

Politechnika Poznańska
Poznan University of Technology



Wydział Automatyki, Robotyki i Elektrotechniki
Instytut Robotyki i Inteligencji Maszynowej

*Faculty of Control, Robotics and Electrical Engineering
Institute of Robotics and Machine Intelligence*

Rozprawa doktorska

**Detekcja uszkodzeń jednostek
napędowych w trakcie lotu
bezzałogowego statku powietrznego**

*Fault detection of unmanned aerial vehicle
propulsion units during flight*

Radostław Puchalski

Promotor: dr hab. inż. Wojciech Giernacki, prof. PP

Poznań, grudzień 2024

Streszczenie

Wraz z dynamicznym wzrostem liczby bezzałogowych statków powietrznych (BSP) operujących w przestrzeni lotniczej, zwiększa się liczba incydentów i niebezpieczeństw wynikających z uszkodzeń i anomalii pracy ich układów napędowych i wykonawczych. Obecnie duże wyzwanie stanowi wczesne i skuteczne wykrywanie wszelkich awarii i uszkodzeń, mogących skutkować niekontrolowanym zachowaniem drona w czasie jego eksploatacji. W BSP szczególnie narażone na uszkodzenia są śmigła, których kondycja warunkuje bezpieczeństwo i precyzję lotów autonomicznych. Tematyka rozprawy dotyczy diagnostyki stanu śmigieł wielowirnikowych dronów, przeprowadzanej w zautomatyzowany sposób w trakcie wykonywanego lotu. Autor w rozprawie stawia i udowadnia tezę, że taka diagnostyka może być prowadzona skutecznie w reżimie czasu rzeczywistego, przy użyciu zaproponowanych metod uczenia maszynowego, na bazie przetwarzania danych sensorycznych w mikrokontrolerze pokładowym drona.

W ramach prac zaprojektowano niezależny od systemów pokładowych drona układ akwizycji danych sensorycznych, który posłużył do zebrania pomiarów z czujników zamontowanych w pobliżu śmigieł. Sygnały rejestrowano w trakcie licznych kampanii lotów eksperymentalnych z użyciem wybranych modeli dronów, zarówno w przestrzeni miejskiej, jak i w laboratorium. Rozpatrywano scenariusze obejmujące różne typy usterek oraz dwa rozmiary uszkodzeń śmigieł w rozmaitych zestawieniach ich lokalizacji na ramionach BSP. Pomiary z akcelerometrów, żyroskopów oraz mikrofonów cyfrowych zebrano i upubliczniono w ogólnodostępnym repozytorium danych pomiarowych.

Na podstawie zgromadzonych danych przygotowano klasyfikatory uszkodzeń w postaci modeli sztucznej sieci neuronowej. Klasyfikator zaimplementowany w mikrokontrolerze przeprowadzał diagnostykę, określając rodzaj, miejsce i rozmiar uszkodzenia dwu- i trójpłatowych śmigieł. Podczas eksperymentów badano klasyfikatory przygotowane z użyciem różnych czujników i ich

kombinacji, korzystających z danych w dziedzinie czasu, wyekstrahowanych cech dziedziny częstotliwości oraz sygnatur czasowo-częstotliwościowych. Część modeli pozwoliła na diagnozowanie anomalii w reżimie czasu rzeczywistego, zapewniając klasyfikację przeprowadzaną 500 razy na sekundę.

W rozprawie zaproponowano także dwa nowe wskaźniki oceny jakości klasyfikatorów. Pierwszy z nich pozwala porównać różne klasyfikatory z uwzględnieniem ich dokładności oraz liczby rozpatrywanych klas. Drugi wskaźnik dotyczy systemów, w których czas detekcji jest wielkością krytyczną. Umożliwia on ocenę klasyfikatorów o różnej dokładności, liczbie klas oraz czasie przetwarzania.

