	4	Temp. na wejściu, VRMS promieniowo
30%	1.0	4	VRMS promieniowo, Przepływ na wyjściu
40%	1.0	4	VRMS promieniowo, Temp. wejściowa
50%	1.0	4	VRMS promieniowo, Ciśn. na przeciekach
60%	1.0	3	Temp. na wejściu, Ciśn. na przeciekach
70%	1.0	3	Temp. na wejściu, Ciśn. na przeciekach
80%	1.0	3	Temp. na wejściu, Ciśn. na przeciekach
90%	1.0	3	Temp. na wejściu, Ciśn. na przeciekach
100%	1.0	5	VRMS promieniowo, Temp. na przeciekach

Tabela 3.10: Porównanie drzew decyzyjnych wytrenowanych estymatach drgań oraz zmiennych procesowych dla klasyfikacji wieloklasowej.

Obciążenie	F1 zb. testowy	Najistotniejsze cechy
10%	0.52	VRMS promieniowo, VP2P promieniowo
20%	0.63	VP2P promieniowo, VRMS promieniowo
30%	0.65	VRMS promieniowo, VP2P promieniowo
40%	0.64	VRMS promieniowo, VP2P promieniowo
50%	0.66	VRMS promieniowo, VP2P promieniowo
60%	0.75	VRMS promieniowo, VP2P promieniowo
70%	0.71	VRMS promieniowo, VP2P promieniowo
80%	0.79	VP2P promieniowo, VRMS promieniowo
90%	0.64	VRMS promieniowo, VP2P promieniowo
100%	0.73	VRMS promieniowo, VP2P promieniowo

Tabela 3.11: Porównanie klasyfikatorów typu regresja logistyczna wytrenowanych na danych pochodzących z czujnika drgań zamontowanego promieniowo dla klasyfikacji wieloklasowej.

Obciążenie	F1 zb. testowy	Najistotniejsze cechy
10%	0.76	VRMS promieniowo, VP2P promieniowo
20%	0.86	VP2P osiowo, VRMS osiowo
30%	0.81	VP2P promieniowo, VRMS promieniowo
40%	0.81	VP2P promieniowo, VP2P osiowo
50%	0.85	VRMS promieniowo, VP2P promieniowo
60%	0.84	VRMS promieniowo, VP2P promieniowo
70%	0.8	VRMS promieniowo, VP2P promieniowo
80%	0.83	VP2P promieniowo, VRMS promieniowo
90%	0.73	VRMS osiowo, VRMS promieniowo
100%	0.86	VP2P promieniowo, VRMS osiowo

Tabela 3.12: Porównanie klasyfikatorów typu regresja logistyczna wytrenowanych na estymatach drgań z obu czujników dla klasyfikacji wieloklasowej.

Obciążenie	F1 zb. testowy	Najistotniejsze cechy
10%	1.0	VRMS osiowo, VP2P promieniowo
20%	1.0	VP2P promieniowo, Ciśn. na wyjściu
30%	1.0	Temp. na przeciekach, VP2P promieniowo
40%	1.0	Temp. na przeciekach, VP2P promieniowo
50%	1.0	Temp. na przeciekach, Ciśn. na wyjściu
60%	1.0	Temp. na przeciekach, VP2P promieniowo
70%	1.0	VRMS osiowo, Temp. na przeciekach
80%	1.0	VRMS osiowo, Temp. na przeciekach
90%	1.0	VRMS osiowo, Temp. na przeciekach
100%	1.0	VRMS osiowo, Temp. na przeciekach

Tabela 3.13: Porównanie klasyfikatorów typu regresja logistyczna wytrenowanych na estymatach drgań oraz zmiennych procesowych dla klasyfikacji wieloklasowej.

wskaźnika F1) niż jego odpowiednik dla zadanego obciążenia dla klasyfikacji binarnej. Natomiast są od tego wyjątki wyjaśnione w dyskusji na temat wskaźników jakości w rozdziale 3.2.2. Najistotniejsze cechy z punktu widzenia klasyfikacji wieloklasowej są w wielu przypadkach inne, niż w przypadku klasyfikacji binarnej. Można natomiast wskazać szereg generalizujących tez. Modele drzewa decyzyjnego uwzględniające wszystkie dostępne pomiary dla klasyfikacji binarnej wskazują wibracje promieniowe oraz temperaturę na wyjściu z pompy jako istotne. Z kolei drzewo decyzyjne korzystające z wszystkich dostępnych pomiarów dla klasyfikacji wieloklasowej wskazuje temperaturę i ciśnienie na przeciekach jako najistotniejsze pomiary. Drzewa wytrenowane wyłącznie na danych wibracyjnych niezależnie od rodzaju klasyfikacji wskazują na te same pomiary jako najistotniejsze. W przypadku regresji logistycznej i wielorakiej charakter zestawienia najważniejszych cech wypada podobnie jak w przypadku drzew decyzyjnych. Inne są natomiast konkretne pomiary wskazane przez modele. Regresja logistyczna wskazuje na istotność ciśnienia i temperatury medium tłoczonego przez pompę na przeciekach. Podczas gdy regresja logistyczna wskazuje na ciśnienie tłoczonego medium oraz na wibracje. Analiza złożoności drzew decyzyjnych wskazuje, że dla klasyfikacji wieloklasowej głębokość drzew jest większa. Dla mniejszego zakresu dostępnych zmiennych regresja logistyczna i wieloraka cechuje się gorszą jakością klasyfikacji niż drzewa decyzyjne.

Dla klasyfikatora typu regresja wieloraka nie obliczano wartości AUC. Powód takiej decyzji został omówiony w rozdziale 3.2.2. Na potrzeby analizy zdecydowano się przedstawić macierze pomyłek. Rysunki 3.9 oraz 3.10 prezentują przykładowe macierze pomyłek dla regresji wielorakiej. Macierz zaprezentowana na rysunku 3.10 ilustruje

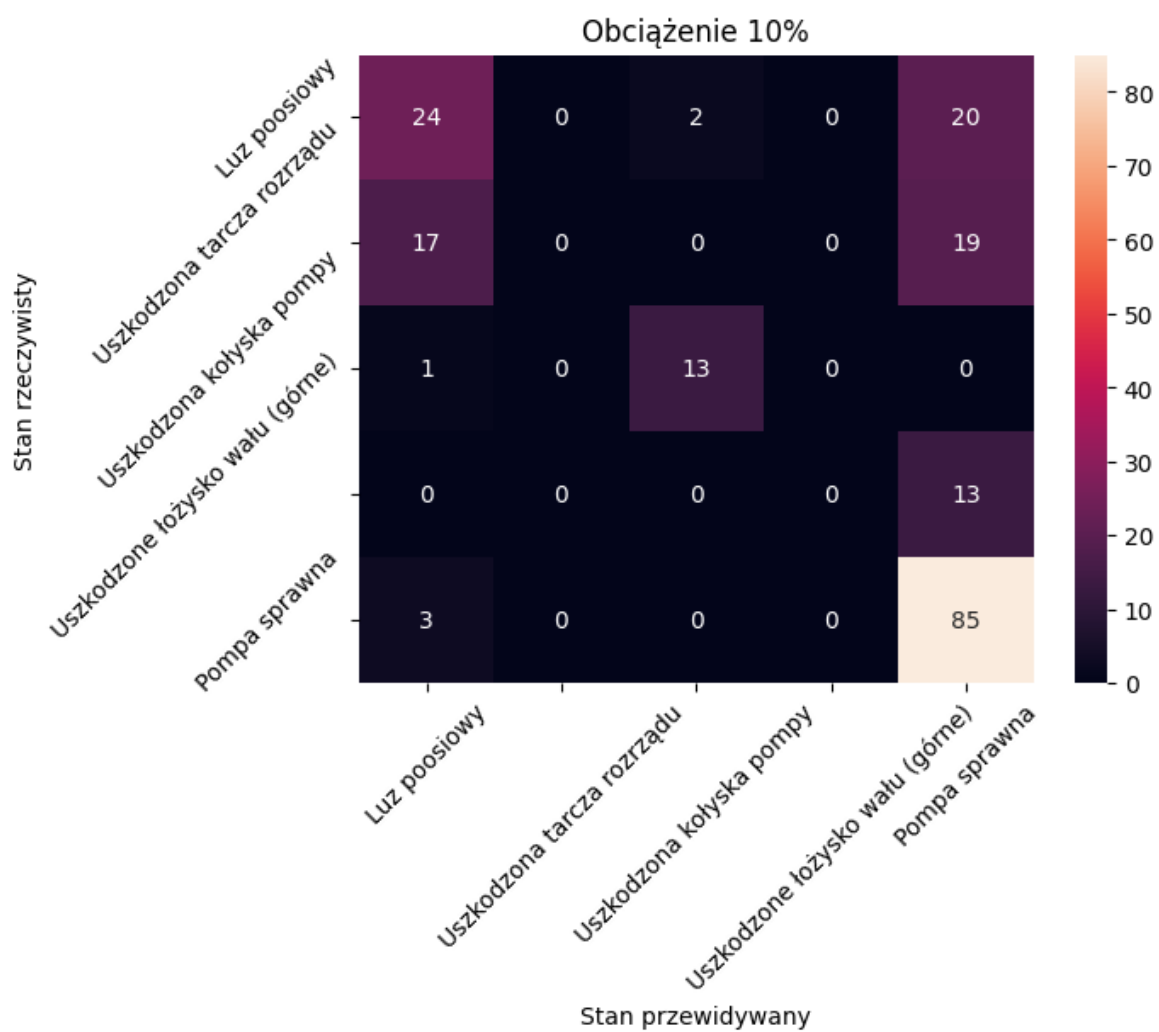
idealną klasyfikację. Niezerowe wartości znajdują się jedynie na głównej przekątnej macierzy. Z kolei na rysunku 3.9 zaobserwować można między innymi absolutny brak poprawności klasyfikacji dla pompy z uszkodzonym łożyskiem oraz z uszkodzoną tarczą rozrządu. Ponadto widoczne jest, że klasyfikator związany z macierzą najczęściej błędnie określa pompę jako sprawną, podczas gdy w rzeczywistości jest uszkodzona. Klasyfikator cechuje się bardzo niską czułością, gdyż wiele jest przypadków w których pompę uszkodzoną klasyfikuje jako sprawną. Z drugiej strony cechuje się wysoką precyzją z tego względu, że jest tylko kilka przypadków które zostały błędnie zaklasyfikowane jako sprawna, podczas gdy pompa była uszkodzona.

Zarówno modele służące klasyfikacji binarnej jak i klasyfikacji wieloklasowej są zdolne do klasyfikacji dobrej, a nawet idealnej, jakości. Modele jako dane wejściowe przyjmują wektor danych związanych z sekundą pracy pompy. Zakładając, że podczas wdrożenia próbkowanie danych procesowych będzie sekundowe korzystanie z któregośkolwiek z modeli nie będzie wymagało wstępnego przetwarzania danych.

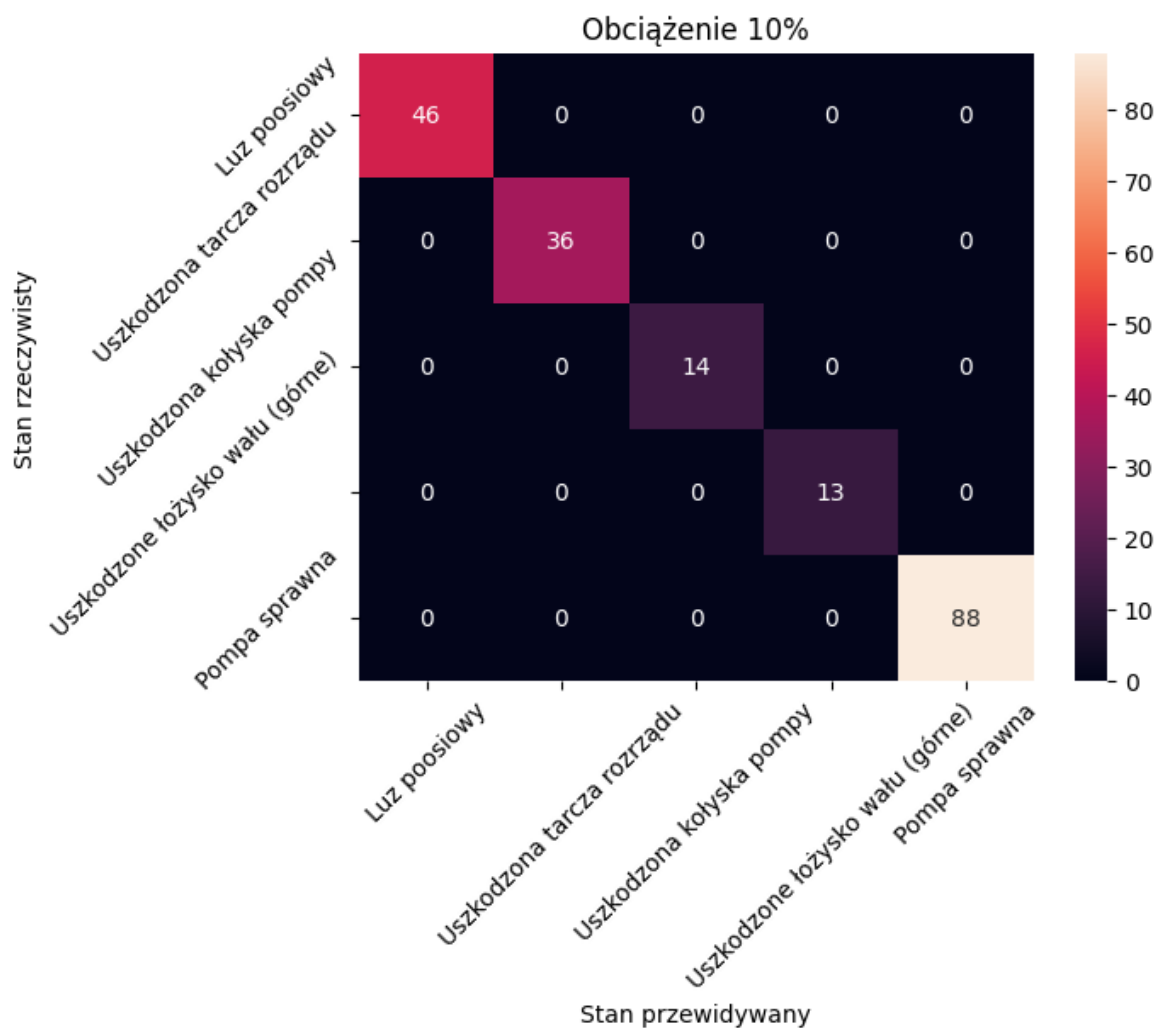
Widmo drgań

Widmo drgań odczytywane jest z urządzenia EXT/FFT co sekundę jako strumień bitów transmitowanych za pomocą portu szeregowego w urządzeniu. Akwizycja widma umożliwi syntezę spektrogramu. Na potrzeby dalszej analizy może być on traktowany jako obraz monochromatyczny. Każdy piksel w takim obrazie jest wartością typu całkowitego bez znaku (UINT). Z racji tego, że EXT/FFT próbkuje sygnał drgań z częstotliwością 20kHz, to zgodnie z twierdzeniem o próbkowaniu widmo drgań zawiera informacje o składowych drgań do częstotliwości 10kHz. Twierdzenie o próbkowaniu stanowi bowiem, że jeśli sygnał ciągły nie ma składowych widma o częstotliwości równej lub większej niż f , może on zostać wiernie odtworzony z ciągu jego próbek tworzących sygnał dyskretny, o ile próbki te zostały pobrane z częstotliwością co najmniej $2f$ [78]. Spektrogramy pochodzą z analizy danych z czujnika zamontowanego promiennie. Przykładowy spektrogram z jednego pełnego eksperymentu pomiarowego ukazano na wykresie 3.11.

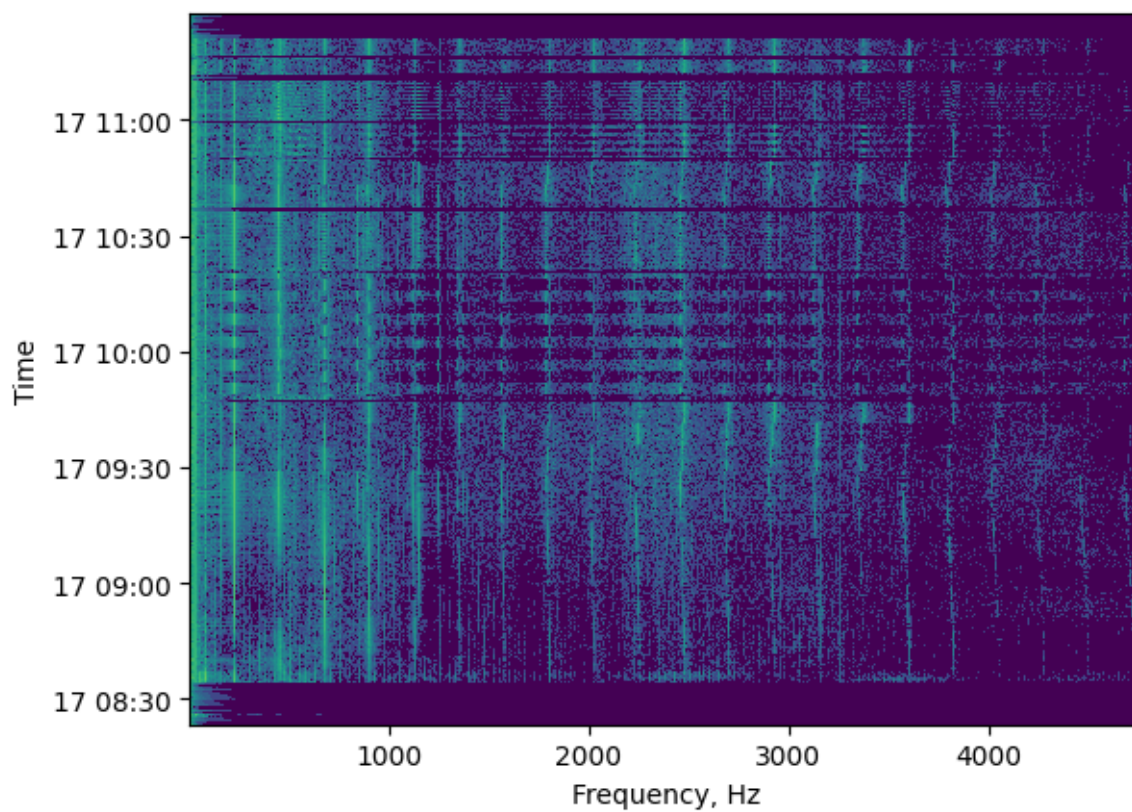
Oprócz przeprowadzenia klasyfikacji dobrej jakości należy zadbać o przejrzysty sposób wizualizacji. Tak, aby dawał on użytkownikowi końcowemu pewien ogląd na to, czym różni się pompa sprawna od uszkodzonej. Problem klasyfikacji spektrogramów można sprowadzić do zagadnienia klasyfikacji obrazów. Rozwiązuje się je często wykorzystując konwolucyjne sieci neuronowe (CNN) [67]. W ramach takiej sieci wydzielić



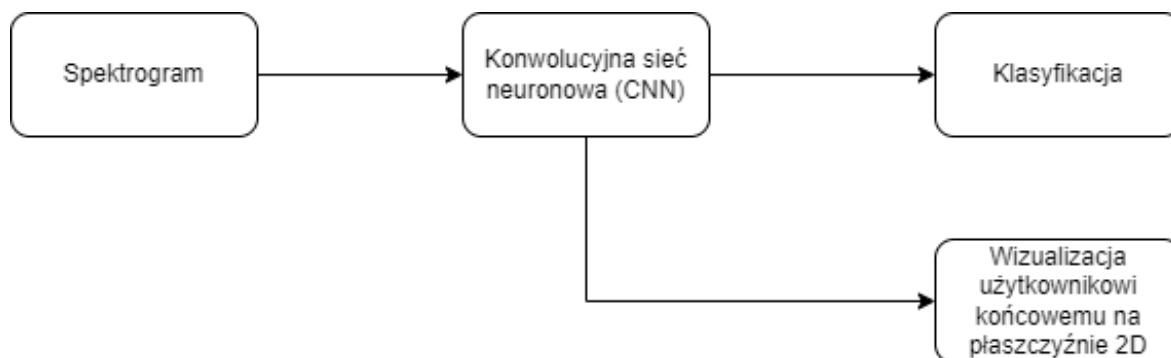
Rysunek 3.9: Macierz pomyłek dla klasyfikatora typu regresja wieloraka dla obciążenia stałego równego 10%. Klasyfikator został wytrenowany na danych pochodzących wyłącznie z czujnika drgań zamontowanego promieniowo.



Rysunek 3.10: Macierz pomyłek dla klasyfikatora typu regresja wieloraka dla obciążenia stałego równego 10%. Klasyfikator został wytrenowany wszystkich dostępnych zmiennych mierzonych.