W drodze badań eksperymentalnych wykazano wysoką skuteczność detekcji dla zaproponowanego podejścia w zakresie lokalizacji i identyfikacji uszkodzeń śmigieł wielowirnikowych BSP. Z uwagi na specyfikę lotów bezzałogowymi statkami powietrznymi klasy wielowirnikowej (mocno ograniczony udźwig i czas lotu), szczególnie duży nacisk położono na optymalizację zaproponowanych rozwiązań, aby:

- a) opracowany elektroniczny system pomiarowy był lekki i uniwersalny (możliwy do użycia w dowolnym dronie wielowirnikowym),
- b) zaproponowane metody przetwarzania sygnałów cechowała możliwie niska złożoność obliczeniowa.

Dzięki temu proponowane metody wykazują potencjał ich implementacji w systemach awioniki pokładowej dronów zarówno produkowanych seryjnie, jak i konstrukcji własnych.

Abstract

With the rapid expansion of the number of unmanned aerial vehicles (UAVs) operating in the airspace, the incidents and dangers resulting from damage and anomalies in the operation of their propulsion and actuation systems are increasing. Today, a major challenge is the early and effective detection of any malfunctions and faults that may result in uncontrolled behavior of the drone during its operation. In UAVs, particularly vulnerable to fault are propellers, the condition of which determines the safety and precision of autonomous flight. The subject of the dissertation concerns the condition diagnostics of the propellers of multi-rotor drones, carried out in an automated manner during the flight being performed. In the dissertation, the author posits and proves the thesis that such diagnostics can be carried out effectively in the real-time regime, using the proposed machine learning methods, based on the processing of sensory data in the drone's on-board microcontroller.

As part of the work, a sensory data acquisition system independent of the drone's onboard systems was designed and used to collect measurements from sensors mounted near the propellers. Signals were recorded during numerous experimental flight campaigns using selected drone models, both in urban space and in the laboratory. Scenarios involving different types of defects and two sizes of propeller faults in various sets of their locations on UAV arms were considered. Measurements from accelerometers, gyroscopes, and digital microphones were collected and made publicly available in a sensory data repository.

Based on the collected data, fault classifiers were prepared in the form of artificial neural network models. The classifier implemented in the microcontroller performed diagnostics, determining the type, location, and size of faults in two- and three-bladed propellers. During the experiments, classifiers prepared with various sensors and their combinations were tested, using time-domain data, extracted frequency-domain features, and time-frequency

signatures. Some of the models allowed for diagnosing anomalies in the real-time regime, providing classification performed 500 times per second.

The dissertation also proposes two new indicators for evaluating the quality of classifiers. The first one allows for the comparison of different classifiers with respect to their accuracy and the number of classes considered. The second indicator applies to systems in which detection time is a critical quantity. It allows the evaluation of classifiers with different accuracy, number of classes, and processing time.

Through experimental research, high detection performance was demonstrated for the proposed approach in locating and identifying UAV multi-rotor propeller faults. Given the nature of multi-rotor class drone flight (severely limited payload and flight time), particular emphasis was placed on optimizing the proposed solutions to:

- a) the developed electronic measurement system was lightweight and versatile (usable in any multi-rotor drone),
- b) the proposed signal processing methods had the lowest possible computational complexity.

As a result, the proposed methods show potential for their implementation in on-board avionics systems of both mass-produced drones and custom designs.

Spis treści

Wykaz skrótów i oznaczeń	IX
Spis rysunków	XIII
Spis tabel	XVII
1 Wstęp	1
1.1 Znaczenie problematyki badawczej	2
1.2 Cel pracy	4
1.3 Teza rozprawy	5
1.4 Podstawowe pojęcia	5
1.5 Przegląd zawartości rozprawy	8
2 Aktualny stan wiedzy	11
3 Sygnały i układy sensoryczne	21
3.1 Przyśpieszenia liniowe	21
3.2 Prędkości kątowe	22
3.3 Sygnały akustyczne	23
4 System akwizycji danych sensorycznych	25
4.1 Budowa układu pomiarowego	25
4.2 Działanie systemu akwizycji	28
4.3 Rozwój i modyfikacje systemu	31
5 Repozytorium danych sensorycznych	33
5.1 Dane z drona <i>Parrot Bebop 2</i>	34
5.2 Dane z drona <i>3DR Solo</i>	40
5.3 Ocena danych pod względem statystycznym	46
6 Diagnostyka uszkodzeń w czasie rzeczywistym	55
6.1 Opracowanie systemu detekcji uszkodzeń	56
6.2 Przygotowanie oprogramowania	56
6.2.1 Trenowanie klasyfikatora	56

6.2.2	Implementacja klasyfikatora w mikrokontrolerze	61
6.3	Działanie systemu detekcji	61
7	Testy skuteczności klasyfikatorów	65
7.1	Wskaźniki oceny jakości klasyfikatorów	65
7.1.1	Ocena klasyfikacji niezależnej od czasu przetwarzania .	70
7.1.2	Ocena klasyfikacji z krytycznym czasem przetwarzania	71
7.2	Sposób przeprowadzania eksperymentów	73
7.3	Sygnaly wyrażone w dziedzinie czasu	75
7.4	Ekstrakcja cech wyrażonych w dziedzinie częstotliwości	87
7.5	Ekstrakcja cech czasowo-częstotliwościowych	103
7.6	Detekcja uszkodzeń spoza zbioru treningowego modelu	106
7.7	Podsumowanie testów skuteczności klasyfikatorów	108
8	Analiza uzyskanych rezultatów	111
9	Podsumowanie	121
9.1	Istotny wkład w tematykę diagnozowania uszkodzeń BSP	123
9.2	Wnioski końcowe	125
	Bibliografia	127

Wykaz skrótów i oznaczeń

μSD karta pamięci mikro-SD, ang. *micro Secure Digital*

BGMM bayesowski model mieszanin rozkładów Gaussa, ang. *Bayesian Gaussian Mixture Model*

BLDC bezszczotkowy silnik prądu stałego, ang. *Brushless Direct-Current Motor*

BSP bezzałogowy statek powietrzny, ang. *Unmanned Aerial Vehicle*

CNN konwolucyjna sieć neuronowa, ang. *Convolutional Neural Network*

CrazyPAD dane anomalii śmigieł drona Crazyflie, ang. *Crazyflie Propeller Anomaly Data*

DFT dyskretna transformacja Fouriera, ang. *Discrete Fourier Transform*

DWT dyskretna transformacja falkowa, ang. *Discrete Wavelet Transform*

EMD empiryczna analiza modów, ang. *Empirical Mode Decomposition*

FD detekcja uszkodzeń, ang. *Fault Detection*

FDD detekcja i diagnostyka uszkodzeń, ang. *Fault Detection and Diagnosis*

FDI detekcja i izolacja uszkodzeń, ang. *Fault Detection and Isolation*

FFT szybka transformacja Fouriera, ang. *Fast Fourier Transform*

FN liczba przewidywań fałszywie negatywnych, ang. *False Negative*

FP liczba przewidywań fałszywie pozytywnych, ang. *False Positive*

- FTC** sterowanie odporne na uszkodzenia, ang. *Fault-Tolerant Control*
- HHT** transformacja Hilberta-Huanga, ang. *Hilbert-Huang Transform*
- I²C** zintegrowana magistrala szeregową, ang. *Inter-Integrated Circuit bus*
- IEC** Międzynarodowa Komisja Elektrotechniczna, ang. *International Electrotechnical Commission*
- IMF** wewnętrzna funkcja modów, ang. *Intrinsic Mode Function*
- IMU** czujnik inercyjny, ang. *Inertial Measurement Unit*
- KDE** jądrowy estymator gęstości, ang. *Kernel Density Estimation*
- LSTM** długa pamięć krótkich wzorców, ang. *Long Short Term Memory*
- MACC** złożoność mnożenia i akumulacji, ang. *Multiply-and-Accumulate Complexity*
- MEMS** mikroukład elektromechaniczny, ang. *MicroElectroMechanical System*
- MFCC** melowo-częstotliwościowe współczynniki cepstralne, ang. *Mel-Frequency Cepstral Coefficients*
- PADRE** repozytorium danych pomiarowych dotyczących uszkodzeń śmigieł, ang. *Propeller Anomaly Data REpository*
- PCB** płytki obwodu drukowanego, ang. *Printed Circuit Board*
- PCM** modulacja impulsowo-kodowa, ang. *Pulse-Code Modulation*
- PDM** modulacja gęstością impulsów, ang. *Pulse-Density Modulation*
- RAM** pamięć o dostępie swobodnym, ang. *Random-Access Memory*
- ReLU** rektyfikowana jednostka liniowa, ang. *Rectified Linear Unit*
- RMSProp** propagacja średniej kwadratowej, ang. *Root Mean Squared Propagation*
- SAI** interfejs szeregowy audio, ang. *Serial Audio Interface*