Rysunek 3.11: Spektrogram prędkości drgań dla pełnego eksperymentu pomiarowego pompy z uszkodzonymi łożyskami. Zakres częstotliwości został przycięty na potrzeby lepszej wizualizacji.



Rysunek 3.12: Wysokopoziomowy schemat przetwarzania spektrogramów w celu ekstrakcji i wizualizacji cech oraz klasyfikacji.

można część służącą do ekstrakcji cech. Zawiera ona warstwy odpowiedzialne za przeprowadzenie operacji konwolucji i pooling. Ostatnia z warstw pooling poddawana jest operacji spłaszczenia celem otrzymania jednowymiarowego wektora danych. Wektor ten jest pierwszą z warstw drugiej części sieci CNN, mianowicie części klasyfikującej. Jest ona klasyczną siecią typu wielowarstwowy perceptron (MLP) z wyjściową warstwą softmax [67]. Zdolność sieci neuronowej do ekstrakcji cech i klasyfikacji uszkodzenia umożliwia przetwarzanie spektrogramów dwutorowo. Dokonać można klasyfikacji uszkodzenia, a ponadto uzyskać wektor liczb opisujących każdy ze spektrogramów w wielowymiarowej przestrzeni. Wysokopoziomowy schemat przetwarzania spektrogramów ukazano na diagramie 3.12.

Aby uczyć sieć a następnie ją wykorzystywać spektrogramy należy poddać obróbcę w celu uzyskania standardowego rozmiaru. Każdy ze spektrogramów został podzielony na części na osi czasu. Każda z części związana jest z etapami eksperymentu pomiarowego, wyszczególnionego w tabeli 3.1. Długość każdej z części spektrogramu po podzieleniu wynosi 64 piksele, a zatem 64 sekundy. Spektrogram na osi częstotliwości został podzielony na części zawierające 64 piksele, a zatem biny częstotliwości. Liczba pikseli została wybrana arbitralnie i na potrzeby trenowania analogicznych modeli może być zmieniana. Wynikowa macierz pikseli jest kwadratowa, ponieważ umożliwia to efektywne przetwarzanie danych wejściowych przez konwolucyjną sieć neuronową [67]. Wzięto pod uwagę fakt, że im mniejsze są obrazy wejściowe, tym krócej trwa proces uczenia. Jest to swego rodzaju kompromis pomiędzy liczbą informacji potencjalnie zawartych w obrazie, a długością trwania procesu uczenia. W przypadku gdy spektrogram związany z konkretnym etapem eksperymentu pomiarowego był za długi bądź za krótki odpowiednio dokonywano operacji przycięcia. Z kolei gdy spektrogram był za

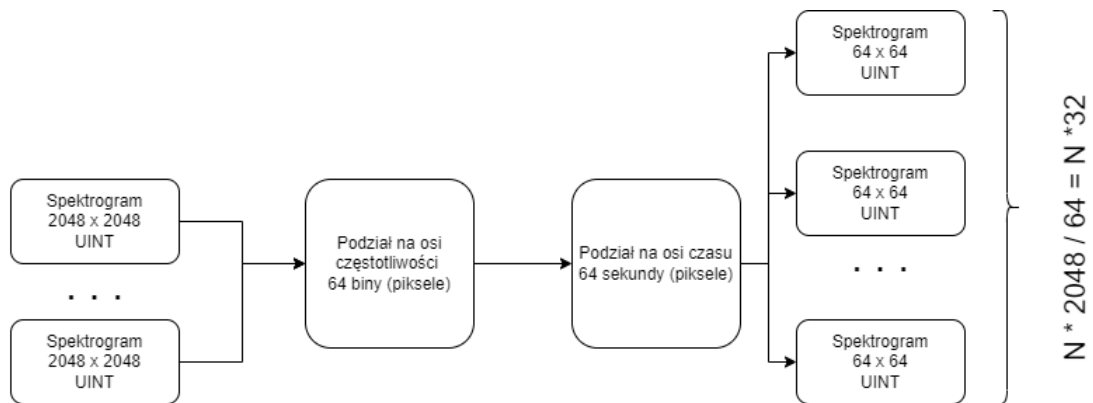
krótki, sztucznie powiększono powielając widmo znajdujące się na końcu spektrogramu, to jest w ostatniej dostępnej sekundzie serii pomiarowej. Z racji tego, że jeden bin częstotliwości to ok. 4,88Hz (20000Hz / 4096 próbek w buforze) każdy z fragmentów spektrogramu na osi częstotliwości obejmuje pasmo częstotliwości o szerokości 312,5Hz ($64 * 4,88\text{Hz}$). Dla każdego pasma częstotliwości został wytrenowany osobny model klasyfikacji i ekstrakcji cech.

Bardzo istotne znaczenie ma fakt, że danych potrzebnych do wytrenowania modeli było niewiele. Nie sposób efektywnie wytrenować sieci neuronowe mając do dyspozycji niewiele przykładów w zbiorze danych. Przeprowadzono zatem transfer learning [67]. Wytrenowano szereg modeli na syntetycznych danych w celu ich późniejszego wykorzystania w połączeniu z danymi rzeczywistymi.

Modele wytrenowane na syntetycznych spektrogramach

Proces syntezy modeli należy podzielić na dwie części. W pierwszej kolejności dokonano generacji danych syntetycznych. Z racji tego, że modele posłużyć mają do klasyfikacji uszkodzeń wygenerowano pięć klas spektrogramów. Każda z klas cechowała się innym zestawem częstotliwości. Częstotliwość ta była niezmienna w czasie.

Generowano spektrogramy o takim samym zakresie częstotliwości oraz o takiej samej liczbie binów częstotliwości co spektrogramy pochodzące z eksperymentu pomiarowego. Z kolei na osi czasu miały taki rozmiar, aby wynikowe spektrogramy były macierzami kwadratowymi. Każda komórka macierzy jest typu całkowitoliczbowego bez znaku (UINT). Finalnie wygenerowane dane można traktować jak obrazy monochromatyczne o rozmiarach 2048 x 2048. Obrazy te zostały zaszumione z wykorzystaniem szumu o rozkładzie jednostajnym [69] w celu powiększenia zbioru dostępnych danych oraz uodpornienia sieci na szum pomiarowy [67]. Nie dokonywano innych operacji związanych z procesem poszerzania liczebności zbioru treningowego takich jak rozciąganie, skalowanie, obrót. Proces generacji spektrogramu powtórzono wielokrotnie w celu zapewnienia odpowiednio obszernego zbioru danych. Z tak powstałych spektrogramów zarówno na osi czasu jak i na osi częstotliwości dokonano podziału na mniejsze spektrogramy o rozmiarach 64 x 64. Na osi częstotliwości w skład spektrogramu wchodziły kolejne 64 biny. Zaś na osi czasu w skład kolejnych spektrogramów wchodziły kolejne 64 sekundy. Proces dostosowywania rozmiaru spektrogramów na potrzeby treningu sieci odpowiada przygotowaniu spektrogramów pochodzących z eksperymentu pomiarowego. Zachowanie spójności związanej z rozmiarem oraz zakresem częstotliwości danych



Rysunek 3.13: Schemat generacji i przetwarzania spektrogramów na potrzeby treningu modeli z wykorzystaniem danych syntetycznych.

wejściowych jest niezbędne w kontekście uzyskania wysokiej jakości modeli. Schemat generacji i przetwarzania spektrogramów ukazano na rysunku 3.13.

W związku z konkretnym formatem danych wejściowych oraz liczbą klas generowanych spektrogramów równą pięć, część architektury sieci jest narzucona. Warstwy ukryte w części sieci ekstrahującej cechy składają się z trzech warstw konwolucji, oraz dwóch warstw max pooling. Każda z warstw cechuje się funkcją aktywacji typu relu. Część klasyfikująca sieci posiada warstwę spłaszczającą macierze cech ostatniej warstwy konwolucji oraz z dwóch warstw typu dense z których warstwa wyjściowa ma pięć neuronów w związku z pięcioma syntetycznymi klasami spektrogramów. Podczas przetwarzania obrazów przez sieci CNN należy rozwiązać kwestię pikseli będących na brzegach obrazu. W rozpatrywanym przypadku skrajne piksele zostały odrzucone. Stąd pierwsza warstwa konwolucji posiada rozmiar 62 x 62 zamiast 64 x 64. Analogiczne odrzucanie skrajnych pikseli odbywa się w kolejnych warstwach konwolucji oraz pooling. Projektowanie sieci neuronowych jest zadaniem złożonym między innymi z uwagi na mnogość rozmiaru, typu warstw oraz funkcji aktywacji. Przyjęta architektura została zaczerpnięta z [64] jako dobrze nadająca się do klasyfikacji obrazów z bazy CIFAR [84]. Szczegółowy opis architektury sieci ukazany jest w tabeli 3.14.

Jak ukazuje tabela 3.15 modele wykorzystujące dane syntetyczne były zdolne do idealnej klasyfikacji syntetycznych spektrogramów we wszystkich zakresach częstotliwości. Trening i ewaluacja sieci odbywała się z wykorzystaniem podziału na zbiór treningowy i testowy w proporcjach 80:20. W celu zapobiegnięcia zjawisku nadmiernego dopasowania sieci zastosowano regularyzację early stopping [67]. Jako funkcję kosztu przyjęto

Warstwa (typ)	Wymiary	Liczba parametrów
conv2d (Conv2D)	(62, 62, 32)	320
max pooling2d (MaxPooling2D)	(31, 31, 32)	0
conv2d 1 (Conv2D)	(29, 29, 64)	18496
max pooling2d 1 (MaxPooling2D)	(14, 14, 64)	0
conv2d 2 (Conv2D)	(12, 12, 64)	36928
flatten (Flatten)	(9216)	0
dense (Dense)	(64)	589888
dense 1 (Dense)	(5)	320
Suma		645957

Tabela 3.14: Architektura sieci CNN

Zakres częstotliwości, Hz	F1 zb. testowy
0-307	1.0
312-620	1.0
625-932	1.0
937-1245	1.0
1250-1557	1.0

Tabela 3.15: Ewaluacja wybranych modeli klasyfikujących syntetyczne spektrogramy na podstawie wskazanych zakresów częstotliwości

sparse categorical cross-entropy jako że pozwala maksymalizować pewność neuronu wyjściowego związanego z właściwą klasą [37]. Optymalizacji wag sieci dokonano posługując się procedurą optymalizacji Adam [82].

Projektowania i treningu sieci dokonano z wykorzystaniem biblioteki tensorflow [29]. Zawiera ona definicję wszystkich wykorzystanych typów warstw, funkcji aktywacji, funkcji kosztu oraz procedury optymalizacji parametrów sieci.

Analiza obciążenia stałego

W odróżnieniu od podejścia zastosowanego dla danych procesowych, dla wszystkich wartości obciążenia stałego wytrenowano jeden model. Architektura modelu jest tożsama z przedstawioną w tabeli 3.14. Danymi wejściowymi do modelu są spektrogramy pochodzące z eksperymentu pomiarowego. Ich przygotowanie jest bardzo zbliżone do tego, którym poddane były spektrogramy syntetyczne. Z eksperymentu pomiarowego zostały pobrane fragmenty związane z obciążeniem stałym. Podczas ich przygotowania na potrzeby analizy zadbane, aby analizowany był tylko stan ustalony, bez momentów tranzycji pomiędzy obciążeniami.

Klasa	Liczebność
Uszkodzona tarcza rozrządu	21
Uszkodzona kołyska pompy	11
Pompa sprawna	41
Luz poosiowy	22
Uszkodzone łożysko wału (górne)	11

Tabela 3.16: Zestawienie liczebności klas spektrogramów wyekstrahowanych z eksperymentu pomiarowego dla obciążenia stałego.

Efektywny trening sztucznych sieci neuronowych zakłada odpowiednio duży zbiór danych treningowych. Jednakże populacja wyodrębnionych spektrogramów jest zbyt mała, aby wyłącznie na niej wytrenować sieć o wysokiej zdolności klasyfikacji. Skuteczność sieci wytrenowanej na danych syntetycznych podczas ewaluacji na spektrogramach rzeczywistych była tylko nieco lepsza niż klasyfikatora losowego. Liczebność każdej z klas, z którą związane są spektrogramy z eksperymentu pomiarowego przedstawiono w tabeli 3.16. Klasy nie są zbalansowane, co uwzględniono podczas ewaluacji modelu.

W celu podniesienia jakości klasyfikacji przydatny okazuje się mechanizm zwany transferem wiedzy. Wykorzystując dane rzeczywiste, dotrenowano modele wytrenowane na danych syntetycznych. Sieć po treningu na danych syntetycznych jest niejako punktem startowym dla kolejnego treningu. Wykorzystano dwa podejścia [99]. Jedno obejmuje ponowny trening jedynie części sieci odpowiedzialnej za klasyfikację (warstwy dense wyszczególnione w tabeli 3.14). Drugie zaś zakłada ponowny trening całej sieci.

Na potrzeby treningu sieci warstwy służące do ekstrakcji cech (konwolucja, pooling) zostają niejako zamrożone. Oznacza to, że wagi sieci związane z tymi warstwami nie ulegają zmianie podczas uczenia. Konceptyjnie można umotywić to następująco. Zakładamy zdolność sieci do pozyskania ze spektrogramu wejściowego wartościowych informacji. Może to być zdolność do detekcji wiodących częstotliwości czy też wykrywanie zależności pomiędzy częstotliwościami. Z uwagi na złożoność sieci neuronowej nie sposób precyzyjnie opisać jakiego rodzaju wzorce sieć jest w stanie rozpoznawać [64]. Trening na danych syntetycznych miał na celu nauczenie sieci ekstrakcji użytecznych cech ze spektrogramu. Cech niekoniecznie łatwych do nazwania, aczkolwiek istotnych z punktu widzenia klasyfikacji spektrogramów rzeczywistych.

Część sieci służąca do klasyfikacji wymaga dotrenowania. Inaczej bowiem sieć jest w stanie kwantyfikować cechy pochodzące ze spektrogramów rzeczywistych a inaczej z syntetycznych. Podczas douczania wagi ostatnich dwóch warstw sieci ulegają zmianie.

Zakres częstotliwości, Hz	F1 zb. testowy
0-307	1.0
2500-2807	1.0
1250-1557	0.97
2812-3120	0.96
3125-3432	0.96
9375-9682	0.24
8750-9057	0.19
6875-7182	0.17
8437-8745	0.14

Tabela 3.17: Metryki kilku modeli klasyfikacji spektrogramów wyekstrahowanych z eksperymentu pomiarowego dla obciążenia stałego. Sieci dotrenowane w części klasyfikującej.

Dzięki czemu znacząco poprawiła się jakość klasyfikacji w porównaniu do modeli wytrenowanych na danych syntetycznych i ewaluowanych na danych rzeczywistych, gdyż sieci te klasyfikowały spektrogramy rzeczywiste w sposób zbliżony do klasyfikatora losowego. Metryki dla kilku modeli CNN dla których przeprowadzono dotrenowanie w części klasyfikującej przedstawiono w tabeli 3.17.

Na podstawie danych z tabeli 3.17 można określić, które zakresy częstotliwości są najbardziej, a które najmniej użyteczne na potrzeby klasyfikacji. Do najistotniejszych zakresów częstotliwości należą zakresy 0-307Hz oraz 2500-2807Hz. Z kolei do najmniej istotnych należą zakresy związane z wysokimi częstotliwościami.

Aby opracować drugie z podejść przeanalizowanych w kontekście transferu wiedzy dokonano treningu całej sieci na danych rzeczywistych. Podczas treningu zmieniały się wagi we wszystkich warstwach sieci. Tabela 3.18 ukazuje zestawienie modeli dotrenowanych na rzeczywistych spektrogramach. Na podstawie przedstawionych danych można zauważyć, że modele o najlepszej jakości klasyfikacji związane są z tymi samymi zakresami częstotliwości co dla sieci wytrenowanych jedynie w części klasyfikującej. Podobnie jest w przypadku najmniej użytecznych zakresów częstotliwości. Podobnie jak w przypadku modeli w których wytrenowano jedynie część klasyfikującą wyższe zakresy częstotliwości widma okazują się najmniej użyteczne.

Porównano rozkład wartości F1 dla sieci CNN dotrenowanej w całości oraz wyłącznie w części klasyfikacyjnej. Tabela 3.19 ukazuje, że średnie wartości F1 dla sieci dotrenowanych na oba analizowane sposoby są podobne. Różnice w obliczonych statystykach sięgają kilku procent. Inaczej kształtują się wartości statystyk dla różnicy

Zakres częstotliwości, Hz	F1 zb. testowy
0-307	0.96
312-620	0.96
1250-1557	0.89
3125-3432	0.89
9062-9370	0.2
9375-9682	0.2
9687-9995	0.2
8437-8745	0.19

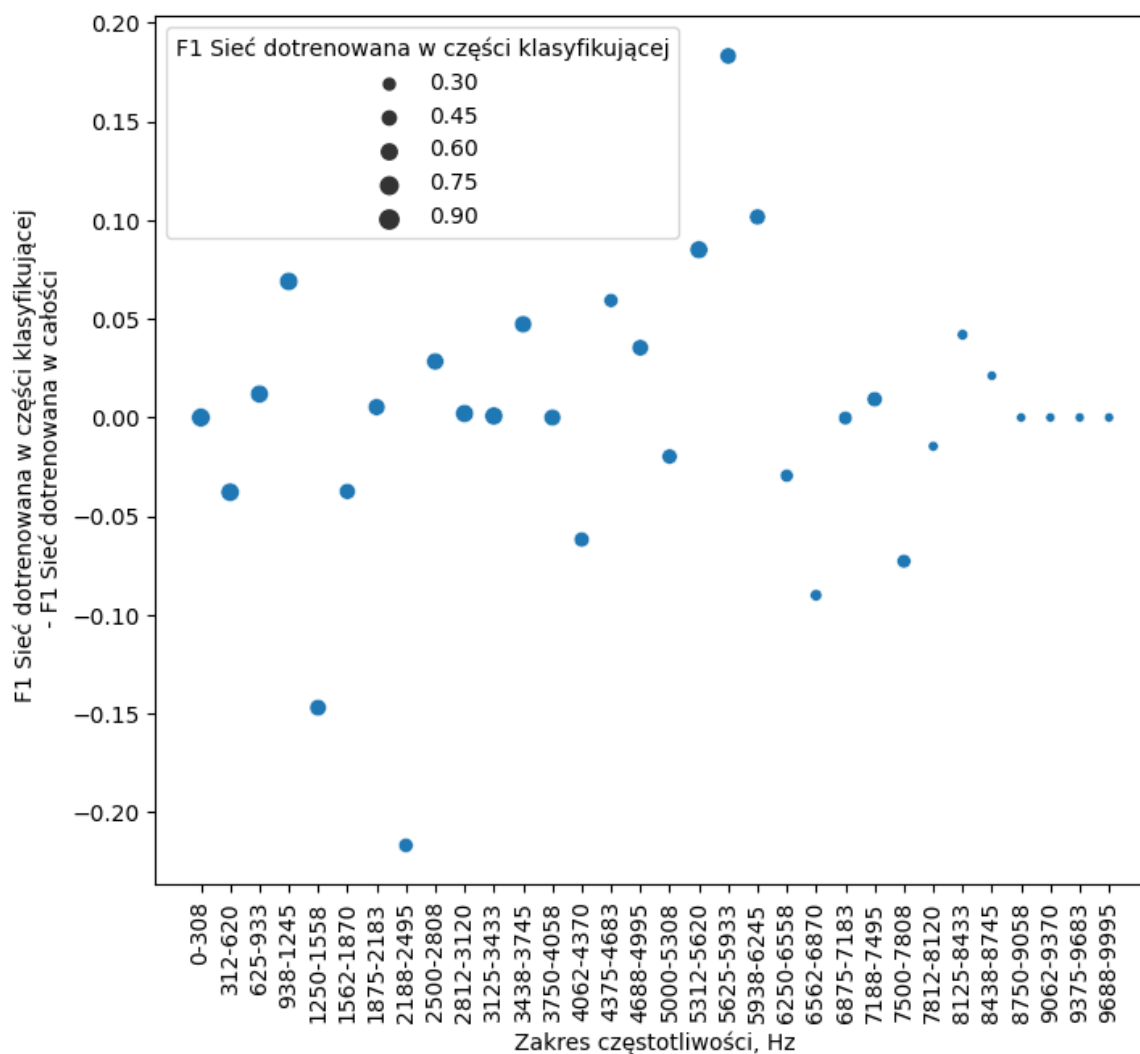
Tabela 3.18: Metryki kilku modeli klasyfikacji spektrogramów wyekstrahowanych z eksperymentu pomiarowego dla obciążenia stałego. Sieci dotrenowane w całości.

Statystyka	F1 Sieć dotrenowana w części klasyfikującej	F1 Sieć dotrenowana w całości	Różnica pomiędzy wartościami F1
mean	0.6	0.61	-0.0
std	0.26	0.25	0.07
min	0.2	0.19	-0.22
25%	0.42	0.46	-0.02
50%	0.66	0.66	0.0
75%	0.81	0.78	0.03
max	1.0	0.96	0.18

Tabela 3.19: Porównanie wartości F1 dla dotrenowanych sieci oraz różnicy pomiędzy wartościami F1 dla modeli z analogicznego zakresu częstotliwości między sobą.

między wartościami F1. Rysunek 3.14 ilustruje to, jak odmienna dla modeli z niektórych zakresów częstotliwości jest metryka F1. Najlepiej widoczne jest to dla zakresów częstotliwości 2188-2485Hz oraz 5625-5993Hz. W tych zakresach wyraźnie lepszą jakością klasyfikacji cechuje się odpowiednio model dotrenowany w całości oraz model dotrenowany w części klasyfikującej.

Metryka F1 obliczona tak samo jak dla modeli drzewa decyzyjnego, regresji logistycznej i sieci CNN pozwala porównać podejścia do modelowania modelowania między sobą. Analiza spektrogramów oparta jest wyłącznie o dane z czujnika wibracji zamontowanego promieniowo. W pierwszej kolejności należy zatem odnieść jakość klasyfikacji do modeli wytrenowanych na danych pochodzących wyłącznie z tego samego czujnika. Porównując tabele 3.17, 3.18, 3.8,3.11, można zauważyć, że model wytrenowany z wykorzystaniem spektrogramów cechuje się najlepszą jakością klasyfikacji. Niezależnie od tego, do której procentowej wartości obciążenia się odniesiono. Warty odnotowania zatem jest fakt, że modele typu drzewo decyzyjne i regresja logistyczna zostały wytre-



Rysunek 3.14: Różnica wartości F1 pomiędzy siecią dotrenowaną w części klasyfikującej, a siecią dotrenowaną w całości. Zestawienie dla wszystkich modeli związanych ze wszystkimi dostępnymi zakresami częstotliwości.

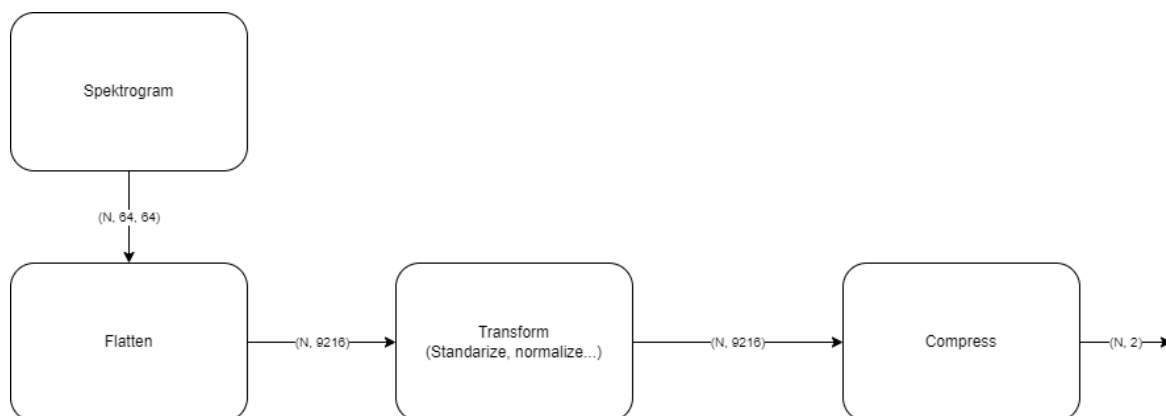
Rodzaj modelu	F1 zb. testowy
CNN dotrenowana w części klasyfikującej, 0-307Hz	1.0
CNN dotrenowana w części klasyfikującej, 2500-2807Hz	1.0
CNN dotrenowana w całości, 0-307Hz	1.0
Drzewo decyzyjne, obc. 90%	0.92
Drzewo decyzyjne, obc. 70%	0.87
Regresja logistyczna, obc. 80%	0.79
Regresja logistyczna, obc. 60%	0.75

Tabela 3.20: Porównanie modeli wytrenowanych na danych z czujnika wibracji zamontowanego promieniowo

nowane przy założeniu znajomości obciążenia pod którym pracuje pompa. Sieć CNN jest bardziej elastyczna i do jej właściwego działania nie jest potrzebna informacja o zadanym obciążeniu. Sieć CNN dotrenowana wyłącznie w części klasyfikującej cechuje się lepszą jakością klasyfikacji aniżeli sieć dotrenowana w całości. Wynika to z tego, że dotrenowanie całej sieci w sposób efektywny wymagałoby zgromadzenia większej ilości danych. Szersze omówienie powyższego zjawiska, a więc konfrontację jakości uczenia w zależności od rozmiaru zbioru danych w ramach transferu wiedzy, można w literaturze zaobserwować w [52]. Jako podsumowanie syntezy i ewaluacji wypracowanych modeli klasyfikacji w tabeli 3.20 umieszczono ich porównanie.

Rzutowanie wyekstrahowanych cech spektrogramów na płaszczyznę 2D

Poprzez wytrenowanie modeli syntetycznych dokonano transferu wiedzy [64]. Bardzo dobra jakość klasyfikacji danych syntetycznych implikuje umiejętność sieci do wykrywania wzorców zawartych w spektrogramach. Jest więc pomocna w kwestii ekstrakcji cech ze spektrogramów. Ostatnia z warstw pooling w części ekstrahującej cechy poddawana jest operacji spłaszczenia celem otrzymania jednowymiarowego wektora danych. Tak przygotowana warstwa (tzw. warstwa flatten) jest warstwą wejściową do części klasyfikacyjnej sieci. Jak ukazuje tabela 3.14 spłaszczona warstwa pooling posiada ponad dwa razy więcej wartości aniżeli spektrogram wejściowy. Tak liczebny wektor danych zawiera w sobie wiele użytecznych informacji związanych z klasyfikacją spektrogramów, a więc również potencjalną możliwością ich prezentacji w sposób zrozumiały dla klienta. Jednakże efektywna prezentacja danych zakłada opis zjawiska w sposób nieskomplikowany. Tak, aby osoba nie mająca wiedzy domenowej związanej z analizą danych potrafiła właściwie zinterpretować przedstawione hipotezy. W tym celu dokonano redukcji wymiarowości danych pochodzących z warstwy flatten. Zdecydowano, aby



Rysunek 3.15: Schemat przetwarzania spektrogramu na punkt na płaszczyźnie 2D.

dokonać rzutowania wektora 9216 cech na płaszczyznę dwuwymiarową. W celu redukcji wymiarowości danych wykorzystano algorytm PCA [127] oraz tying weights stacked autoencoder [64]. Następnie porównano efektywność ich działania. Wytrenowano algorytmy KNN [118] dla danych po kompresji. Następnie dla każdego z KNN wyznaczono metrykę balanced accuracy [40]. Wyższa wartość metryki oznacza, że dane po kompresji na płaszczyznę 2D są łatwiej separowalne. Wszystkie algorytmy zostały dopasowane do danych syntetycznych i zapisane na potrzeby późniejszego przetwarzania danych rzeczywistych. Schemat przetwarzania spektrogramów na punkt na płaszczyźnie 2D zaprezentowano na rysunku 3.15.

Analiza obciążenia stałego

W pierwszej kolejności zbadano jakość i przejrzystość rzutowania cech spektrogramów dla obciążenia stałego. Wyniki porównania znajdują się w tabeli 3.21 oraz 3.22.

Bez dotrenowania modeli na danych rzeczywistych dokonano próby ekstrakcji cech na spektrogramach pochodzących z eksperymentu pomiarowego. Jako że najlepszą przejrzystość płaszczyzny 2D zgodnie z tabelą 3.21 zapewnia PCA bez preprocessingu danych, wykorzystując algorytm PCA bez preprocessingu danych z warstwy flatten dokonano redukcji wymiarowości. Efekt działania algorytmu widoczny jest na rysunku 3.16. Gołym okiem trudno rozróżnić gdzie na płaszczyźnie znajduje się punkt pracy związany z danym scenariuszem. Natomiast wyświetlenie wykresu dla konkretnej wartości obciążenia umożliwia identyfikację punktu pracy oraz stanu pompy. Przykładem takiej charakterystyki jest rysunek 3.17. W rozdziale 3.2.3 omówiony zostanie mechanizm predykcji uszkodzeń na podstawie śledzenia punktu pracy na płaszczyźnie 2D.

Zakres częstotliwości	Alg. redukcji wymiarowości	Preprocessing	Balanced accuracy
0-307Hz	PCA	None	0.8
0-307Hz	PCA	normalize	0.78
0-307Hz	PCA	standarize	0.76
3750-4057Hz	PCA	None	0.66
3750-4057Hz	PCA	standarize	0.66
2500-2807Hz	PCA	None	0.65
2500-2807Hz	PCA	standarize	0.65
3437-3745Hz	PCA	None	0.64

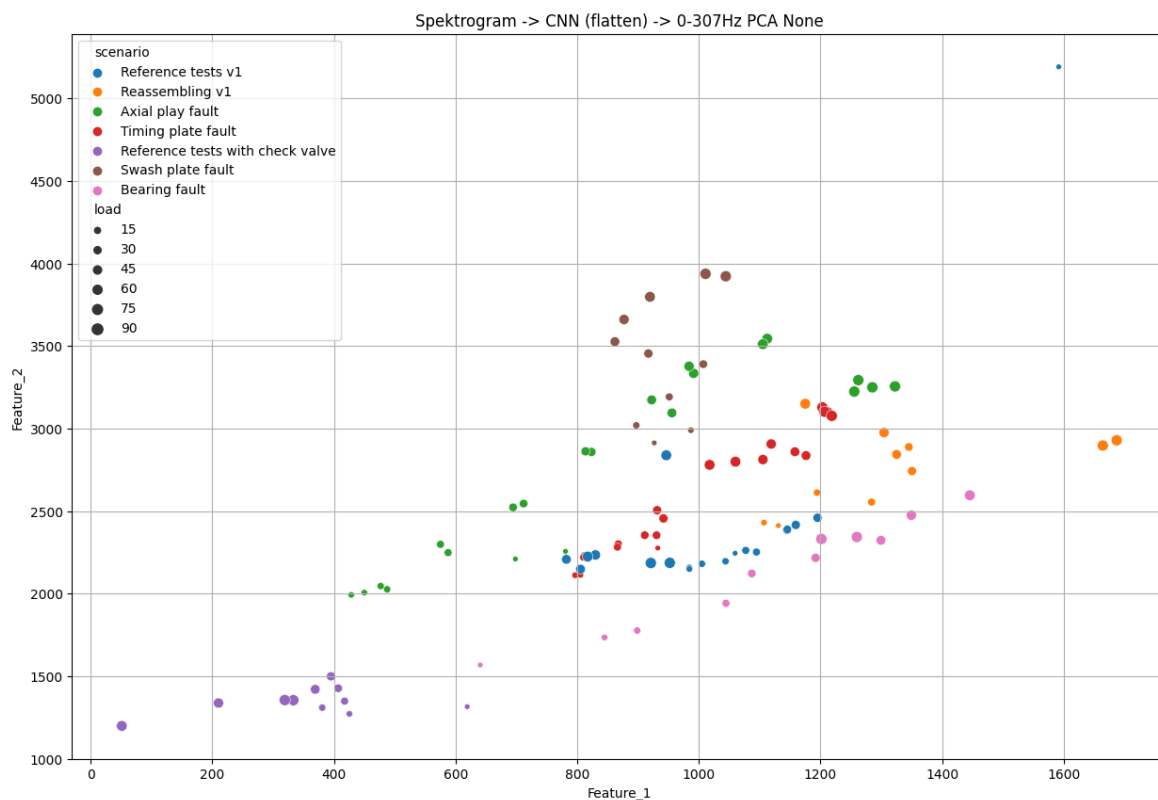
Tabela 3.21: Porównanie jakości działania algorytmów KNN na danych rzeczywistych.

Dane z warstwy flatten pochodzą z sieci wytrenowanej wyłącznie na danych syntetycznych. Wybrano kilka cechujących się najwyższą wartością metryki balanced accuracy

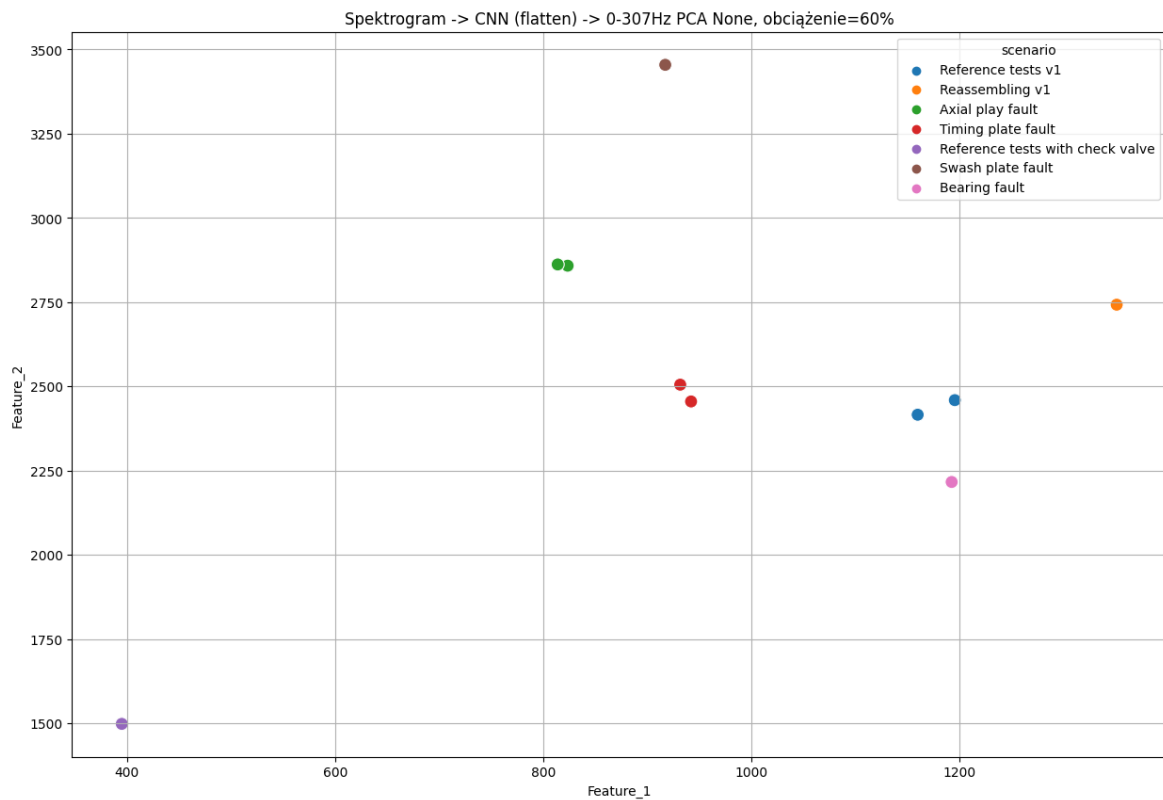
Zakres częstotliwości	Alg. redukcji wymiarowości	Preprocessing	Balanced accuracy
0-307Hz	PCA	normalize	0.9
0-307Hz	PCA	standarize	0.79
0-307Hz	PCA	None	0.78
0-307Hz	TWSAE	None	0.71
3125-3432Hz	PCA	normalize	0.67
3750-4057Hz	PCA	None	0.66
1250-1557Hz	PCA	None	0.65
3125-3432Hz	TWSAE	normalize	0.65

Tabela 3.22: Porównanie jakości działania algorytmów KNN na danych rzeczywistych.

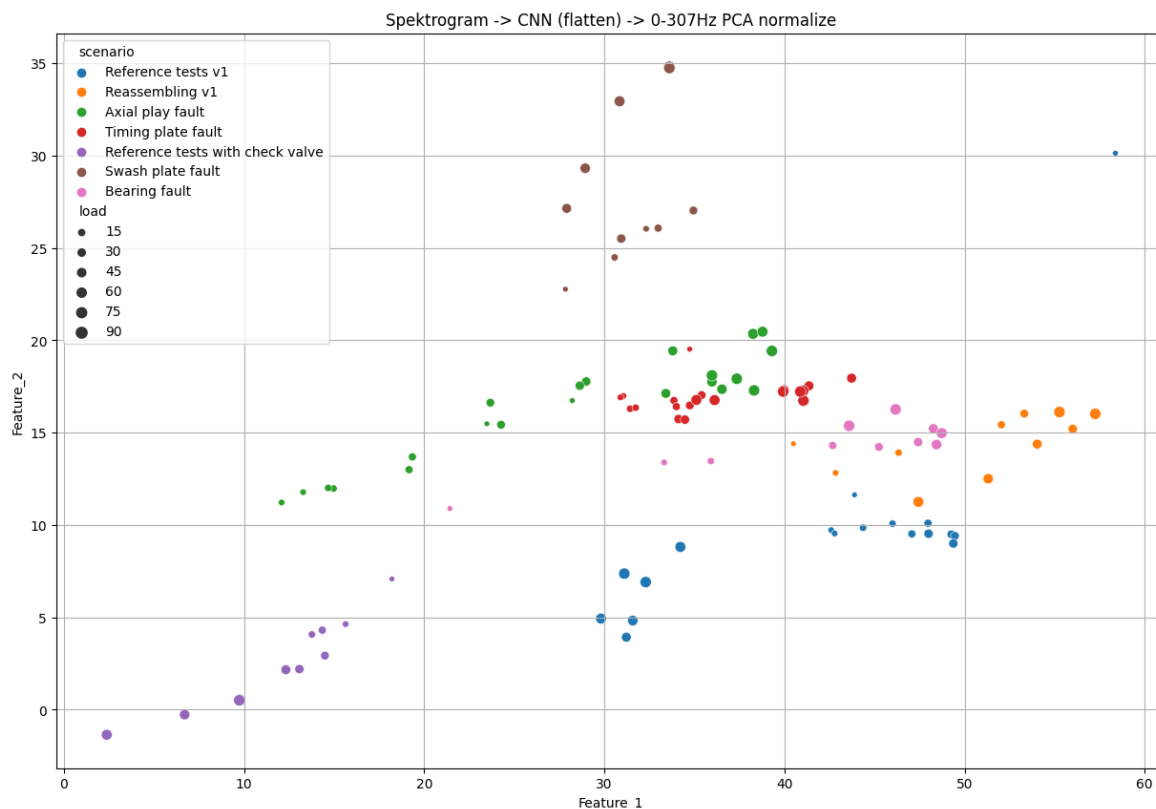
Dane z warstwy flatten pochodzą z sieci dotrenowanej z wykorzystaniem rzeczywistych spektrogramów. Wybrano kilka cechujących się najwyższą wartością metryki balanced accuracy



Rysunek 3.16: Naniesione punkty po ekstrakcji cech dla obciążeń 10...100%. Wektor wejściowy do algorytmu kompresji pochodzi z sieci wytrenowanej wyłącznie na danych syntetycznych.

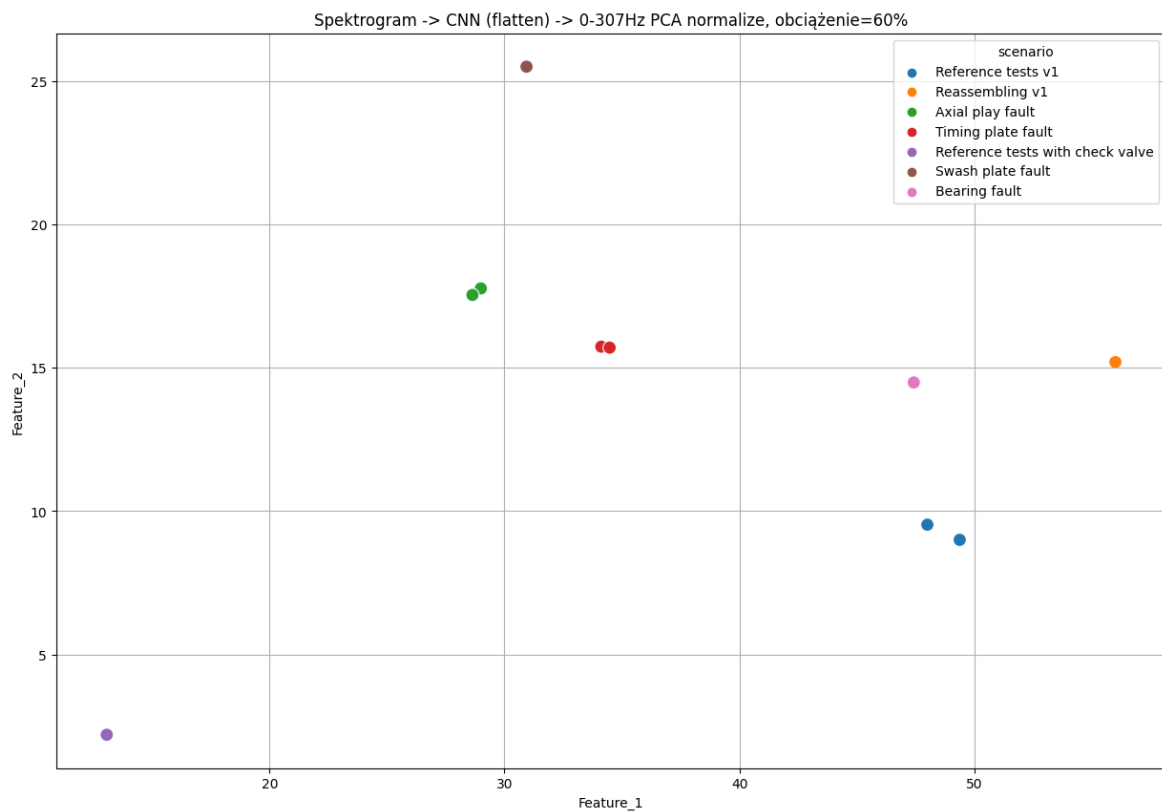


Rysunek 3.17: Naniesione punkty po ekstrakcji cech dla obciążenia 60%. Wektor wejściowy do algorytmu kompresji pochodzi z sieci wytrenowanej wyłącznie na danych syntetycznych.