- SBC** komputer jednopłytkowy, ang. *Single Board Computer*
- SDIO** interfejs wejścia-wyjścia kart pamięci SD, ang. *Secure Digital Input Output*
- SI** sztuczna inteligencja, ang. *Artificial Intelligence*
- SSN** sztuczna sieć neuronowa, ang. *Artificial Neural Network*
- STFT** krótkookresowa transformacja Fouriera, ang. *Short-Time Fourier Transform*
- SVM** maszyna wektorów nośnych, ang. *Support Vector Machine*
- SWD** szeregowy interfejs programowania, ang. *Serial Wire Debug*
- TN** liczba przewidywań prawdziwie negatywnych, ang. *True Negative*
- UART** uniwersalny asynchroniczny nadajnik-odbiornik, ang. *Universal Asynchronous Receiver-Transmitter*
- UAV-RFD** baza danych rzeczywistych usterek drona, ang. *UAV-Realistic-Fault-Dataset*
- WPD** dekompozycja pakietów falkowych, ang. *Wavelet Packet Decomposition*

Spis rysunków

1.1	Etapy diagnostyki uszkodzeń (opracowanie na podstawie [100])	7
2.1	Popularność metod opartych na modelu oraz opartych na danych wśród artykułów zestawionych w [100]	12
3.1	Schemat ideowy jednoosiowego akcelerometru (opracowanie na podstawie [81, 74])	22
3.2	Schemat ideowy jednoosiowego żyroskopu (opracowanie na podstawie [81, 74])	23
3.3	Budowa mikrofonu MEMS (opracowanie na podstawie [68, 59])	24
4.1	Widok górnej i dolnej warstwy systemu do akwizycji danych z czujników	26
4.2	Tryb akwizycji (opracowanie na podstawie [103])	29
5.1	Repozytorium PADRE (opracowanie na podstawie [103])	34
5.2	<i>Parrot Bebop 2</i> z zamontowanym na kadłubie systemem akwizycji [103]	36
5.3	<i>3DR Solo</i> z zamontowanym pod kadłubem systemem akwizycji [103]	41
5.4	Wykresy jądrowego estymatora gęstości pomiarów z drona <i>Parrot Bebop 2</i> (opracowanie na podstawie [103])	48
5.5	Współczynniki korelacji Pearsona poszczególnych par osi czujników z drona <i>Parrot Bebop 2</i> (opracowanie na podstawie [103])	49
5.6	Wykresy jądrowego estymatora gęstości odczytów inercyjnych z drona <i>3DR Solo</i> (opracowanie na podstawie [103])	51
5.7	Wykresy jądrowego estymatora gęstości odczytów mikrofonowych z drona <i>3DR Solo</i> (opracowanie na podstawie [103])	52
5.8	Współczynniki korelacji Pearsona poszczególnych par osi czujników inercyjnych z drona <i>3DR Solo</i> (opracowanie na podstawie [103])	52
5.9	Współczynniki korelacji Pearsona poszczególnych mikrofonów z drona <i>3DR Solo</i> (opracowanie na podstawie [103])	52