Rysunek 3.18: Naniesione punkty po ekstrakcji cech dla obciążeń 10...100%. Wektor wejściowy do algorytmu kompresji pochodzi z sieci dotrenowanej na danych rzeczywistych.

Dokonano również rzutowania spektrogramów na płaszczyznę 2D z wykorzystaniem warstwy flatten z sieci dotrenowanej na spektrogramach rzeczywistych. Należy podkreślić, że chodzi o sieci, których wszystkie warstwy zostały dotrenowane, czyli zarówno te wchodzące w skład ekstraktora cech, jak i klasyfikatora. Zestawienie w tabeli 3.22 ukazuje, że najbardziej przejrzystą płaszczyznę można otrzymać stosując transformację PCA oraz stosując uprzednio normalizację wektorów z warstwy flatten. Podobnie jak na wykresie 3.16 również na wykresie 3.18 gołym okiem ciężko rozróżnić niektóre z punktów według klasy do której przynależą. Natomiast wyświetlenie płaszczyzn 2D dla każdego z obciążeń pompy osobno, umożliwia łatwą identyfikację każdego ze stanów charakteryzujących pompę. Rysunek 3.19 jest przykładem tego, że dla konkretnego rodzaju uszkodzenia w przejrzysty sposób można odróżnić każdy ze stanów pompy.



Rysunek 3.19: Naniesione punkty po ekstrakcji cech dla obciążenia 60%. Wektor wejściowy do algorytmu kompresji pochodzi z sieci dotrenowanej na danych rzeczywistych.

Wskaźniki jakości

To w jaki sposób uczone mają być modele zależy od ich przeznaczenia. W praktyce przemysłowej rozważany jest kompromis pomiędzy precyzją a czułością klasyfikatora. W niektórych procesach przemysłowych należy potraktować poważnie każde, nawet mało prawdopodobne, ostrzeżenie o błędzie. Wielkością której maksymalizacja w takim wypadku jest odpowiednia w celu spełnienia wymagań stawianych modelowi jest czułość [124]. Naturalnie nie należy przy tym całkowicie porzucić informacji związanej z precyzją klasyfikatora. Każdy bowiem klasyfikator o zerowej precyzji cechuje się doskonałą czułością. Inne procesy przemysłowe cechują się taką specyfiką, że dopuszczalne jest ich działanie podczas którego niekiedy występują błędy. Zaistniałe błędy, owszem należy wskazywać, jednakże aby dokonać ingerencji w proces należy mieć pewność, że jest ku temu powód. Monitoring tego rodzaju procesów wymaga maksymalizacji precyzji klasyfikatora.

Obraną metryką podczas porównywania modeli klasyfikacji binarnej i wieloklasowej była wartość F1. Zgodnie z definicją [67] zawiera ona w sobie informację o wartości *precision* (precyzji) oraz *recall* (czułości) danego klasyfikatora. W celach poglądowych jest ona zatem odpowiednia aby dać pewien pogląd na to, jak kształtuje się proporcja czułości względem precyzji dla konkretnego modelu. Rozbieżności między wynikami F1 dla klasyfikacji binarnej i wieloklasowej wynikają ze sposobu liczenia metryki F1 przez bibliotekę sklearn. Zakłada on obliczanie F1 jako średniej ważonej dla wartości F1 dla każdej z klas. Wagą dla każdej z klas jest jej liczebność. Może się zatem okazać, że wyższa jest wartość metryki F1 dla klasyfikacji wieloklasowej niż binarnej. Nie są to natomiast wartości znacząco różne.

Porównywanie jakości klasyfikatorów, których działanie oparte jest na wyznaczeniu funkcji decyzyjnej może być dokonane z wykorzystaniem krzywych ROC [38] oraz wynikających zeń wartości AUC [43]. Wartość AUC określa jak elastyczny jest klasyfikator w kontekście kompromisu precyzji i czułości. Dla wartości równej 1 nie trzeba rozpatrywać kompromisu, bowiem klasyfikacja przebiega idealnie. Aby obliczyć AUC należałoby wyznaczyć ROC dla klasyfikatora typu One vs Rest (OvR) i dla każdego klasyfikatora, czyli dla każdej krzywej ROC, można byłoby obliczyć AUC [64]. Jednakże bardziej przejrzystym przedstawieniem jakości klasyfikacji jest macierz pomyłek. Z tego względu nie wykreślano krzywych ROC.

Macierz pomyłek jest narzędziem intuicyjnym dla osoby posiadającej wiedzę domenową z zakresu analizy danych. Dla klienta jest ona trudniejsza w interpretacji niż

Pompa uszkodzona	Liczebność	Pompa sprawna	Liczebność
Axial play fault	16561	Reassembling v1	17101
Bearing fault	11909	Reassembling v2	15561
Swash plate fault	15901	Reference tests v1	13321
Timing plate fault	14761	Reference tests v2	18301
		Reference tests with check valve	16201
Suma	59132	Suma	82025

Tabela 3.23: Zestawienie liczebności zbioru danych z podziałem na charakter przeprowadzonego eksperymentu.

pojedyncze wskaźniki liczbowe takie jak F1 lub AUC. Jednakże wypracowanie macierzy pomyłek jest niezbędne, aby wyjaśnić które z klas najłatwiej, a które najtrudniej klasyfikować. Dzięki czemu odbiorca ma większą świadomość tego, jak działa model i jakiego rodzaju błędów można się po nim spodziewać. Wskazując odpowiednie komórki macierzy można również znaleźć kompromis pomiędzy precyzją a czułością klasyfikatora.

Podczas wyboru wskaźników jakości rozważono problem niezbilansowania klas. W zbiorze danych związanych z pompą sprawną było 82025 rekordów, zaś z pompą uszkodzoną 59133 rekordów. Są to rekordy obejmujące sekundę pomiarów zmiennych procesowych i wibracji. Liczebność przykładów w zbiorze danych ukazuje tabela 3.23. W przypadku klasyfikacji binarnej problem niezbilansowania klas nie jest zbyt problematyczny. Jednakże rozpatrując zagadnienie wieloklasowe proporcje drastycznie się zmieniają. Każda z klas związanych z pompą uszkodzoną jest średnio osiem razy mniej liczebna niż zbiór pomiarów związanych z pompą sprawną. Tym samym powodując, że należy ostrożnie podejść do wyboru i obliczania wskaźników jakości. Podobnie należy postąpić w przypadku ewaluacji modeli wytrenowanych z wykorzystaniem spektrogramów. Jak ukazuje tabela 3.16 problem niezbilansowania klas również występuje.

W kontekście wyboru najbardziej przejrzystej charakterystyki 2D wykorzystano metrykę balanced accuracy [40]. Zakłada ona obliczanie wskaźnika miarodajnego na potrzeby klasyfikacji niezbalansowanych zbiorów danych. Takim właśnie jest analizowany zbiór. Wysoka wartość tej metryki umożliwiła odnalezienie takiego rozkładu punktów na płaszczyźnie, który jest zrozumiały i przejrzysty dla odbiorcy końcowego.

3.2.3 Predykcja

Biznesowo i wdrożeniowo istotniejsza od samej diagnozy stanu maszyny jest predykcja horyzontu czasowego po którym ulegnie ona awarii. Dzięki takiej informacji można z wyprzedzeniem zaplanować przegląd techniczny i serwis urządzenia. Zarówno przeglądy jak i serwisy należy bowiem planować z wyprzedzeniem. Co więcej, w przypadku niektórych części zamiennych czas oczekiwania na ich dostawy jest długi [24]. W przypadku fabryk jest to szczególnie ważne z powodu strat finansowych poniesionych z tytułu potencjalnych awarii i przestojów w produkcji. W przypadkach niezwiązanych z produkcją straty finansowe spowodowane przestojem urządzenia mogą być powodem opóźnień w świadczeniu usług. Niezależnie od przeznaczenia maszyn, z biznesowego punktu widzenia pożądana jest dokładana informacja pozwalająca planować przestoje w ich pracy.

Powszechnie omawianym wskaźnikiem związanym z predykcją awarii jest RUL (ang. remaining useful life). Jest to informacja o czasie pozostałej bezawaryjnej pracy urządzenia [105]. Najczęściej jest ona podawana w dniach z uwagi na dynamikę procesów w skład których wchodzi monitorowane maszyny oraz dokładności estymacji czasu potrzebnego na zaplanowanie serwisów i dostaw części zamiennych. Obliczanie RUL może bazować na zmiennych różnego rodzaju, nie tylko tych związanych z wibracjami [30]. W rozdziale 3.1.2 przedstawiono koncepcyjnie jak podczas wdrożeń wyznaczano RUL bazując na wartości prędkości skutecznej drgań. Więcej przykładów obliczania RUL za pomocą ekstrapolacji trendu badanego wskaźnika można znaleźć w [80].

Ekstrapolacja trendu wymaga wprawdzie jego wyznaczenia. W przypadku analizy maszyn w [146] proponowano pomiar w długim horyzoncie czasowym. Tak, aby obserwować postępujące w czasie uszkodzenie. W miarę upływu czasu można było zaobserwować degradację stanu maszyny, a co za tym idzie zmianę wartości badanych wskaźników. Takie podejście zakłada posiadanie pomiarów pośrednich pomiędzy stanem dobrym, a stanem wskazującym na uszkodzenie. W przypadku braku pomiarów pośrednich należy estymować RUL w inny sposób. W literaturze proponowano wykorzystanie uczenia półnadzorowanego oraz dwukierunkowych rekurencyjnych sieci neuronowych opisane w publikacji [142]. Artykuł [117] opisuje metody statystyczne i probabilistyczne szacowania wartości RUL. Ponadto w powyższej publikacji zwrócono szczególną uwagę na obliczanie wartości RUL w zależności od rodzaju i zakresu dostępnych danych dla monitorowanej maszyny. Propozycję innych sposobów szacowania wartości RUL przeprowadzono w niniejszej pracy w podrozdziale 3.2.3.

Generacja stanów pośrednich

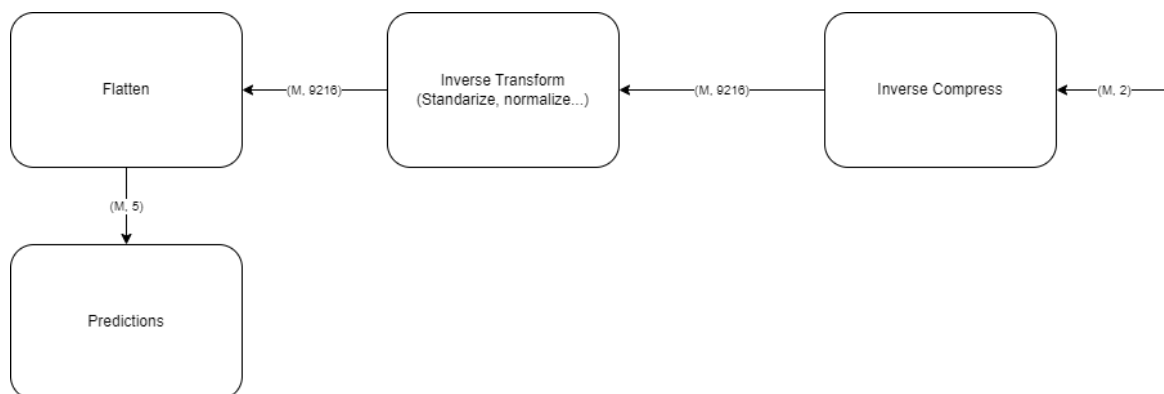
Pomiary zebrane na pompie wielotłoczkowej w PONAR Wadowice S.A. nie zawierają stanów pośrednich między maszyną sprawną a uszkodzoną. Wyznaczenie trendu poprzez ekstrapolację wykorzystując pomiary rzeczywiste jest niemożliwe. Możliwa jest natomiast sztuczna generacja stanów pośrednich z wykorzystaniem mechanizmów wypracowanych podczas klasyfikacji w ramach rozdziału 3.2.2. Należy również mieć na uwadze, że docelowo rozwiązanie ma być przejrzyste dla klienta. Zatem na sposób przedstawienia wyników również kładziony jest odpowiedni nacisk. Płaszczyzna 2D jest wystarczająco dobrym narzędziem do ilustracji aby posłużyła za podstawę dalszych rozważań.

Śledzenie punktu pracy na płaszczyźnie 2D - transformacja odwrotna

Zaprezentowana w rozdziale 3.2.2 metodyka zakłada transformację spektrogramu w punkt na płaszczyźnie 2D. Odbywa się to zgodnie ze schematem przedstawionym na rysunku 3.15. Generacja stanów pośrednich odbywać się może na zasadzie transformacji odwrotnej. Punkt z płaszczyzny 2D poddać można operacji zwiększenia wymiarowości do przestrzeni o wymiarze 9216 z wykorzystaniem odwrotnej transformacji algorytmu kompresji. Jednakże, spożytkowany na potrzeby zwiększenia wymiarowości algorytm PCA w pierwz został dopasowany do oryginalnych danych na potrzeby zmniejszenia liczby wymiarów co niesie ze sobą implikacje w związku z transformacją odwrotną. Mianowicie, operacja redukcji wymiarowości powoduje w większości przypadków brak możliwości idealnego odtworzenia pierwotnego zbioru danych korzystając z danych po kompresji [98] [109].

Mając na uwadze wskazane powyżej niedoskonałości algorytmu kompresji przeprowadzono próbę wykorzystania transformacji odwrotnej. Wektor powstałych wskutek transformacji odwrotnej 9216 liczb podano na wejście warstwy flatten, a zatem na wejście części będącej klasyfikatorem w ramach CNN. Klasyfikator dokonuje przetworzenia wektora liczb w wyniku którego otrzymywany jest kolejny wektor. Znajduje się w nim 5 wartości. Każda z tych wartości pochodząca z zakresu $< 0, 1 >$. Reasumując, część będąca klasyfikatorem w ramach CNN dokonuje klasyfikacji, ale nie na rzeczywistych spektrogramach, tylko na wektorze cech powstałych z punktu na płaszczyźnie 2D. Schematycznie ukazuje to rysunek 3.20.

Przeprowadzono analizę poprawności zaproponowanego podejścia dla modelu i al-



Rysunek 3.20: Przetwarzanie punktu z płaszczyzny 2D na wektor pewności klasyfikacji. Schemat jest bardzo zbliżony do rysunku 3.15 natomiast z racji tego, że prezentuje transformację odwrotną, należy analizować go od prawej do lewej strony.

gorytmu kompresji skutkującego najbardziej przejrzystą płaszczyzną 2D w kontekście prezentacji punktów pracy z rozróżnieniem na stan maszyny. Zgodnie z porównaniem zamieszczonym w tabeli 3.22 najlepszy wykorzystany model związany był z zakresem częstotliwości 0-307Hz, zaś algorytmem kompresji było PCA z normalizacją danych wejściowych. Rysunek 3.18 ukazuje płaszczyznę 2D powstałą z wykorzystaniem takiego rodzaju kompresji i modelu związanego ze wskazanym zakresem częstotliwości.

Punkty już istniejące na płaszczyźnie 2D, pochodzące z rzeczywistych spektrogramów, poddano transformacji odwrotnej algorytmu PCA. Opis transformacji odwrotnej algorytmu PCA zaprezentowano w [106]. Szczegóły dotyczące błędów związanych z rekonstrukcją danych, które wynikają z zastosowania algorytmu PCA znajdują się w [63]. Transformacja odwrotna, choć niewątpliwie powoduje utratę informacji, to w publikacji [108] odnaleźć można jej zastosowanie dające dobre wyniki. Przystąpiono więc do dalszych analiz.

Z punktów o dwu współrzędnych otrzymano wektory o 9216 wartościach. Następnie na bazie powstałych wektorów cech dokonano klasyfikacji z wykorzystaniem wytrenowanej sieci CNN. W efekcie otrzymano szereg pięcioelementowych wektorów, z których każdy porównano z jego rzeczywistym odpowiednikiem. Rzeczywistym, czyli powstałym w wyniku klasyfikacji rzeczywistego spektrogramu odpowiadającego punktowi na płaszczyźnie 2D. Porównanie odbywało się dwójako. Odnosiło się do tego, czy klasyfikator przyporządkował odpowiednią klasę do zadanego przypadku, oraz jak bardzo różni się wektor cech będących wyjściem z klasyfikatora dla sieci zasilanej spektrogramami oraz wektorem 9216 cech pochodzących z transformacji odwrotnej punktu

Funkcja	Opis	Wartość
SCCE	Oryginalne predykcje tzn. funkcja kosztu sieci CNN	5.86e-02
SCCE	Ekspansja 2 → 9216, następnie klasyfikacja	9.67
CCE	Podobieństwo wektorów pewności	9.65
F1	Ekspansja 2 → 9216, następnie klasyfikacja	0.22

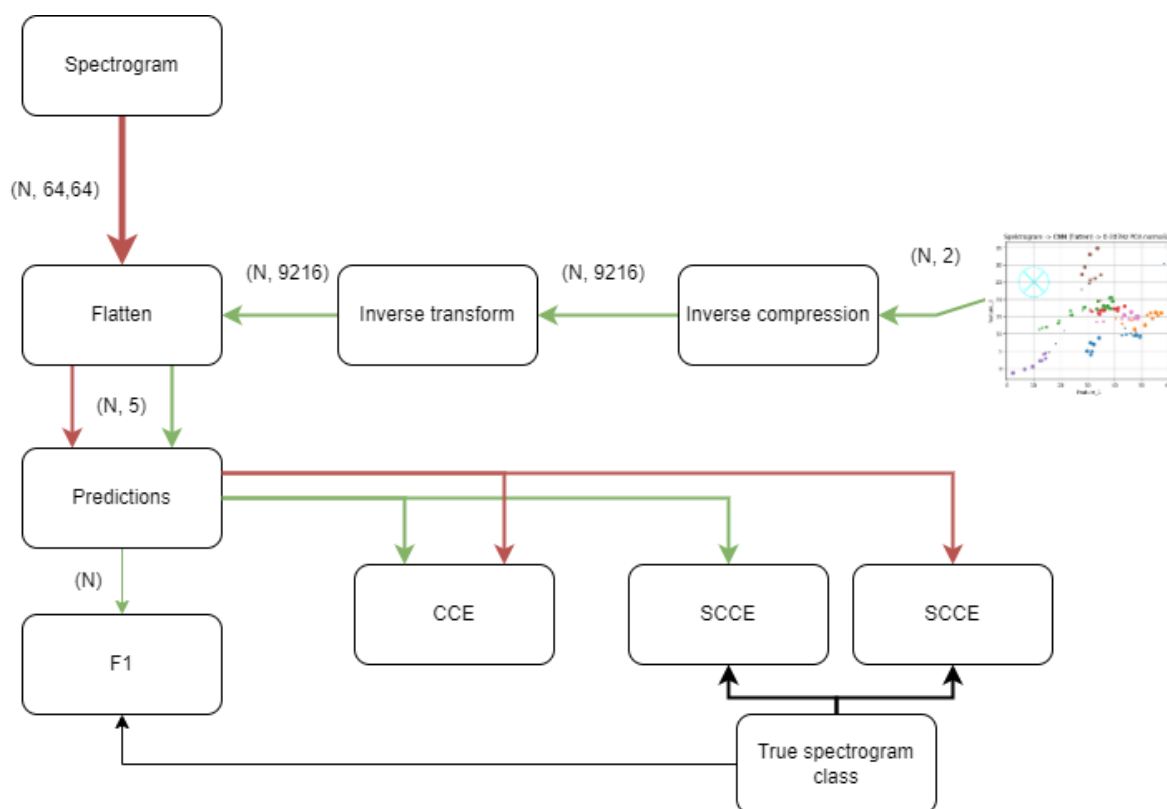
Tabela 3.24: Zestawienie wartości służących ocenie jakości transformacji odwrotnej punktu z płaszczyzny 2D na wektor pewności klasyfikacji.

skojarzonego ze spektrogramem na płaszczyźnie 2D.