6.1	Wykresy funkcji <i>ReLU</i> oraz krzywej logistycznej	59
6.2	Wykresy funkcji logistycznych z różnymi parametrami a (opracowanie na podstawie [119])	59
6.3	Wykresy dokładności i straty na etapie trenowania i walidacji modelu SSN	60
6.4	Schematyczna budowa bufora cyklicznego	63
6.5	Tryb detekcji (opracowanie na podstawie [103])	64
7.1	Przykładowa macierz błędów dla klasyfikatora czteroklasowego	66
7.2	Macierz błędów modelu dziedziny czasu dla 5 klas z pojedynczymi uszkodzeniami w postaci wyszczerbionej krawędzi śmigła dla szerokości okna wynoszącego 8 próbek	79
7.3	Macierz błędów modelu dziedziny czasu dla 5 klas z pojedynczymi uszkodzeniami w postaci wyszczerbionej krawędzi śmigła dla szerokości okna wynoszącego 2 próbki	80
7.4	Macierz błędów modelu dziedziny czasu dla 5 klas z pojedynczymi uszkodzeniami w postaci wygiętej końcówki śmigła . . .	81
7.5	Macierz błędów modelu dziedziny czasu dla 9 klas z pojedynczymi uszkodzeniami dwóch rodzajów	82
7.6	Macierz błędów modelu dziedziny czasu dla 20 klas uszkodzeń	84
7.7	Macierz błędów modelu dziedziny czasu dla 9 klas uszkodzeń o dwóch rozmiarach	85
7.8	Macierz błędów modelu dziedziny czasu dla 9 klas uszkodzeń o dwóch rozmiarach użytego podczas rzeczywistych testów terenowych	86
7.9	Porównanie metody drzew decyzyjnych oraz proponowanego klasyfikatora w postaci modelu SSN	92
7.10	Macierz błędów modelu dziedziny częstotliwości dla 5 klas z pojedynczymi uszkodzeniami w postaci wyszczerbionej krawędzi śmigła	94
7.11	Macierz błędów modelu dziedziny częstotliwości dla 5 klas z pojedynczymi uszkodzeniami w postaci wygiętej końcówki śmigła	95
7.12	Macierz błędów modelu dziedziny częstotliwości dla 9 klas z pojedynczymi uszkodzeniami dwóch rodzajów	96
7.13	Macierz błędów modelu dziedziny częstotliwości dla 20 klas uszkodzeń	97
7.14	Macierz błędów modelu dziedziny częstotliwości dla 20 klas uszkodzeń przeprowadzającego klasyfikację w czasie rzeczywistym	99

7.15	Macierz błędów modelu dziedziny częstotliwości dla 9 klas uszkodzeń o dwóch rozmiarach	100
7.16	Macierz błędów modelu dziedziny częstotliwości dla 9 klas uszkodzeń o dwóch rozmiarach użytego podczas rzeczywistych testów terenowych	101
7.17	Macierz błędów modelu dziedziny częstotliwości wykorzystującego sygnały dźwiękowe dla 9 klas uszkodzeń o dwóch rozmiarach	102
8.1	Wpływ liczby neuronów warstwy ukrytej na czas przetwarzania przez SSN (opracowanie na podstawie [102])	111
8.2	Wpływ liczby neuronów warstwy ukrytej na wartość dokładności i straty (opracowanie na podstawie [102])	112
8.3	Wpływ szerokości okna pomiarowego na czas obliczeń w poszczególnych etapach przetwarzania danych: zastosowanie funkcji okna Hanna, FFT, klasyfikacja SSN (opracowanie na podstawie [102])	113
8.4	Wpływ szerokości okna pomiarowego na wartość dokładności i straty (opracowanie na podstawie [102])	114
8.5	Wpływ częstotliwości próbkowania na wyniki klasyfikacji dla różnych szerokości okna pomiarowego dla sygnałów dziedziny czasu	115
8.6	Wpływ częstotliwości próbkowania na wyniki klasyfikacji dla różnych szerokości okna pomiarowego dla sygnałów dziedziny częstotliwości (FFT)	115
8.7	Wpływ częstotliwości próbkowania na wyniki klasyfikacji dla różnych szerokości okna pomiarowego dla sygnałów czasowo-częstotliwościowych DWT	116
8.8	Wpływ częstotliwości próbkowania na wyniki klasyfikacji dla różnych szerokości okna pomiarowego dla sygnałów czasowo-częstotliwościowych EMD	116
8.9	Wpływ częstotliwości próbkowania na wyniki klasyfikacji dla różnych szerokości okna pomiarowego dla sygnałów czasowo-częstotliwościowych HHT	116