Kwantyfikacja poprawności klasyfikacji została przeprowadzona z wykorzystaniem wartości F1. Została ona obliczona nie w oparciu o rzeczywistą przynależność, ale w oparciu o przynależność do klasy, którą dla danego spektrogramu przewiduje sieć CNN. Kolejno wykorzystano funkcję sparse categorical cross-entropy (SCCE), która jest również funkcją kosztu podczas uczenia sieci CNN, a którą opisano w rozdziale 3.2.2. Wartość SCCE obliczono dla przypadków gdzie danymi wejściowymi do sieci CNN były spektrogramy oraz dla przypadków, w których sieć zasilana była wektorem 9216 cech na wejściu części klasyfikującej. Dla obu przypadków obliczono ją odnosząc się do numeru klasy, z którą skojarzony jest spektrogram będący przedmiotem analizy. Obliczono także categorical cross-entropy (CCE) [49] pomiędzy wektorami pięcioelementowymi predykcji dla rzeczywistych spektrogramów oraz predykcji powstałych z zasilenia sieci CNN w części klasyfikującej wektorem 9216 cech. Zmierzono zatem podobieństwo wartości w wektorach zawierających pewność klasyfikacji.

Koncepcyjnie, w idealnym przypadku wskazywane klasy się pokrywają. Niezależnie od tego, czy zostały one wypracowane w oparciu o rzeczywiste spektrogramy, czy też o wektor cech pochodzący z transformacji odwrotnej punktu z płaszczyzny 2D. Warto podkreślić, że na potrzeby opisanego podejścia nie jest istotne to z jaką rzeczywistością klasą związany jest wektor cech, ale to czy klasyfikator wskaże taką samą klasę dla wektora pochodzącego z dwóch źródeł. Z kolei wektory wyjściowe z klasyfikatora w idealnym przypadku są jednakowe zarówno dla cech pochodzących ze spektrogramu rzeczywistego, jak i dla cech pochodzących z punktu na płaszczyźnie 2D. Tabela 3.24 zawiera zestawienie opisanych powyżej wskaźników pozwalających ocenić jakość transformacji odwrotnej. Schematycznie proces wyznaczania wspomnianych wartości SCCE, CCE oraz F1 ukazany jest na rysunku 3.21.

Jak ukazuje tabela 3.24 proces przekształcenia punktu z płaszczyzny 2D w celu jego klasyfikacji jest bardzo słabej jakości pod kątem wszystkich obliczonych metryk.



Rysunek 3.21: Schemat obliczania metryk służących ocenie jakości transformacji odwrotnej. Im grubsza strzałka, tym większa jest wymiarowość danych, które przezeń przechodzą.

Informuje o tym niska wartość F1. Niska jakość klasyfikacji widoczna jest też w rozbieżności wartości funkcji SCCE. Jej wartość obliczona dla oryginalnych predykcji jest o ponad dwa rzędy wielkości mniejsza niż wartość SCCE związana z predykcjami wypracowanymi w oparciu o cechy powstałe z punktu na płaszczyźnie 2D skojarzonego z rzeczywistym spektrogramem. Wysoka jest wartość funkcji CCE, a zatem podobieństwo pomiędzy wektorami pewności pochodzącymi z klasyfikacji jest niewielkie.

Nie jest możliwe zatem wykorzystanie analizowanego podejścia w celu generowania stanów pośrednich na płaszczyźnie 2D. Co za tym idzie, nie jest możliwe aby wspomagając się tym procesem szacować pozostały czas użytkowania. Należy rozważyć inne koncepcje.

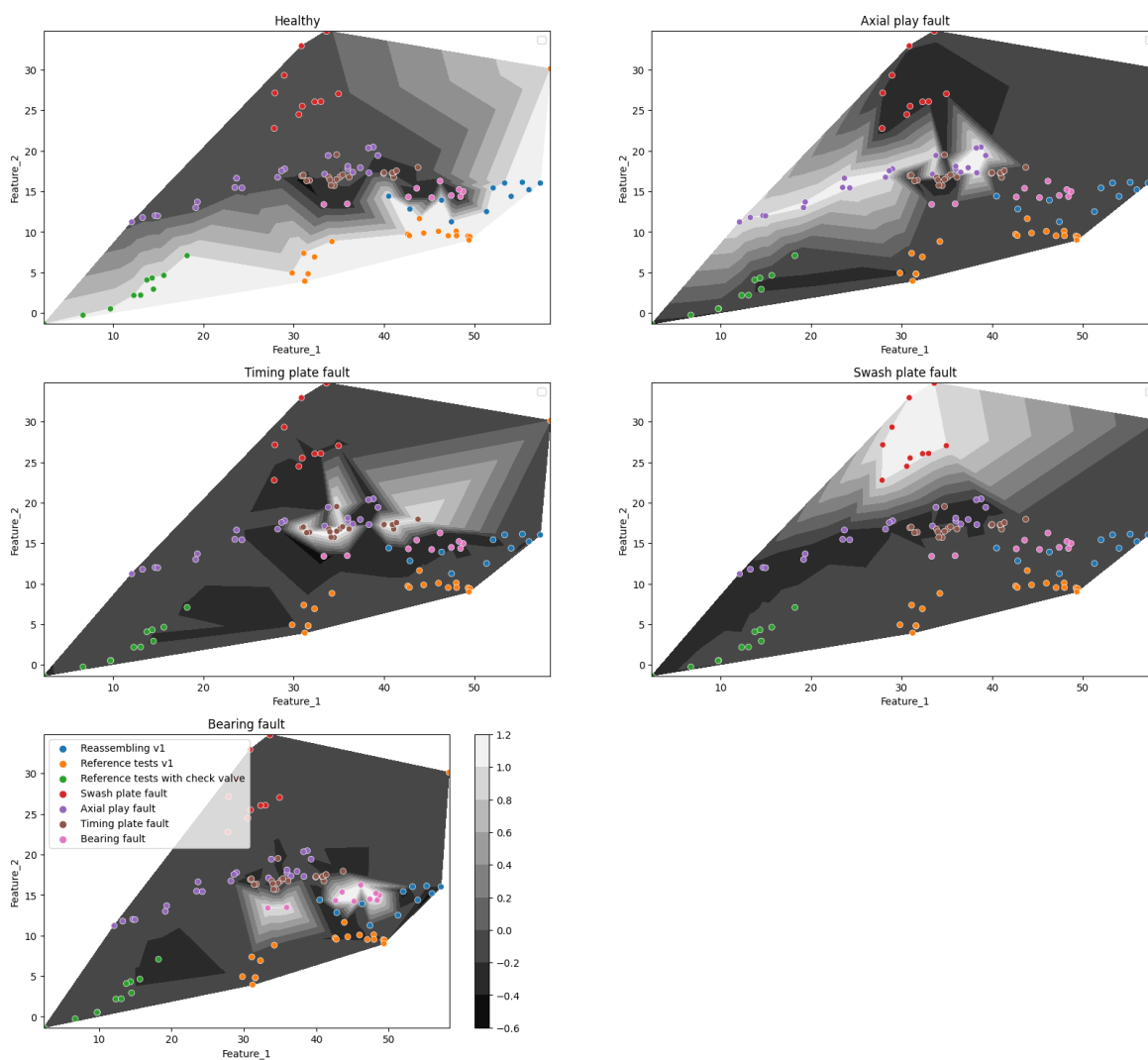
Śledzenie punktu pracy na płaszczyźnie 2D - regresja

Generowanie stanów pośrednich odbywać może się na zasadzie syntezy przestrzeni 3D zbudowanej w oparciu o cechy uzyskane po zastosowaniu algorytmu kompresji, a tworzące płaszczyznę 2D. Trzecim wymiarem może w tym wypadku być wartość związana z wartością pewności klasyfikacji otrzymanej z neuronów wyjściowych sieci CNN dla poszczególnych klas. Docelowo pożądanym jest uzyskać gęstą siatkę punktów na płaszczyźnie 2D dopasowaną do dostępnych pomiarów. Problem ten można sprowadzić do zadania regresji. Dzięki czemu będzie można przeprowadzić szacowanie pozostałego bezawaryjnego czasu pracy maszyny, co opisano w rozdziale 3.2.3. W ramach niniejszego rozdziału wykorzystywane będą parametry algorytmu syntezy płaszczyzny najbardziej przejrzystej ukazane w tabeli 3.22. Proces doboru tych parametrów oraz syntezy płaszczyzny 2D został opisany szerzej w rozdziale 3.2.2. Z kolei proces generacji wektorów wartości pewności klasyfikacji opisano w rozdziale 3.2.2. Każda z wartości w wektorach pewności cech zostały poddane transformacji opisanej równaniem (3.1)

$$confidence_vector = \frac{1}{\log(features) + 1} \quad (3.1)$$

aby bardziej rozdzielić zgromadzone w wektorach wartości.

Rysunek 3.22 ukazuje zależności pomiędzy wartościami z wektorów pewności klasyfikacji oraz cechami z płaszczyzny 2D dla wszystkich rozróżnianych stanów badanej pompy w formie mapy konturowej estymowanej dla dostępnych punktów. Wykres 3.22 został wygenerowany z wykorzystaniem funkcji dokonującej aproksymacji na zasadzie triangulacji [8].



Rysunek 3.22: Mapy konturowe wektorów pewności klasyfikacji z podziałem na klasy. Wykreślone na podstawie wartości wektorów pewności uzyskanych z sieci CNN.

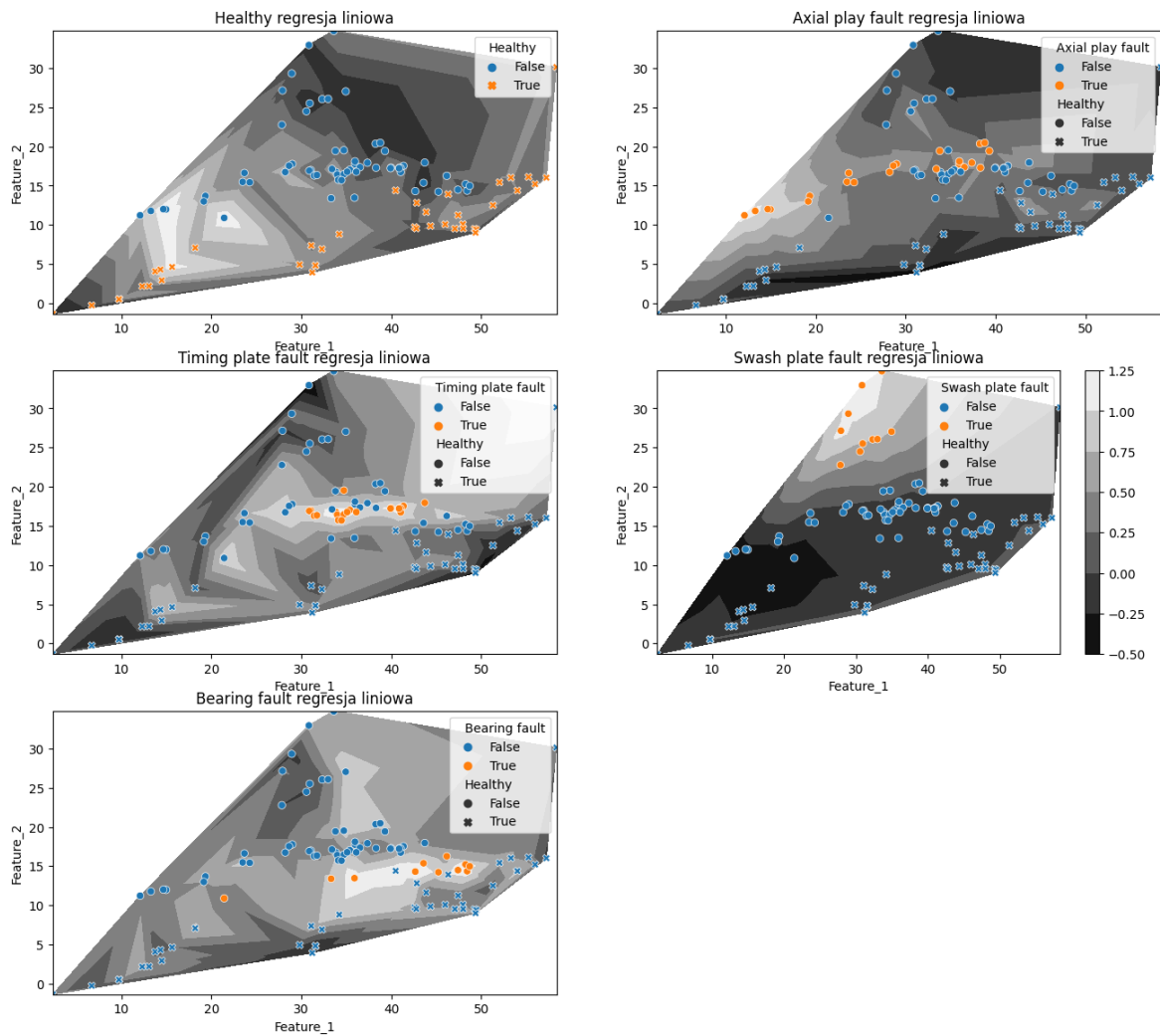
Klasa	MSE zb. testowy	R^2
Healthy	11.65	0.06
Axial play fault	0.18	0.29
Timing plate fault	0.14	0.37
Swash plate fault	0.02	0.82
Bearing fault	0.09	0.15

Tabela 3.25: Zestawienie jakości dopasowania przestrzeni 3D dla modelu typu regresja liniowa.

Analizując rysunek 3.22 można zaobserwować wyraźnie odizolowane obszary wskazujące na wysoką wartość pewności klasyfikacji dla każdej z wyróżnionych klas. Dokonano zatem próby dopasowania przestrzeni 3D. W pierwszej kolejności wykorzystano algorytm regresji liniowej zasilany wartościami cech z płaszczyzny 2D (oznaczone na rysunku 3.22 jako Feature_1 oraz Feature_2, tak zwane cechy pierwotne) oraz kilku powstałych z nich cech pochodnych uzyskanych po zastosowaniu nieliniowych transformacji. Konieczność wprowadzenia nieliniowości poprzez syntezę cech pochodnych umotywowane jest kształtem charakterystyk z rysunku 3.23, które to w wymiarze związanym z wartością pewności klasyfikacji są nieliniowe [88]. Przekształcenia cech przeprowadzono korzystając z kilku transformacji sugerowanych w publikacjach [70] oraz [131]. Dla każdej z cech pierwotnych są to:

1. Podniesienie do kwadratu
2. Podniesienie do sześcienu
3. Sinus
4. Cosinus

Ponadto dokonano wymnożenia cech pierwotnych ze sobą. Łącznie wektor cech związany z pojedynczym punktem z płaszczyzny 2D liczyć będzie 11 wartości. Dwie cechy pierwotne, cztery transformacje dla każdej z cech pierwotnych, ponadto wymnożenie cech pierwotnych ze sobą ($2 + 4 \cdot 2 + 1 = 11$). Dla każdego punktu z płaszczyzny 2D zostanie przeprowadzony ten sam zestaw transformacji. W efekcie algorytm regresji liniowej zasilac będą wektory o jedenastu wartościach. Tabela 3.25 ukazuje zestawienie jakości dopasowania dla każdej z klas. Wykorzystano metryki MSE (ang. Mean Squared Error) [97] oraz R^2 [50] jako miarodajne w kontekście oceny jakości rozwiązywania zadania regresji [50].



Rysunek 3.23: Mapy konturowe wygenerowane z wykorzystaniem modelu regresji liniowej z podziałem na stan pompy.

Nazwa parametru	Wartości
Liczba warstw	1, 2, 3, 4
Liczba neuronów w warstwie	5, 6, 7, 8, 10, 15, 20
Warstwa batch normalization [77] pomiędzy warstwami dense	Tak, Nie
Harmonogramowanie współczynnika uczenia za pomocą 1Cycle [119]	Tak, Nie

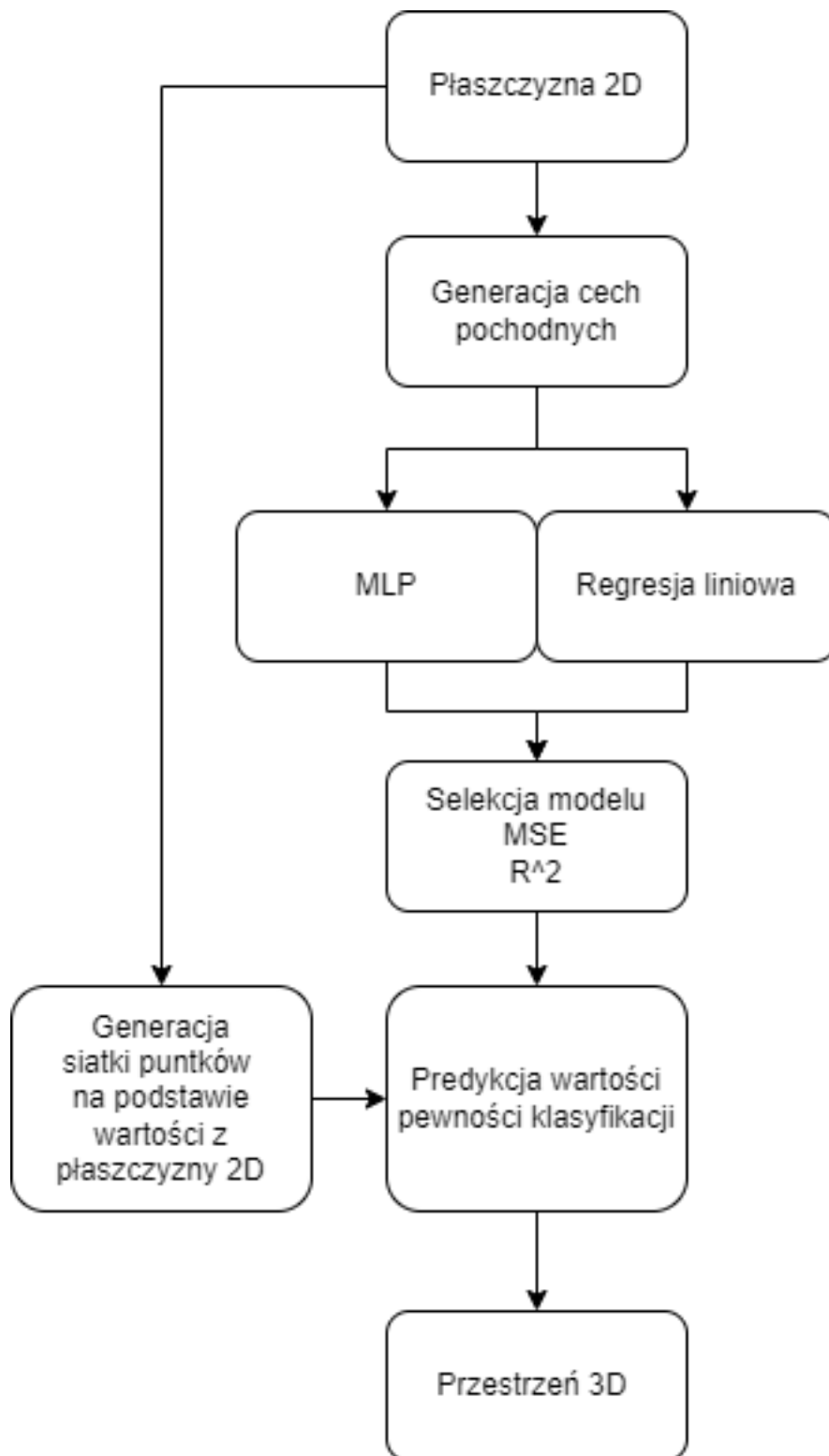
Tabela 3.26: Pula parametrów spośród których dokonano wyboru najlepszego zestawu na potrzeby dopasowania przestrzeni 3D.

Wyniki z tabeli 3.25 pokrywają się z wykresami konturowymi ilustrującymi jakość dopasowania 3.23. Najlepiej wypada model związany z uszkodzoną kołyską, najgorzej zaś model dopasowania przestrzeni 3D do wartości z transformowanego wektora cech związanego z pompą sprawną.

Regresja dotycząca uszkodzonej kołyski za pomocą prostego modelu cechuje się dość dobrym dopasowaniem. Podjęto próbę wykorzystania bardziej skomplikowanego algorytmu regresji, sieci typu wielowarstwowy perceptron (MLP) [62]. Architektura wykorzystanej sieci została wypracowana w oparciu o grid search [47]. Pula parametrów dla których dokonywano treningu i ewaluacji sieci ukazana jest w tabeli 3.26. W ramach grid search rozpatrywano nie tylko parametry związane z architekturą sieci, ale również z procesem uczenia. Zestawienie kilku modeli cechujących się najlepszą jakością modelowania przedstawiono w tabeli 3.27. Architekturę wykorzystanej sieci zaprezentowano w tabeli 3.28. Wartości wyjściowe z sieci zostały przeskalowane, zatem porównanie modeli MLP z modelami regresji liniowej odbywać się może jedynie z wykorzystaniem metryki R^2 . Trenowane sieci zasilane były tymi samymi cechami i tymi samymi cechami pochodnymi co model typu regresja liniowa. Wykorzystano regularyzację early stopping w sposób analogiczny jak w przypadku treningu sieci CNN opisanego w podrozdziale 3.2.2. Modele wskazane w tabeli 3.27 wykazują lepszą jakość klasyfikacji aniżeli model typu regresja liniowa. Na potrzeby wyznaczenia trajektorii punktu pracy wykorzystany zostanie najlepszy model MLP. Schematycznie proces syntezy modelu regresji opisany w niniejszym podrozdziale, został przedstawiony na rysunku 3.24.