Spis tabel

2.1	Klasyczny podział metod detekcji uszkodzeń (opracowanie na podstawie [100]).	11
2.2	Porównanie baz danych dotyczących detekcji uszkodzeń dronów wielowirnikowych.	18
4.1	Parametry mikrokontrolera STM32H743IIT6 [129].	26
5.1	Wybrane parametry drona <i>Parrot Bebop 2</i> [95].	35
5.2	Parametry modułu inercyjnego GY-6500 [61].	35
5.3	Wybrane parametry drona <i>3DR Solo</i> [1].	40
5.4	Parametry modułu inercyjnego GY-91 [60].	42
5.5	Parametry modułu mikrofonowego Adafruit 3492 [124].	43
5.6	Dane statystyczne z drona <i>Parrot Bebop 2</i> (opracowanie na podstawie [103]).	47
5.7	Dane statystyczne z drona <i>3DR Solo</i> (opracowanie na podstawie [103]).	50
7.1	Popularne metryki dla przykładowej macierzy błędów z rysunku 7.1.	69
7.2	Wartości wskaźnika Q_{AC} dla $\alpha = 3$ oraz zmiennych parametrów A i C	71
7.3	Wartości wskaźnika Q_{ACT} dla $\alpha = 3$, $T_0 = 2$ ms oraz zmiennych parametrów A , C i T	73
7.4	Parametry modelu dziedziny czasu dla 5 klas z pojedynczymi uszkodzeniami w postaci wyszczerbionej krawędzi śmigła dla szerokości okna wynoszącego 8 próbek.	77
7.5	Metryki dla poszczególnych klas uszkodzeń modelu dziedziny czasu uwzględniającego 5 klas z pojedynczymi uszkodzeniami w postaci wyszczerbionej krawędzi śmigła dla szerokości okna wynoszącego 8 próbek.	78
7.6	Parametry modelu dziedziny czasu dla 5 klas z pojedynczymi uszkodzeniami w postaci wyszczerbionej krawędzi śmigła dla szerokości okna wynoszącego 2 próbki.	80

7.7	Parametry modelu dziedziny czasu dla 5 klas z pojedynczymi uszkodzeniami w postaci wygiętej końcówki śmigła.	81
7.8	Parametry modelu dziedziny czasu dla 9 klas z pojedynczymi uszkodzeniami dwóch rodzajów.	82
7.9	Parametry modelu dziedziny czasu dla 20 klas uszkodzeń. . . .	83
7.10	Parametry modelu dziedziny czasu dla 9 klas uszkodzeń o dwóch rozmiarach.	84
7.11	Parametry modelu dziedziny czasu dla 9 klas uszkodzeń o dwóch rozmiarach użytego podczas rzeczywistych testów terenowych.	86
7.12	Porównanie skuteczności klasyfikacji w zależności od różnych parametrów przetwarzania sygnałów (opracowanie na podstawie [99]).	91
7.13	Parametry modelu dziedziny częstotliwości dla 5 klas z pojedynczymi uszkodzeniami w postaci wyszczerbionej krawędzi śmigła.	93
7.14	Parametry modelu dziedziny częstotliwości dla 5 klas z pojedynczymi uszkodzeniami w postaci wygiętej końcówki śmigła. . . .	94
7.15	Parametry modelu dziedziny częstotliwości dla 9 klas z pojedynczymi uszkodzeniami dwóch rodzajów.	96
7.16	Parametry modelu dziedziny częstotliwości dla 20 uszkodzeń. . . .	97
7.17	Parametry modelu dziedziny częstotliwości dla 20 uszkodzeń przeprowadzającego klasyfikację w czasie rzeczywistym.	98
7.18	Parametry modelu dziedziny częstotliwości dla 9 klas uszkodzeń o dwóch rozmiarach.	99
7.19	Parametry modelu dziedziny częstotliwości dla 9 klas uszkodzeń o dwóch rozmiarach użytego podczas rzeczywistych testów terenowych.	101
7.20	Parametry modelu dziedziny częstotliwości wykorzystującego sygnały dźwiękowe dla 9 klas uszkodzeń o dwóch rozmiarach.	102
7.21	Wyniki klasyfikacji 9 klas uszkodzeń drona <i>3DR Solo</i> z wykorzystaniem cech czasowo-częstotliwościowych.	105
7.22	Wyniki detekcji uszkodzeń, na których model nie był trenowany.	107
8.1	Podsumowanie wyników testów przebadanych klasyfikatorów uszkodzeń.	117
8.2	Porównanie jakości klasyfikacji różnych badań z uwzględnieniem wskaźnika Q_{AC}	118