Trajektoria punktu pracy na płaszczyźnie 2D

Poprzez rozwiązanie zadania regresji z wykorzystaniem sieci MLP uzyskano stany pośrednie pomiędzy punktami pracy uzyskanymi na bazie danych z pomiarów. Kolej-



Rysunek 3.24: Proces syntezy modelu regresji na potrzeby uzyskania przestrzeni 3D.

Liczba warstw	Liczba neuronów w warstwie	Batch normalization	1Cycle	MSE zb. testowy	R^2 zb. testowy
2	5	Nie	Nie	0.029	0.9
2	8	Nie	Nie	0.040	0.88
3	10	Nie	Nie	0.041	0.88

Tabela 3.27: Porównanie najlepszych modeli MLP z wyszczególnieniem ich architektury.

Warstwa (typ)	Wymiary	Liczba parametrów
dense11 (Dense)	(None, 5)	50
dense12 (Dense)	(None, 5)	30
dense13 (Dense)	(None, 1)	6
Suma		86

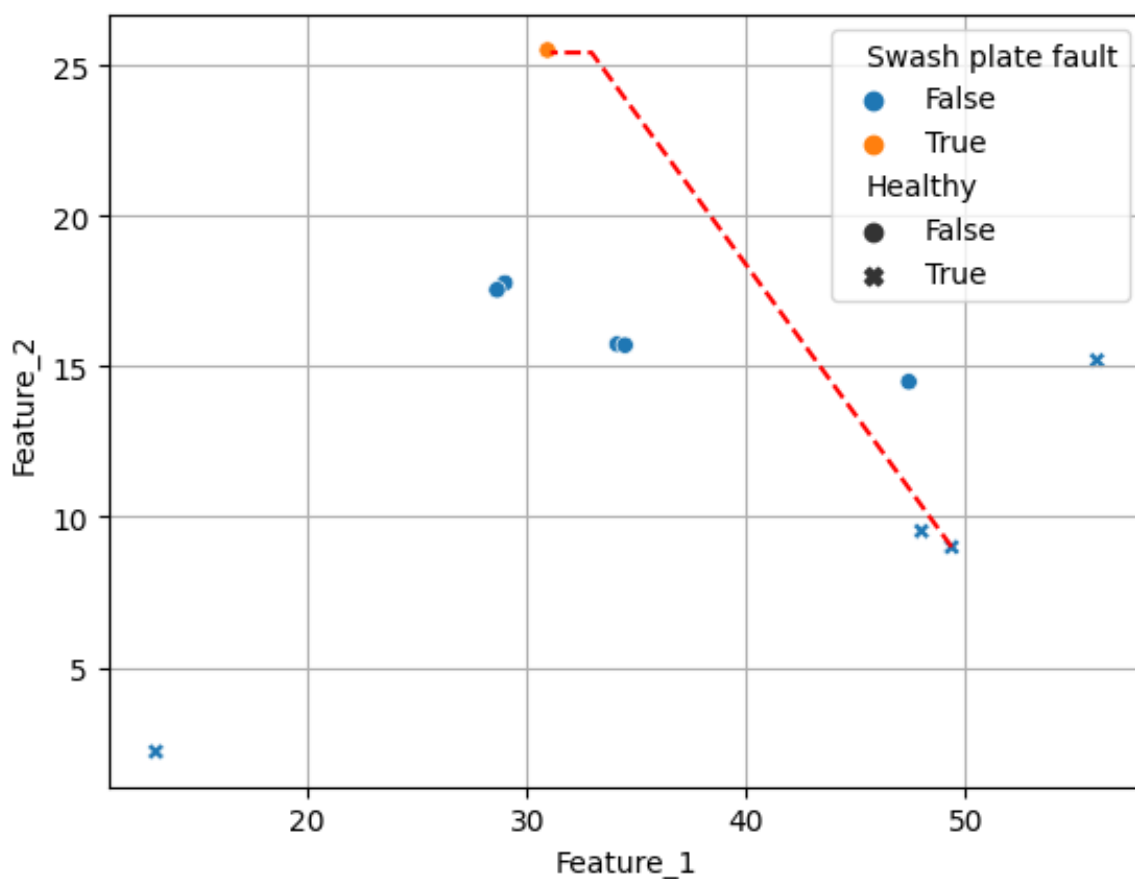
Tabela 3.28: Architektura sieci MLP o najlepszej jakości dopasowania do zbioru testowego.

nym krokiem jest wyznaczenie trajektorii pomiędzy punktem pracy określającym pompę sprawną, a punktem pracy określającym pompę uszkodzoną. Z powodów związanych z jakością modelowania omówionych w podrozdziale 3.2.3 omawiana będzie charakterystyka związana z uszkodzeniem kołyski pompy. Uzyskana trajektoria posłuży estymacji czasu pozostałego bezawaryjnego użytkowania monitorowanego urządzenia.

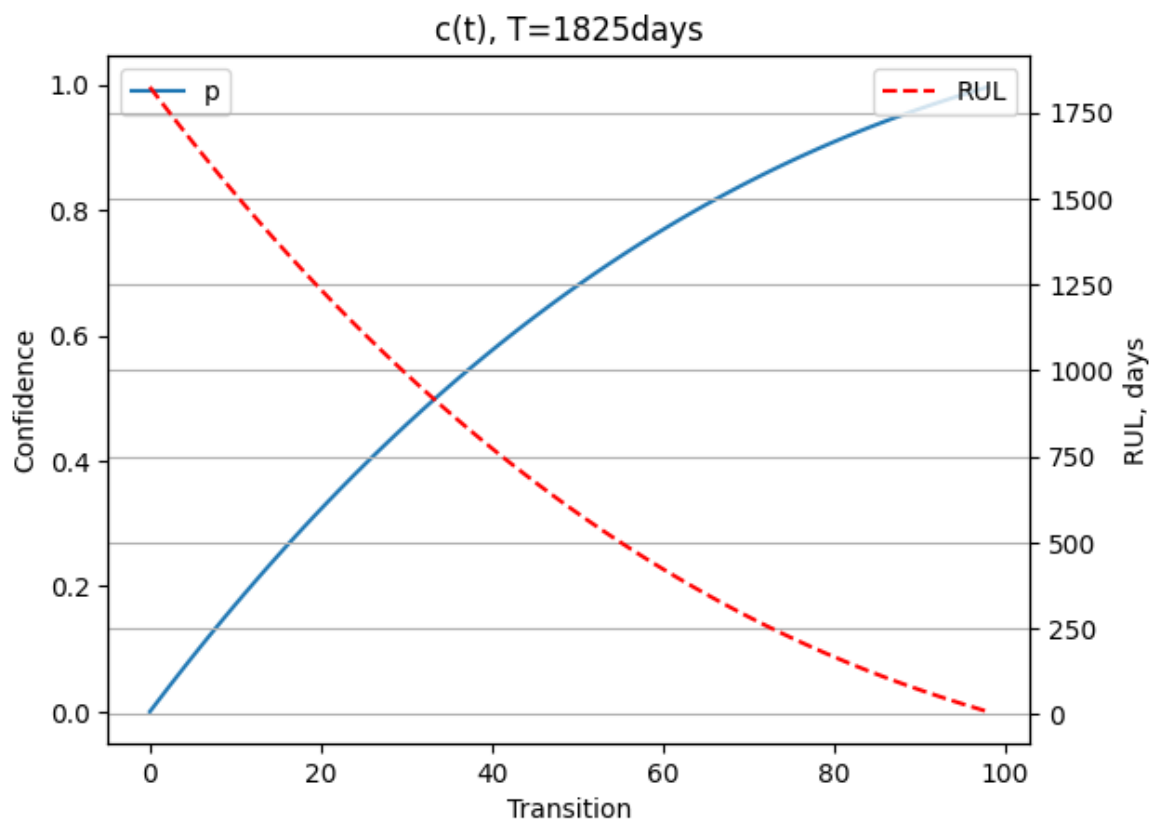
Trajektoria przejścia punktu pracy pomiędzy stanami pompy odbywać się będzie z podziałem na wartość obciążenia. Podstawą takiego działania jest założenie, że w zależności od tego w jakim reżimie pracuje pompa, inaczej będzie przebiegać jej degradacja. Krzywa łącząca punkty pracy przebiegająca pomiędzy stanem oznaczającym pompę sprawną, a stanem oznaczającym uszkodzoną kołyskę zostanie wyznaczona poprzez połączenie punktów w przestrzeni 3D. Kształt przestrzeni 3D zostanie wypracowany w oparciu o predykcje sieci MLP o najlepszej jakości dopasowania. Przestrzeń 3D będzie miała charakter dyskretny, będzie określona jedynie w wyznaczonych punktach. Pomiedzy punktami w przestrzeni 3D zostanie zastosowana interpolacja liniowa. Sposób jej wyznaczania trajektorii zostanie przeprowadzony za pomocą następującego algorytmu. Zakłada on punkt początkowy w miejscu na charakterystyce, w którym pompa znajduje się w stanie sprawnym. Następnie punkt początkowy przesuwany jest o jedną jednostkę miary na osi X oraz o jedną jednostkę miary na osi Y. Poprzez jednostkę miary rozumiana jest minimalna odległość pomiędzy punktami na założonej siatce punktów w ramach wybranej osi. Iteracyjne przeprowadzanie przesuwania się po siatce punktów na obu osiach przeprowadzane jest do momentu, w którym na jednej z

osi zostanie osiągnięta współrzędna skojarzona z tą osią równa wartości współrzędnej punktu związanego z uszkodzoną kołyską. W takim wypadku przesuwanie się po siatce punktów odbywa się wyłącznie po drugiej z osi, do momentu aż sukcesywnie przesuwany punkt osiągnie współrzędne takie same jak punkt związany z uszkodzoną kołyską. Ilustrację tak prowadzonej krzywej przeprowadzono na rysunku 3.25. Na rysunku 3.25 zaobserwować można to, że większą liczbę kroków algorytm przeprowadził na osi X. Przedstawiony sposób syntezy trajektorii na płaszczyźnie 2D cechuje się dużą prostotą. Co za tym idzie, kształt wynikowej trajektorii również cechuje się dużą prostotą. Tym niemniej, jest to jedna z możliwych dróg, które punkt pracy może przebyć na płaszczyźnie 2D. Istotny przy wyznaczaniu trajektorii jest również fakt, że w ramach wymiaru związanego z pewnością klasyfikacji trajektoria powinna wskazywać coraz większe wartości. Takie stwierdzenie oparte jest na założeniu, że w miarę przesuwania się od punktu pracy związanego z pompą sprawną do punktu oznaczającego uszkodzenie, algorytm predykcji wartości pewności klasyfikacji powinien wykazywać się coraz większą pewnością predykcji uszkodzenia. Wyznaczona trajektoria posiada taką własność. Jest to dobry punkt wyjścia do tworzenia trajektorii w inny sposób. Na przykład w oparciu o heurystyki, które zakładałyby przeszukiwanie poprzez przemieszczanie się punkt po punkcie po siatce punktów na płaszczyźnie 2D w określony sposób. Przykładem heurystyki możliwej do zaadaptowania i wykorzystania jest depth-first search [54] lub breadth-first search [87].

Na początku rozdziału 3.2.3 omówiony został wskaźnik RUL. Znając trajektorię przemieszczania się punktu pracy na płaszczyźnie 2D oraz wartości w przestrzeni 3D w wymiarze związanym z pewnością klasyfikacji dla punktów wchodzących w skład trajektorii, można szacować wartość RUL w oparciu o pewne założenia. Kolejne fragmenty trajektorii punktu pracy wyznaczone są iteracyjnie jak zostało to opisane na początku niniejszego rozdziału. Numer iteracji podczas której następuje tranzycja pomiędzy kolejnymi punktami w przestrzeni 3D w celu syntezy trajektorii oznaczmy jako t . Założmy arbitralnie wartość średniego czasu uszkodzenia kołyski w pompie wielotłoczkowej jako T . Wartość pewności klasyfikacji w iteracji t wyznaczania trajektorii oznaczmy jako $c(t)$. c pochodzi od angielskiego "confidence", a t od angielskiego "transitions". Zmianę wartości funkcji $c(t)$ można skojarzyć ze zmianą wartości RUL. W miarę zwiększania się wartości funkcji $c(t)$ wartość RUL będzie się zmniejszać. Charakter zmniejszania się RUL opisać można następująco. Osiągnięcie maksymalnej wartości pewności klasyfikacji uszkodzenia następuje po czasie T od momentu rozpoczęcia tranzycji punktu pracy na płaszczyźnie 2D. Procentowa wartość maksymalnej wartości funkcji $c(t)$ oznaczona



Rysunek 3.25: Przebieg przykładowej trajektorii punktu pracy na płaszczyźnie 2D. Od punktu związanego z pompą sprawną do punktu związanego z pompą z uszkodzoną kołyską. Trajektorię oznaczono kolorem czerwonym.



Rysunek 3.26: Zależność zmiany wartości RUL w miarę zmiany położenia punktu pracy na płaszczyźnie 2D poprzez sukcesywne tranzycje. Założono T równe 5 lat.

jako p w miarę kolejnych tranzycji powodować będzie procentową zmianę wartości RUL tak jak opisano to w równaniu (3.2) oraz (3.3). W równaniu (3.2) jako N oznaczono maksymalną liczbę przeprowadzonych tranzycji pomiędzy punktami. Dla wyznaczonej trajektorii zmiana wartości RUL w konfrontacji ze zmianą wartości $c(t)$ wyglądać będzie tak, jak przedstawiono to na rysunku 3.26

$$p = \frac{\sum_{n=0}^t c(n)}{\sum_{n=0}^N c(n)} \quad (3.2)$$

$$RUL = (100\% - p)T \quad (3.3)$$

Ocenę jakości wyznaczonej wartości RUL przy założeniu posiadania rzeczywistych punktów pracy na płaszczyźnie 2D można kwantyfikować z wykorzystaniem różnego rodzaju metryk. Jedną z nich może być wartość MSE pomiędzy punktami w trzech

wymiarach. W przypadku danych rzeczywistych owe trzy wymiary stanowiłyby cechy na płaszczyźnie 2D syntetycznie wypracowane przez algorytm, oraz ekspercka ocena procentowego stopnia degradacji monitorowanego urządzenia przez osobę posiadającą odpowiednią wiedzę domenową podobnie jak wskazano to w [71]. Wartość MSE można byłoby wtedy zdefiniować jako odległość naniesionych punktów od prostej w przestrzeni 3D. W przypadku braku oceny eksperckiej, analiza odbywałaby się w dwóch wymiarach.

Przedstawiony algorytm umożliwia szacowanie wartości RUL na podstawie znajomości trajektorii punktu pracy oraz wartościach pewności klasyfikacji definiujących przestrzeń 3D. Oparty jest na wielu założeniach, a działanie przedstawiono w ramach jednego rodzaju uszkodzenia. Założenia te to:

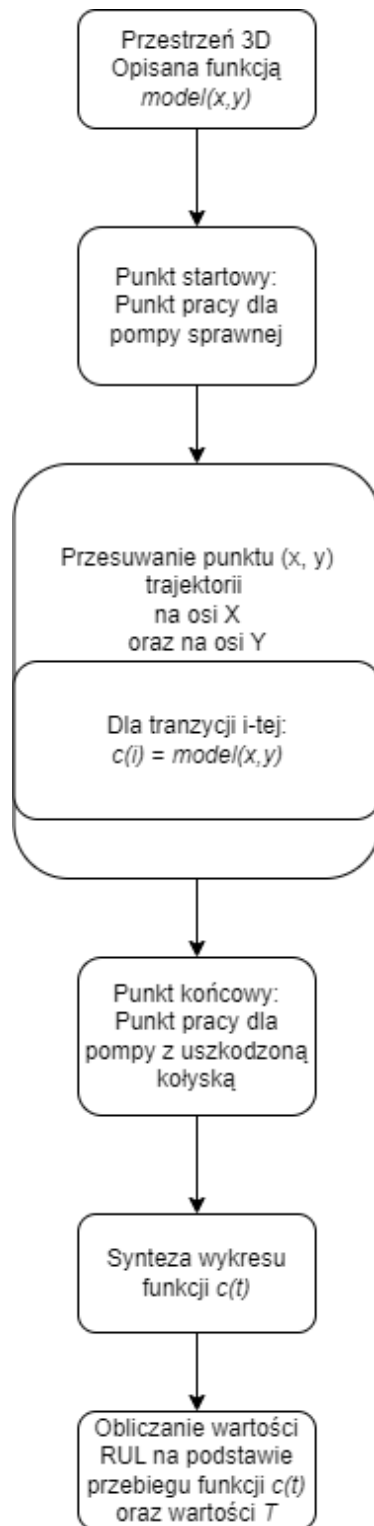
- Znajomość reżimu pracy pompy
- Arbitralnie dobrana liczba syntetycznych punktów pośrednich na płaszczyźnie 2D
- Przesuwanie się od punktu pracy związanego z pompą sprawną do punktu oznaczającego uszkodzenie idzie w parze z coraz większą wartością pewności predykcji uszkodzenia
- Znajomość średniego bezawaryjnego czasu użytkowania urządzenia do momentu wystąpienia konkretnej awarii

Tym niemniej algorytm stanowi punkt wyjścia do dalszych badań. Zwłaszcza jeżeli znane będą rzeczywiste pośrednie punkty pracy i szacowanie kształtu trajektorii w przestrzeni 3D będzie bardziej wiarygodne. Schematycznie algorytm przedstawiono na rysunku 3.27.

Wskaźniki jakości

Z biznesowego punktu widzenia najbardziej miarodajne byłoby odniesienie rzeczywistej wartości wskaźnika RUL do tej pochodzącej z wypracowanego algorytmu. Na podstawie zgromadzonych danych nie było takiej możliwości. Dokonano natomiast generacji sztucznych stanów pośrednich poprzez dopasowanie przestrzeni 3D zawierającej cechy z płaszczyzny 2D oraz wartość pewności klasyfikacji dla danej klasy pochodzących z sieci CNN. To właśnie dopasowanie poddane było ocenie jakości z wykorzystaniem metryki MSE oraz R^2 . Jakość dopasowania z wykorzystaniem algorytmu MLP była

dobra. Poczyniono założenie, aby tak dopasowana przestrzeń 3D stanowiła podstawę dalszych założeń i poczynionych analiz. Jedyne dostępne dane pochodziły z pompy albo w pełni sprawnej, albo uszkodzonej na tyle, że nie była zdolna do dalszej pracy poza środowiskiem laboratoryjnym. Należało zatem w jakiś sposób oszacować jak przebiega tranzycja pomiędzy punktem pracy związanym ze sprawną i uszkodzoną pompą. Miarę przeprowadzonego w taki sposób porównania również mogłoby stanowić wyznaczenie metryki MSE oraz R^2 . W miarę dostarczania kolejnych punktów pracy z degradującego się urządzenia porównanie wartości metryk stanowiłoby istotną informację z punktu widzenia oceny jakości algorytmu predykcji wartości RUL.



Rysunek 3.27: Proces syntezy trajektorii przemieszczania punktu pracy oraz szacowania wartości RUL z wykorzystaniem sztucznie wygenerowanych punktów pracy.

Rozdział 4

Zakończenie

Całość doświadczeń związanych z praktyką przemysłową, a zgromadzonych w trakcie tworzenia niniejszej pracy udowadnia, że efektywne wdrożenie dające biznesową wartość dodaną nowych zaproponowanych algorytmów jest trudne do przeprowadzenia. Znaczną trudność stanowi duża bezawaryjność monitorowanych w ramach pracy doktorskiej urządzeń. Przez to brak jest możliwości obserwowania i kwantyfikowania postępującej degradacji maszyny przemysłowej. W ramach niniejszej pracy w rozdziałach 3.1 oraz 2 opisano przeprowadzone wdrożenia oraz architekturę oprogramowania będącą nierozłączną częścią każdego wdrożenia. Wypracowane oprogramowanie umożliwiło przeprowadzenie wdrożeń, aczkolwiek z uwagi na bezawaryjność monitorowanych urządzeń nie było możliwości efektywnej ewaluacji zaproponowanego podejścia w warunkach produkcyjnych. Przeprowadzono natomiast badania w ośrodku B+R, dzięki którym pozyskano dane pochodzące z pompy sprawnej oraz posiadającej szereg wyszczególnionych defektów. Tak pozyskane dane stanowiły podstawę prowadzonych w ramach pracy rozważań natury badawczej.