W ostatnich latach można zaobserwować dynamiczny wzrost liczby zastosowań BSP (bezzałogowy statek powietrzny, ang. *Unmanned Aerial Vehicle*). Nieustannie rozszerzany jest zbiór zastosowań, w których wykorzystuje się drony¹. Z powodzeniem są one stosowane w działaniach militarnych [144, 106, 136], inspekcji instalacji energetycznych [30, 49] i innych konstrukcji [154, 54, 122], monitorowaniu pożarów lasów [56], rolnictwie [6], udzieleniu pomocy medycznej [67], edukacji [118], logistyce [79, 33], monitoringu morskim [146], tworzeniu sieci komunikacyjnych [96] oraz w celach czysto rekreacyjnych [66].

Coraz powszechniejsza obecność BSP w środowisku zurbanizowanym i otoczeniu bliskim człowiekowi niesie za sobą szereg zagrożeń. Zwiększa się liczba incydentów związanych z rosnącą popularnością tego typu konstrukcji latających, począwszy od aspektów dotyczących możliwości użycia BSP w danym miejscu i czasie przez uprawnionego operatora [89, 31], poprzez czynniki związane z naruszeniami prywatności [70], kończąc na niebezpieczeństwie rozbicia drona [133]. O ile pierwsze dwa zagrożenia mają podłoże prawno-etyczne, o tyle awaria w trakcie lotu może skutkować uszkodzeniami infrastruktury oraz stwarzać realne zagrożenie dla ludzi i zwierząt znajdujących się w pobliżu trasy przelotu BSP.

W pracy [51] z 2022 r. przeanalizowano 154 wypadki i incydenty związane z użytkowaniem BSP. Statystyki pokazują, że ryzyko wystąpienia nieprawidłowości lub wypadku podczas każdej godziny lotu dronem wynosi ok. 19%. Analizując przyczyny awarii, okazało się, że 19,35% z nich związanych jest z usterkami samego BSP. Pozostałe powody to błędy ludzkie (37,66%) oraz czynniki zewnętrzne (12,99%). Artykuł [45] opisuje z kolei 77 wypadków średnich i dużych BSP, które miały miejsce w latach 2013-2024. Dotyczą one czynników ludzkich, które przyczyniły się do niewłaściwego zachowania drona. Ciekawą obserwacją stanowi odsetek wypadków lub incydentów, które wystąpiły podczas faz operacyjnych (tj. kontroli przed lotem, planowania oraz wykonywania lotów), wynoszący aż 54%. Inne badanie [132] z 2024 r. skupia się na ocenie ryzyka wypadków BSP. Jako najczęstsze

¹W kontekście rozprawy zamiennie stosowane są nazwy *bezzałogowy statek powietrzny*, *bezzałogowy pojazd latający* oraz *dron*.

przyczyny niepożądanych zdarzeń wskazuje się m.in. inny statek powietrzny, błędy operatora, zanik zasilania i niekorzystną pogodę. W toku analizy raportów z wypadków stwierdzono, iż w celu zapewnienia bezpieczeństwa lotów konieczna jest regularna konserwacja śmigieł. Brakuje jednak oficjalnych wytycznych dotyczących tego, kiedy należy je wymienić.

Z uwagi na fakt, że liczba BSP będących w użytkowaniu specjalistycznych służb oraz pasjonatów robotów latających wciąż rośnie, coraz ważniejszym aspektem staje się zapewnienie bezpieczeństwa lotów. Skutki nieprawidłowego działania statku powietrznego mogą narazić na niebezpieczeństwo zranienia osoby postronne lub spowodować naruszenie mienia znajdującego się w rejonie działań drona, a także kosztowne uszkodzenia samego pojazdu latającego. Stąd tak ważne staje się zapewnienie wczesnego wykrywania i klasyfikacji wszelkich anomalii, które mogą skutkować wystąpieniem awarii BSP.