Z biznesowego punktu widzenia, w pracy położono nacisk na dwa aspekty:

- Przejrzystość hipotez wypracowanych przez algorytmy z punktu widzenia klienta
- Wysoką jakość działania algorytmów z wykorzystaniem minimalnego kosztu koniecznej do montażu aparatury pomiarowej oraz z uwzględnieniem możliwości montażu tejże aparatury

4.1 Przejrzystość wypracowanych hipotez

Modele wypracowane w rozdziale 3.2.2 mają zdolność do wskazywania najistotniejszych mierzonych zmiennych z punktu widzenia klasyfikacji uszkodzeń. Możliwość wskazania tych zmiennych w połączeniu z wiedzą domenową klienta umożliwia prowadzenie dyskusji prowadzących do poprawy jakości działania modeli. Wiedza ekspercka może okazać się przydatna podczas konstruowania cech, z których korzystają trenowane modele. Finalnie na podstawie zebranych podczas pracy doświadczeń można wywnioskować, że możliwość wskazania klientowi jakie zmienne mają największy wpływ na klasyfikację uszkodzenia czyni system predykcji awarii bardziej przystępnym dla klienta.

Klasyfikator CNN opisany w rozdziale 3.2.2 z racji swojej specyfiki posiada zdolność do ekstrakcji cech użytecznych z punktu widzenia klasyfikacji. Nie są to cechy zrozumiałe i łatwe do analizy dla człowieka. Podjęto natomiast próbę wizualizacji tychże cech w taki sposób, aby osoba analizująca wizualizację w sposób intuicyjny była w stanie wskazać jaki zestaw wskaźników związany jest z pompą uszkodzoną, a jaki z pompą sprawną. Wizualizację uzyskano poprzez redukcję wymiarowości wektora cech pochodzących z sieci CNN. Dokonano rzutowania na płaszczyznę 2D. Dzięki porównaniu stosownych wskaźników jakości w podrozdziale 3.2.2 dzięki algorytmowi KNN wytypowano taką prezentację wyników na płaszczyźnie 2D, aby była ona najbardziej przejrzysta dla odbiorcy.

4.2 Jakość działania algorytmów a koszt aparatury pomiarowej

W rozdziale 3.2.2 podjęto próbę klasyfikacji uszkodzenia pompy wielotłoczkowej z rozróżnieniem na dostępność danych pomiarowych. Wytrenowano takie modele klasyfikacji dzięki którym możliwe było wskazanie które zmienne procesowe mają największe znaczenie w kontekście predykcji awarii. Wzięto pod uwagę estymaty wibracji oraz dane procesowe. Najbardziej pożądanym modelem był taki, który korzysta jedynie z wartości pochodzących z jednego czujnika wibracji. Z tego względu, że jego montaż jest najłatwiejszy i niemal zawsze możliwy z uwagi na infrastrukturę w ramach której funkcjonuje. Co więcej, niezbędna do pomiaru wibracji aparatura pomiarowa i czujniki są relatywnie niedrogie w porównaniu z kosztami montażu aparatury oraz pomiaru

ciśnienia i temperatury tłoczonego przez pompę medium. Modele wykorzystujące dane wibracyjne z jednego czujnika cechowały się jakością klasyfikacji, którą można było zdecydowanie poprawić. Podjęto się analizy widma drgań pochodzącego z jednego czujnika wibracji, mając na celu ograniczenie kosztów. Jakość klasyfikacji modelu CNN analizującego widma była znacznie wyższa aniżeli ta biorąca pod uwagę jedynie estymaty wibracji. Płynąca stąd konkluzja wskazuje, że jeden odpowiednio zamontowany czujnik wibracji jest wystarczający aby dzięki uzyskanym z niego pomiarów dokonywać bardzo dobrej jakości klasyfikacji uszkodzeń, co wyjaśniono szerzej w podrozdziale 3.2.2.

4.3 Przenaszalność algorytmów i dalsze badania

Niniejsza praca w zakresie wypracowanych podejść do klasyfikacji i predykcji uszkodzeń opiera się na pomiarach pochodzących z jednej pompy wielotłoczkowej, w której sztucznie dokonywano poszczególnych analizowanych uszkodzeń. Wypracowane podejście można zastosować do innych urządzeń, ale wymagane jest dotrenowanie algorytmów, to znaczy:

- Sieci neuronowych typu CNN służących do klasyfikacji i ekstrakcji cech
- Algorytmów redukcji wymiarowości, dzięki którym możliwe jest rzutowanie punktów pracy na płaszczyznę 2D
- Sieci typu MLP, które mają za zadanie predykcję wartości pewności klasyfikacji poszczególnych punktów pracy, tworząc tym samym przestrzeń 3D
- Odnalezienie stosownej trajektorii tranzycji pomiędzy stanem wskazującym na urządzenie sprawne, a stanem wskazującym degradację urządzenia

Dotrenowanie sieci neuronowych należy rozumieć jako transfer wiedzy opisany w podrozdziale 3.2.2. Na podstawie znanych z literatury udanych przypadków transferu wiedzy w sieciach neuronowych pomiędzy różnymi domenami [76] można przypuszczać, że dotrenowanie sieci służących do klasyfikacji uszkodzeń pompy wielotłoczkowej będzie pomocne w celu uzyskania dobrej jakości klasyfikacji uszkodzeń na przykład turbiny albo silnika [89]. Wypracowane algorytmy wymagają weryfikacji poprzez ich ewaluację na danych rzeczywistych pochodzących z degradujących się urządzeń o różnym stopniu degradacji. Im więcej jest punktów pracy tym lepsza będzie estymacja trajektorii

przejścia pomiędzy stanami, która służy szacowaniu czasu pozostałego użytkowania monitorowanej maszyny.

4.4 Podsumowanie

Całokształt analiz prowadzonych w ramach niniejszej pracy miał na celu udowodnienie tezy, która brzmi:

”Widmo prędkości drgań pozyskane z odpowiednio umiejscowionego czujnika drgań jest wystarczające, aby poprawnie klasyfikować i w przejrzysty sposób prezentować stan pompy z rozróżnieniem kilku rodzajów uszkodzeń i wartości obciążenia”.

Przeprowadzone badania dotyczące właściwego umiejscowienia czujnika drgań opisano w podrozdziale 3.2.2. Rozpoczęto analizę od klasyfikacji binarnej (pompa sprawna / uszkodzona). Wytrenowano po jednym modelu typu drzewo decyzyjne oraz regresja logistyczna dla każdego rodzaju obciążania stałego. Po uzyskaniu satysfakcjonujących wyników, co przedstawiają między innymi tabele 3.4 oraz 3.7, przystąpiono do klasyfikacji wieloklasowej jako bardziej użytecznej z wdrożeniowego i naukowego punktu widzenia. Rozważania prowadzone w podrozdziale 3.2.2 doprowadziły do wniosku, że czujnik zamontowany promieniowo na pompie wielotłoczkowej jest najbardziej użyteczny na potrzeby klasyfikacji uszkodzeń. Umotywowane zostało to przeprowadzoną analizą istotności mierzonych zmiennych. Wskazała ona, że estymaty drgań pochodzące z tego właśnie czujnika są najistotniejsze. W przypadku drzew decyzyjnych zmienne związane z czujnikiem zamontowanym promieniowo znajdowały się na najwyższym poziomie drzewa decyzyjnego. Dla modeli uwzględniających dane z obu czujników ukazuje to tabela 3.9, zaś dla modeli wytrenowanych z wykorzystaniem danych z obu czujników drgań oraz zmiennych procesowych tabela 3.4. Z kolei modele regresji logistycznej dla modeli uwzględniających dane z obu czujników drgań w większości jako największy co do wartości bezwzględnej współczynnik w równaniu regresji wskazywały zmienne pochodzące z czujnika zamontowanego promieniowo. Powyższe stwierdzenie ma swoje odzwierciedlenie w tabeli 3.12. Natomiast modele uwzględniające dane z czujników drgań i zmienne procesowe w kilku przypadkach wskazały zmienne pochodzące z czujnika zamontowanego promieniowo jako istotne. Ilustruje to tabela 3.13. Decydujący w doborze najbardziej miarodajnego umiejscowienia czujnika drgań był fakt, że po reasemblingu pompy wartości estymat obliczonych na podstawie jego wskazań miały podobną wartość. Podczas gdy wartości estymat pochodzące z czujnika zamontowanego

osiowo cechowały się po reasemblingu pompy dużym rozrzutem.

Po wyznaczeniu najbardziej miarodajnego umiejscowienia czujnika drgań z punktu widzenia klasyfikacji uszkodzeń przystąpiono do analizy wykorzystującej widmo drgań. Wpierw przedmiotem badań były syntetyczne spektrogramy. W podrozdziale 3.2.2 dokonano ich generacji zgodnie z przyjętym schematem ukazany na rysunku 3.13. Uzyskano kompromis pomiędzy rozmiarem spektrogramu a jego użytecznością. Zdecydowano, aby każdy ze spektrogramów związany był z 64 binami częstotliwości oraz z 64 sekundami. Proces prowadzący do uzyskania wspomnianego kompromisu został szerzej opisany w podrozdziale 3.2.2. Syntetyczne spektrogramy posłużyły do wytrenowania konwolucyjnej sieci neuronowej (CNN) mającej za zadanie ich klasyfikację. Sieć CNN miała predefiniowaną architekturę, co zostało umotywowane w podrozdziale 3.2.2. Spektrogramy na potrzeby podania ich na wejście sieci neuronowej zostały koncepcyjnie zinterpretowane jako obrazy monochromatyczne. Ewaluacja sieci wytrenowanych na sztucznie wygenerowanych spektrogramach ukazała, że dla kilku zakresów częstotliwości klasyfikacja uszkodzeń była idealna. Wyniki tak przeprowadzonej klasyfikacji prezentuje tabela 3.15. Całość działań podjętych na potrzeby treningu i ewaluacji sieci CNN nauczonych na syntetycznych spektrogramach związana była z przeprowadzonym na kolejnym etapie transferem wiedzy.

Sieci nauczone na syntetycznych spektrogramach wykorzystano na potrzeby dotrenowania tych sieci na spektrogramach rzeczywistych. Dotrenowanie sieci przebiegło dwojako. Jedno podejście zakładało trening jedynie części klasyfikującej sieci CNN. W drugim podejściu douczona została cała sieć CNN. Lepsze wyniki uzyskano z wykorzystaniem pierwszego podejścia, w którym to dla dwóch zakresów częstotliwości klasyfikacja była idealna. Tym niemniej, sieci dotrenowane w ramach drugiego podejścia również cechują się dobrą jakością klasyfikacji. Rezultaty ewaluacji z wykorzystaniem stosownej metryki dla dotrenowanych sieci ukazują tabele 3.17, 3.18 oraz 3.19. Proces transferu wiedzy opisano szerzej w podrozdziale 3.2.2, zaś bardziej obszerną dyskusję dotyczącą jakości klasyfikacji przeprowadzono w podrozdziale 3.2.2. Rzeczywiste spektrogramy utworzono wykorzystując sekwencyjnie połączone widma drgań pozyskane z eksperymentów pomiarowych, a pochodzące z czujnika drgań zamontowanego promieniowo. Sposób syntezy spektrogramów z wykorzystaniem widm pozyskanych z urządzenia EXT/FFT przeznaczonego do pomiaru wibracji, został szerzej opisany w podrozdziale 3.2.2,

Dotrenowane w całości sieci CNN o dobrej jakości klasyfikacji były punktem wyj-

ścia do stworzenia przejrzystego sposobu prezentacji stanu pompy. Konwolucyjna sieć neuronowa oprócz zdolności klasyfikacyjnych posiada zdolność do ekstrakcji cech. Wykorzystano tę zdolność w następujący sposób. Dla każdego ze spektrogramów wejściowych do sieci dokonywano klasyfikacji, natomiast odbywało się to w celu ekstrakcji 9216 wartości pochodzących z warstwy flatten wskazanej na rysunku 3.14. Wartości te były niejako wektorem opisującym spektrogram. Następnie tak uzyskane wektory dla każdego ze spektrogramów poddane zostały redukcji wymiarowości do dwóch wymiarów z wykorzystaniem algorytmu PCA oraz sieci neuronowych typu autoenkoder. Proces ten schematycznie przedstawia rysunek 3.15. Wyniki działania algorytmów redukcji wymiarowości zostały poddane ocenie pod kątem przejrzystości prezentacji cech na płaszczyźnie 2D. Ocena została przeprowadzona wykorzystując algorytm KNN, który następnie podlegał ewaluacji i analizie uzyskanych metryk, co opisano szerzej w podrozdziale 3.2.2. Tabela 3.21 ukazuje, że proces skutkujący najbardziej przejrzystym sposobem prezentacji cech na płaszczyźnie 2D z rozróżnieniem kilku rodzajów uszkodzeń i wartości obciążenia obejmował normalizację danych pozyskanych z warstwy flatten dla sieci CNN wytrenowanej na spektrogramach z zakresu częstotliwości 0-307Hz. Tenże zakres częstotliwości jest tożsamy z zakresem, dla którego uzyskano najlepszą jakość klasyfikacji sieci CNN dotrenowanej w całości. Rysunek 3.16 ilustruje najbardziej przejrzystą płaszczyznę 2D z punktu widzenia obranej metryki, zaś rysunek 3.16 przedstawia tę samą płaszczyznę 2D jednak dla jednej określonej wartości obciążenia.

Przytoczone w niniejszej pracy argumenty dowodzą prawdziwości postawionej tezy.

Bibliografia

- [1] Datasheet komputera brzegowego avtse-rpu-iiotg. <https://www.farnell.com/datasheets/2828937.pdf>.
- [2] Datasheet komputera brzegowego moxa uc8100. <https://www.moxa.com/getmedia/cfbe743e-ce9a-4699-a14a-b40081554c50/moxa-uc-8100-series-datasheet-v1.8.pdf>.
- [3] Dokumentacja bazy danych influxdb. <https://www.influxdata.com/products/influxdb-overview/>.
- [4] Dokumentacja czujnika hansford hs-150. <https://www.hansfordsensors.com/wp-content/uploads/2022/01/HS-150-2-Pin-MS-TS217.6.pdf>.
- [5] Dokumentacja czujników ciśnienia firmy ifm. <https://www.ifm.com/pl/pl/category/200/020/010>.
- [6] Dokumentacja czujników temperatury firmy ifm. <https://www.ifm.com/pl/pl/category/200/020/040>.
- [7] Dokumentacja pompy hsp-10v firmy hydraut. <http://pompyhydrauliczne.net/pl/p/file/fd560d91fca0e6a9dd0be0566a33464d/HSP-10V/REV11-web.pdf>.
- [8] Funkcja generująca wykresy konturowe. <https://matplotlib.org/stable/api/matplotlib.axes.Axes.tricontourf.html>.
- [9] Informacje o narzędziu 7z. <https://www.7-zip.org/>.
- [10] Informacje o narzędziu netcat. <https://en.wikipedia.org/wiki/Netcat>.
- [11] Informacje o narzędziu nssm. <https://nssm.cc/download>.

- [12] Informacje o usługach systemd. <https://en.wikipedia.org/wiki/Systemd>.
- [13] Internetowy system SCADA MOLOS.CLOUD. <https://molos.cloud>.
- [14] Lista najpopularniejszych narzędzi do tworzenia systemów SCADA. <https://automationforum.co/leading-12-scada-software/>.
- [15] Oficjalna strona internetowa Azure Container Registry. <https://azure.microsoft.com/en-us/services/container-registry/>.
- [16] Oficjalna strona internetowa Docker Hub. <https://hub.docker.com/>.
- [17] Oficjalna strona projektu PyInstaller. <https://pyinstaller.org/en/stable/>.
- [18] Opis chmury Microsoft Azure. <https://azure.microsoft.com/pl-pl/>.
- [19] Opis frameworku Microsoft Azure IOTEDGE. <https://azure.microsoft.com/en-us/services/iot-edge/iotedge-overview>.
- [20] Opis mechanizmu CI/CD. <https://docs.microsoft.com/en-us/azure/architecture/example-scenario/apps/devops-dotnet-webapp>.
- [21] Opis zasadności wykorzystania influxdb jako bazy danych przechowującej szeregi czasowe. <https://www.influxdata.com/time-series-database/>.
- [22] Strona biblioteki asyncio służącej do implementacji asynchroniczności w języku python. <https://pypi.org/project/asyncio/>.
- [23] Strona główna repozytorium PyPI. <https://pypi.org>.
- [24] Zestawienie czasu dostaw części do pomp wirowych. <https://empoweringpumps.com/pump-lead-times-what-to-expect/>.
- [25] Azure functions description. <https://azure.microsoft.com/en-us/services/functions/>, 2022. Dostęp: 2022-06-22.
- [26] Biblioteki open-source służące do dwustronnej komunikacji z chmurą Azure. <https://github.com/Azure>, 2022. Dostęp: 2022-06-22.
- [27] Opis magazynu chmurowego Azure Storage. <https://learn.microsoft.com/en-us/azure/storage/common/storage-introduction>, 2022. Dostęp: 2022-06-22.

- [28] Porównanie trybów dostępu do danych z archiwum Azure Blob Storage. <https://learn.microsoft.com/en-us/azure/storage/blobs/access-tiers-overview>, 2022. Dostęp: 2022-06-22.
- [29] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [30] Farzaneh Ahmadzadeh, Jan Lundberg. Remaining useful life estimation. *International Journal of System Assurance Engineering and Management*, 5(4):461–474, 2014.
- [31] Hosameldin Ahmed, Asoke Nandi. Intrinsic dimension estimation-based feature selection and multinomial logistic regression for classification of bearing faults using compressively sampled vibration signals. *Entropy*, 24(4), 2022.
- [32] Ryan Ahmed, Mohammed El Sayed, Andrew Gadsden, Jimi Tjong, Saeid Habibi. Automotive internal-combustion-engine fault detection and classification using artificial neural network techniques. *IEEE Transactions on Vehicular Technology*, 64(1):21–33, 2015.
- [33] Mohammad Ali, Nasmus Shabbir, Xiaodong Liang, Yu Zhang, Ting Hu. Machine learning-based fault diagnosis for single- and multi-faults in induction motors using measured stator currents and vibration signals. *IEEE Transactions on Industry Applications*, 55(3):2378–2391, 2019.
- [34] Joshuva Arockia, Shridhar, Nandakumar Selvaraju, Jenoris Muthiya, Mohankumar Subramanian. A machine learning approach for vibration signal based fault classification on hydraulic braking system through c4.5 decision tree classifier and logistic model tree classifier. strony 1–6, 09 2020.

- [35] David Arthur, Sergei Vassilvitskii. How slow is the k-means method? *Proceedings of the Twenty-Second Annual Symposium on Computational Geometry*, SCG '06, strona 144–153, New York, NY, USA, 2006. Association for Computing Machinery.
- [36] Petr Baron, Marek Kočiško, Simona Hlavatá, Eduard Franas. Vibrodiagnostics as a predictive maintenance tool in the operation of turbo generators of a small hydropower plant. *Advances in Mechanical Engineering*, 14(5), 2022.
- [37] Christopher Bishop. *Pattern Recognition and Machine Learning*, strona 206. Information Science and Statistics. Springer, 2006.
- [38] Andrew Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.
- [39] Leo Breiman, Jerome. Friedman, Charles Stone, Richard Olshen. *Classification and Regression Trees*. Taylor & Francis, 1984.
- [40] Kay Brodersen, Cheng Ong, Klaas Stephan, Joachim Buhmann. The balanced accuracy and its posterior distribution. *2010 20th International Conference on Pattern Recognition*, strony 3121–3124, 2010.
- [41] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jacques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, strony 108–122, 2013.
- [42] Robin Bäbler, Tim Bäbler, Markus Kley. Classification of load and rotational speed at wire-race bearings using convolutional neural networks with vibration spectrograms. *tm - Technisches Messen*, 89(5):352–362, 2022.
- [43] Yixin Cai, Mo-Yuen Chow, Wenbin Lu, Lexin Li. Evaluation of distribution fault diagnosis algorithms using roc curves. *IEEE PES General Meeting*, strony 1–6, 2010.
- [44] Peter Caie, Neofytos Dimitriou, Ognjen Arandjelović. Chapter 8 - precision medicine in digital pathology via image analysis and machine learning. Stanley

- Cohen, redaktor, *Artificial Intelligence and Deep Learning in Pathology*, strony 149–173. Elsevier, 2021.
- [45] Emanuele Cardillo, Alina Caddemi. Feasibility study to preserve the health of an industry 4.0 worker: a radar system for monitoring the sitting-time. *2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0 IoT)*, strony 254–258, 2019.
- [46] Paolo Casoli, Mirko Pastori, Fabio Scolari, Massimo Rundo. A vibration signal-based method for fault identification and classification in hydraulic axial piston pumps. *Energies*, 12(5), 2019.
- [47] Simon Chan, Philip Treleaven. Chapter 5 - continuous model selection for large-scale recommender systems. Venu Govindaraju, Vijay Raghavan, Calyampudi Rao, redaktorzy, *Big Data Analytics*, wolumen 33 serii *Handbook of Statistics*, strony 107–124. Elsevier, 2015.
- [48] Qun Chao, Xiaoliang Wei, Junbo Lei, Jianfeng Tao, Chengliang Liu. Improving accuracy of cavitation severity recognition in axial piston pumps by denoising time–frequency images. *Measurement Science and Technology*, 33(5):055116, 2022.
- [49] Chien-Hua Chen, Po-Hsiang Lin, Jer-Guang Hsieh, Shu-Ling Cheng, Jyh-Horng Jeng. Robust multi-class classification using linearly scored categorical cross-entropy. *2020 3rd IEEE International Conference on Knowledge Innovation and Invention (ICKII)*, strony 200–203, 2020.
- [50] Davide Chicco, Matthijs Warrens, Giuseppe Jurman. The coefficient of determination r-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Comput. Sci.*, 7(e623).
- [51] Maximilian Christ, Nils Braun, Julius Neuffer, Andreas Kempa-Liehr. Time series feature extraction on basis of scalable hypothesis tests (tsfresh – a python package). *Neurocomputing*, 307:72–77, 2018.
- [52] Harriet Dawson, Olivier Dubrule, Cédric John. Impact of dataset size and convolutional neural network architecture on transfer learning for carbonate rock classification. *Computers Geosciences*, 171:105284, 2023.
- [53] Bill Drury. Institution of Engineering and Technology (The IET), 2009.

- [54] Shimon Even, Guy Even. *Graph Algorithms*. Cambridge University Press, 2011.
- [55] Hongwei Fan, Ceyi Xue, Xuhui Zhang, Xiangang Cao, Shuoqi Gao, Sijie Shao. Vibration images-driven fault diagnosis based on cnn and transfer learning of rolling bearing under strong noise. *Shock and Vibration*, 2021, 2021.
- [56] Paweł Fic. Digital transformation of a batch manufacturing plant, 2022. Zaprezentowano na konferencji POB5 na Politechnice Śląskiej, <https://www.polsl.pl/pob5/en/ps-aktualnosci/priority-research-area-5-pra5-automation-and-industry-4-0-thematic-conference/>.
- [57] Paweł Fic. Możliwości chmury azure wdrażanie modeli uczenia maszynowego na urządzenia brzegowe, 2022. Zaprezentowano na konferencji NWDPN XI, <https://doktorant.com.pl/index.php/2-uncategorised/169-konferencja-nwdpn-xi-edycja-warszawa/>.
- [58] Paweł Fic. Przewidywanie awarii urządzeń przemysłowych w oparciu o analizę wibrodiagnostyczną z wykorzystaniem sztucznej inteligencji, 2022. Zaprezentowano na konferencji NWDPN XI, <https://doktorant.com.pl/index.php/2-uncategorised/169-konferencja-nwdpn-xi-edycja-warszawa/>.
- [59] Paweł Fic. Zestawienie wskaźników wibrodiagnostycznych urządzeń przemysłowych na potrzeby implementacji systemów predictive maintenance, Zagadnienia aktualnie poruszane przez młodych naukowców 22, 2022. strony 126-131, ISBN: 978-83-66772-28-1.
- [60] Paweł Fic, Adam Czornik, Piotr Rosikowski. Anomaly detection for hydraulic power units - a case study. *Future Internet*, 15(6), 2023.
- [61] Caixia Gao, Ke Lv, Tong Wu, Jikai Si, Yihua Hu. Method for determining starting point of rolling bearing life prediction based on linear regression. *Electronics*, 8(9), 2019.
- [62] Martin Gardner, Stephen Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric Environment*, 32(14):2627–2636, 1998.
- [63] Bernhard Geiger, Gernot Kubin. Relative information loss in the PCA. *2012 IEEE Information Theory Workshop*. IEEE, 2012.

- [64] Aurelien Geron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 3rd Edition*. O'Reilly Media, 2022. <https://github.com/ageron/handson-ml3>.
- [65] Marta Gherardini, Evangelos Mazomenos, Arianna Menciassi, Danail Stoyanov. Catheter segmentation in x-ray fluoroscopy using synthetic data and transfer learning with light u-nets. *Computer Methods and Programs in Biomedicine*, 192:105420, 2020.
- [66] Tamara Golubović Sedmak, Emil Veg. Failure prevention of rotating equipment by vibrodiagnostics. *Structural Integrity and Life*, 12:99–104, 2012.
- [67] Ian Goodfellow, Yoshua Bengio, Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [68] Alexander Gouskov, Sergey Voronov, Ilya Ivanov, Igor Kiselev, Vadim Novikov. Investigation of vibratory drilling model with adaptive control. part 2: Mixed control of peak-to-peak vibration displacement and cutting continuity index. *Journal of Vibroengineering*, 17:4301–4312, 2015.
- [69] Charles Harris, Jarrod Millman, Stéfan van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, Travis Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020.
- [70] Jeff Heaton. An empirical analysis of feature engineering for predictive modeling. *SoutheastCon 2016*, strongy 1–6, 2016.
- [71] Patricia Henriquez, Jesus Alonso, Miguel Ferrer, Carlos Travieso. Review of automatic fault diagnosis systems using audio and vibration signals. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(5):642–652, 2014.
- [72] Richard Hoptroff. The principles and practice of time series forecasting and business modelling using neural nets. *Neural Computing & Applications*, 1993.
- [73] Chiu-Han Hsiao, Frank Yeong-Sung Lin, Hao-Jyun Yang, Yennun Huang, Yu-Fang Chen, Ching-Wen Tu, Si-Yao Zhang. Optimization-based approaches for

- minimizing deployment costs for wireless sensor networks with bounded estimation errors. *Sensors*, 21(21), 2021.
- [74] Qiao Hu, Yangkun Zhang, Xingju Xie, Wenbin Su, Yangyang Li, Liuhaohao Shan, Xiaojie Yu. Optimal placement of vibration sensors for industrial robots based on bayesian theory. *Applied Sciences*, 12(12), 2022.
- [75] John Hunter. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.
- [76] Mahbub Hussain, Jordan Bird, Diego Faria. A study on cnn transfer learning for image classification. Ahmad Lotfi, Hamid Bouchachia, Alexander Gegov, Caroline Langensiepen, Martin McGinnity, redaktorzy, *Advances in Computational Intelligence Systems*, strony 191–202, Cham, 2019. Springer International Publishing.
- [77] Sergey Ioffe, Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.
- [78] Abdul Jerri. The shannon sampling theorem—its various extensions and applications: A tutorial review. *Proceedings of the IEEE*, 65(11):1565–1596, 1977.
- [79] Pilsung Kang, Sungmin Lim. A taste of scientific computing on the gpu-accelerated edge device. *IEEE Access*, 8:208337–208347, 2020.
- [80] Ziqiu Kang, Cagatay Catal, Bedir Tekinerdogan. Remaining useful life (RUL) prediction of equipment in production lines using artificial neural networks. *Sensors (Basel)*, 21(3):932.
- [81] Mike Keen, Phil Klahr. Successful applications throughout the manufacturing lifecycle. *Annual Technical Conference of the British Computer Society Specialist Group on Expert Systems*, strona 267. Cambridge University Press, 1991.
- [82] Diederik Kingma, Jimmy Ba. Adam: A method for stochastic optimization. Yoshua Bengio, Yann LeCun, redaktorzy, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
- [83] Czesław Kowalski. Oficyna Wydaw. Politechniki Wrocławskiej, 2013.

- [84] Alex Krizhevsky. Learning multiple layers of features from tiny images. Raport instytutowy, 2009.
- [85] Lubo Kucera, Tom Gajdosik. The vibrodiagnostics of gears. *54th International Conference of Machine-Design-Departments, Hejnice, Czech Republic. In: The Vibrodiagnostics of Gears*, strony 113–118, 2014.
- [86] Mitchell Lebold, Katherine McClintic, Robert Campbell, Carl Byington, Kenneth Maynard. Review of vibration analysis methods for gearbox diagnostics and prognostics. *Proceedings of the 54th meeting of the society for machinery failure prevention technology*, wolumen 634, strona 16. Virginia Beach, VA, 2000.
- [87] Chester Lee. An algorithm for path connections and its applications. *IRE Transactions on Electronic Computers*, EC-10(3):346–365, 1961.
- [88] Ao Li, Zhiqiang Wu, Huaiyin Lu, Deyun Chen, Guanglu Sun. Collaborative self-regression method with nonlinear feature based on multi-task learning for image classification. *IEEE Access*, 6:43513–43525, 2018.
- [89] Chuan Li, Shaohui Zhang, Yi Qin, Edgar Estupinan. A systematic review of deep transfer learning for machinery fault diagnosis. *Neurocomputing*, 407:121–135, 2020.
- [90] Wei Li, Shuai Ding, Yi Chen, Hao Wang, Shanlin Yang. Transfer learning-based default prediction model for consumer credit in China. *The Journal of Supercomputing*, 2019.
- [91] Wei-Chao Lin, Shih-Wen Ke, Chih-Fong Tsai. Top 10 data mining techniques in business applications: a brief survey. *Kybernetes*, 2017.
- [92] Karol Listewnik, Grzegorz Grzeczka, Maciej Kłaczyński, Witold Cioch. An on-line diagnostics application for evaluation of machine vibration based on standard iso 10816-1. *Journal of Vibroengineering*, 17:4248–4258, 12 2015.
- [93] Xinbo Liu, Yice Zhao. Comparative study of amplitude spectrum estimation methods for vibration test under continuous sinusoidal excitation. *2020 7th International Conference on Information Science and Control Engineering (ICISCE)*, strony 1763–1767, 2020.

- [94] Yaqiong Lv, Wenqin Zhao, Zhiyao Zhao, Weidong Li, Kam Hung Ng. Vibration signal-based early fault prognosis: Status quo and applications. *Advanced Engineering Informatics*, 52:101609, 2022.
- [95] Chandrabhanu Malla, Isham Panigrahi. Review of condition monitoring of rolling element bearing using vibration analysis and other techniques. *Journal of Vibration Engineering & Technologies*, 7(4):407–414, 2019.
- [96] Sally McInerny, Yuxing Dai. Basic vibration signal processing for bearing fault detection. *IEEE Transactions on Education*, 46(1):149–156, 2003.
- [97] Martin Mihelich, Charles Dognin, Yan Shu, Michael Blot. A characterization of mean squared error for estimator with bagging, 2019.
- [98] Cassandra Milbradt, Martin Wahl. High-probability bounds for the reconstruction error of PCA. *Statistics & Probability Letters*, 161:108741, 2020.
- [99] Puneet Mishra, Dário Passos. Realizing transfer learning for updating deep learning models of spectral data to be used in new scenarios. *Chemometrics and Intelligent Laboratory Systems*, 212:104283, 2021.
- [100] Sajal Misra, Satish Kumar, Sameer Sayyad, Arunkumar Bongale, Priya Jadhav, Ketan Kotecha, Ajith Abraham, Lubna Abdelkareim Gabralla. Fault detection in induction motor using time domain and spectral imaging-based transfer learning approach on vibration data. *Sensors*, 22(21), 2022.
- [101] Omar Mohammed, Matti Rantatalo, Jan-Olov Aidanpää. Improving mesh stiffness calculation of cracked gears for the purpose of vibration-based fault analysis. *Engineering Failure Analysis*, 34:235–251, 2013.
- [102] Katarina Monkova, Peter Monka. Vibrodiagnostics and its application in manufacturing practice. *Applied Mechanics and Materials*, 390:220–224.
- [103] Marianne Mosher. Understanding vibration spectra of planetary gear systems for fault detection. wolumen 4 serii *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, strony 645–652, 2003.
- [104] Ahmed Moustafa, Ashraf El-Hamalawi, Andrew Wheatley. The impact of bad sensors on the water industry and possible alternatives. *Electronic Journal of Information Technology in Construction*, 13, 04 2008.

- [105] Caxton Okoh, Rajkumar Roy, Jörn Mehnen, Louis Redding. Overview of remaining useful life prediction techniques in through-life engineering services. *Procedia CIRP*, 16:158–163, 2014. Product Services Systems and Value Creation. Proceedings of the 6th CIRP Conference on Industrial Product-Service Systems.
- [106] Giulia Pascoletti, Alessandra Aldieri, Mara Terzini, Pinaki Bhattacharya, Michele Cali, Elisabetta Zanetti. Stochastic PCA-based bone models from inverse transform sampling: Proof of concept for mandibles and proximal femurs. *Applied Sciences*, 11(11), 2021.
- [107] Minh Tuan Pham, Jong-Myon Kim, Cheol Hong Kim. Accurate bearing fault diagnosis under variable shaft speed using convolutional neural networks and vibration spectrogram. *Applied Sciences*, 10(18), 2020.
- [108] Venkatramana Reddy, William Miller, Michael Mavrouniotis. Inverse-signal analysis with PCA. *Chemometrics and Intelligent Laboratory Systems*, 36(1):17–30, 1997.
- [109] Markus Reiß, Martin Wahl. Non-asymptotic upper bounds for the reconstruction error of PCA, 2016.
- [110] Sudarsan Sahoo, Kuldeep Kushwah, Arun Kumar Sunaniya. Health monitoring of wind turbine blades through vibration signal using advanced signal processing techniques. *2020 Advanced Communication Technologies and Signal Processing (ACTS)*, strony 1–6, 2020.
- [111] Bernhard Schölkopf, John Platt, John Shawe-Taylor, Alex Smola, Robert Williamson. Estimating the support of a high-dimensional distribution. *Neural Comput.*, 13(7):1443–1471.
- [112] Patrick Seeling, Martin Reisslein, Beshan Kulapala. Network performance evaluation using frame size and quality traces of single-layer and two-layer video: A tutorial. *IEEE Communications Surveys & Tutorials*, 6(3):58–78, 2004.
- [113] Siyu Shao, Wenjun Sun, Peng Wang, Robert Gao, Ruqiang Yan. Learning features from vibration signals for induction motor fault diagnosis. *2016 International Symposium on Flexible Automation (ISFA)*, strony 71–76, 2016.

- [114] Aditya Sharma, M Amarnath, PK Kankar. Feature extraction and fault severity classification in ball bearings. *Journal of Vibration and Control*, 22(1):176–192, 2016.
- [115] Junxian Shen, Di Song, Tianchi Ma, Feiyun Xu. Quantitative detection of blade crack damage based on vibro-acoustic information and multi- dimensional feature fusion cnn. *2022 Global Reliability and Prognostics and Health Management (PHM-Yantai)*, strony 1–6, 2022.
- [116] Geon-Ho Shin, Jang-Wook Hur. Correlation coefficient based optimal vibration sensor placement and number. *Sensors*, 22(3), 2022.
- [117] Xiao-Sheng Si, Wenbin Wang, Chang-Hua Hu, Dong-Hua Zhou. Remaining useful life estimation – a review on the statistical data driven approaches. *European Journal of Operational Research*, 213(1):1–14, 2011.
- [118] Bernard Silverman, Chris Jones. An important contribution to nonparametric discriminant analysis and density estimation: Commentary on fix and hodes (1951). *International Statistical Review / Revue Internationale de Statistique*, 57(3):233–238, 1989.
- [119] Leslie Smith. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay, 2018.
- [120] Anzhelika Stakhova. Application for vibration diagnostics. *Informatyka, Automatyka, Pomiar w Gospodarce i Ochronie Środowiska*, 12(3), 2022.
- [121] Christopher Stephen. A review about the diagnostics of rotodynamic pump using vibro-acoustic method. *Indian journal of science and technology*, 9, 2016.
- [122] Yanan Sun, Bing Xue, Mengjie Zhang, Gary Yen. Completely automated cnn architecture design based on blocks. *IEEE Transactions on Neural Networks and Learning Systems*, 31(4):1242–1254, 2020.
- [123] Kanae Takahashi, Kouji Yamamoto, Aya Kuchiba, Tatsuki Koyama. Confidence interval for micro-averaged f1 and macro-averaged f1 scores. *Applied Intelligence*, 2022.
- [124] Masayuki Tamura, Shinsuke Tsujita. A study on the number of principal components and sensitivity of fault detection using pca. *Computers & Chemical Engineering*, 31(9):1035–1046, 2007.

- [125] Shengnan Tang, Yong Zhu, Shouqi Yuan, Guangpeng Li. Intelligent diagnosis towards hydraulic axial piston pump using a novel integrated cnn model. *Sensors*, 20(24), 2020.
- [126] Jing Tian, Carlos Morillo, Michael Azarian, Michael Pecht. Motor bearing fault detection using spectral kurtosis-based feature extraction coupled with k-nearest neighbor distance analysis. *IEEE Transactions on Industrial Electronics*, 63(3):1793–1803, 2016.
- [127] Michael Tipping, Christopher Bishop. Mixtures of Probabilistic Principal Component Analyzers. *Neural Computation*, 11(2):443–482, 1999.
- [128] František Trebuňa, František Šimčák, Jozef Bocko, Róbert Huňady, Miroslav Pástor. Complex approach to the vibrodiagnostic analysis of excessive vibration of the exhaust fan. *Engineering Failure Analysis*, 37:86–95, 2014.
- [129] Cornelis van Rijsbergen. An algorithm for information structuring and retrieval. *Computer Journal*, 14:407–412, 1971.
- [130] Blesson Varghese, Nan Wang, Sakil Barbhuiya, Peter Kilpatrick, Dimitrios Nikolopoulos. Challenges and opportunities in edge computing. *2016 IEEE International Conference on Smart Cloud (SmartCloud)*, strony 20–26, 2016.
- [131] Jinlong Wang, Yi Su, Alagu Subramaniam, John Hock Lye Pang. Archard model guided feature engineering improved support vector regression for rail wear analysis. *Engineering Failure Analysis*, 137:106248, 2022.
- [132] Tianyi Wang, Jianbo Yu, David Siegel, Jay Lee. A similarity-based prognostics approach for remaining useful life estimation of engineered systems. *2008 International Conference on Prognostics and Health Management*, strony 1–6, 2008.
- [133] Yi Wang, Peter Tse, Baoping Tang, Yi Qin, Lei Deng, Tao Huang, Guanghua Xu. Order spectrogram visualization for rolling bearing fault detection under speed variation conditions. *Mechanical Systems and Signal Processing*, 122:580–596, 2019.
- [134] Yiting Wang, Shah Nazir, Muhammad Shafiq. An overview on analyzing deep learning and transfer learning approaches for health monitoring. *Computational and Mathematical Methods in Medicine*, 2021.

- [135] Michael Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021.
- [136] Maohua Xiao, Kai Wen, Cunyi Zhang, Xiao Zhao, Weihua Wei, Dan Wu. Research on fault feature extraction method of rolling bearing based on nmd and wavelet threshold denoising. *Shock and Vibration*, 2018:9495265, 2018.
- [137] Hongmei Yan, Yingtao Jiang, Jun Zheng, Chenglin Peng, Qinghui Li. A multilayer perceptron-based medical decision support system for heart disease diagnosis. *Expert Systems with Applications*, 30(2):272–281, 2006.
- [138] Yi Yang, Wenkai Xia, Jianming Han, Yunfei Song, Jiangfeng Wang, Yiping Dai. Vibration analysis for tooth crack detection in a spur gear system with clearance nonlinearity. *International Journal of Mechanical Sciences*, 157-158:648–661, 2019.
- [139] Isa Yesilyurt. The application of the conditional moments analysis to gearbox fault detection—a comparative study using the spectrogram and scalogram. *NDT & E International*, 37(4):309–320, 2004.
- [140] Li Yizhen, Ling Lin, Wu Jun. The application of pipeline technology: An overview. *2011 6th International Conference on Computer Science & Education (ICCSE)*, strony 47–51, 2011.
- [141] Abdelraouf Youcef, Noureddine Guersi, Nacer Saadi, Nadir Boutasseta. Rolling element bearing fault diagnosis for rotating machinery using vibration spectrum imaging and convolutional neural networks. *The International Journal of Advanced Manufacturing Technology*, 2020.
- [142] Wennian Yu, II Yong Kim, Chris Mechefske. An improved similarity-based prognostic algorithm for rul estimation using an rnn autoencoder scheme. *Reliability Engineering & System Safety*, 199:106926, 2020.
- [143] Hong Zhang, Jian Zhou, Danial Jahed Armaghani, Muhammad Tahir, Binh Thai Pham, Van Van Huynh. A combination of feature selection and random forest techniques to solve a problem related to blast-induced ground vibration. *Applied Sciences*, 10(3), 2020.

-
- [144] Kai Zhang, Xiaoqing Hu, Lei Kang, Qiaoyu Ma, Xin Zhang. Comparative analysis of 2D-PCA based dimensionality reduction and feature extraction. *Journal of Physics: Conference Series*, 2024(1):012034, 2021.
- [145] Jingyi Zhao, Chunhai Gao, Tao Tang. A review of sustainable maintenance strategies for single component and multicomponent equipment. *Sustainability*, 14(5), 2022.
- [146] Shuai Zheng, Kosta Ristovski, Ahmed Farahat, Chetan Gupta. Long short-term memory network for remaining useful life estimation. *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, strony 88–95, 2017.
- [147] Daniel Zuth, Tomáš Marada. Utilization of machine learning in vibrodiagnostics. *23rd International Conference on Soft Computing*, strony 271–278. Springer, 2017.