1.1 Znaczenie problematyki badawczej

Drony często wykonują zadania krytyczne lub działają w bliskiej odległości od ludzi. W związku z tym bezpieczeństwo ich użytkowania stało się istotnym zagadnieniem. Awaria aktuatora może doprowadzić do niekontrolowanego przyziemienia i w efekcie do rozbicia drona. Zadania wykonywane przez BSP muszą być zabezpieczone przed wystąpieniem anomalii, ponieważ potencjalnie każda awaria podczas lotu może skutkować poważnymi konsekwencjami. Koszty utrzymania i obsługi floty dronów stają się także coraz istotniejsze. Poszczególne elementy pojazdów naturalnie się zużywają. W przewidywanej przyszłości, przy spodziewanym dużym zagęszczeniu pojazdów latających, badanie stanu technicznego przeprowadzane wyłącznie przez ludzi może okazać się trudne do realizacji. Inspekcja przed rozpoczęciem i po zakończeniu misji wymaga czasu oraz zatrudnienia załogi do jej wykonania, natomiast inspekcja w locie najczęściej opiera się na procedurach automatycznych. Zużycie lub drobne uszkodzenia komponentów nie zawsze są dostrzegalne dla ludzi, dlatego też opracowywanie nowych, skutecznych metod FD (detekcja uszkodzeń, ang. *Fault Detection*) jest zadaniem priorytetowym. Odpowiednia diagnostyka, w tym określenie typu, lokalizacji i stopnia uszkodzeń, pozwala na wczesne wykrycie nieprawidłowości. Takie informacje mogą posłużyć do zaalarmowania zdalnego operatora, który podejmie decyzję o kontynuacji lotu, awaryjnym lądowaniu drona lub zastosowaniu technik

FTC (sterowanie odporne na uszkodzenia, ang. *Fault-Tolerant Control*), aby kontynuować lot pomimo awarii [105].

Naukowcy od lat pracują nad efektywnymi metodami automatycznej diagnostyki i klasyfikacji [37]. Zaproponowane dotychczas rozwiązania są zwyczajowo dzielone na dwie podstawowe grupy: metody bazujące na modelu oraz metody bazujące na danych [100]. Pierwsze z nich wymagają znajomości matematycznego modelu wykorzystywanego drona. Im bardziej zawiły jest model, tym dokładniej może on odwzorować rzeczywisty obiekt. Większy stopień złożoności wiąże się jednak ze skomplikowanymi i czasochłonnymi obliczeniami. Ponieważ modele są przybliżonymi odpowiednikami rzeczywistych statków powietrznych, metody te zawsze charakteryzują się pewnym poziomem niepewności, a ich adekwatność daleka jest od oczekiwanej. Z tego powodu coraz więcej badań opiera się na podejściach bezmodelowych. W metodach bazujących na danych badacze wykorzystują szeroki zakres pomiarów z czujników, z których uzyskują parametry elektryczne lub różnorodne charakterystyki statyczne i dynamiczne lotu. Najczęściej skupiają się na pomiarach z czujników inercyjnych. Przewodzą tutaj akcelerometry [115, 48, 102], żyroskopy [139, 140, 99] oraz magnetometry [109, 24, 114]. Coraz częściej stosowane są również pomiary akustyczne [5, 44, 69]. Do najczęstszych komponentów wielowirnikowych BSP, których uszkodzenia można wykrywać tymi metodami, należą silniki, śmigła oraz same czujniki [38]. Śmigło, jako element ruchomy i często wystający poza kadłub drona, uważane jest za najbardziej podatne na uszkodzenia [51]. Z jednej strony jest to stosunkowo tani i łatwo wymienialny komponent, z drugiej jednak, pomimo ogromnego wpływu na stabilność i bezpieczeństwo lotu, śmigła często są stosunkowo mało wytrzymałymi elementami.

Współczesne mikrokontrolery posiadają wydajność obliczeniową wystarczającą do realizacji zaawansowanych zadań cyfrowego przetwarzania sygnałów [83]. Ponieważ są one niewielkich rozmiarów i cechują się stosunkowo niskim zapotrzebowaniem energetycznym, często są wykorzystywane w roli jednostek obliczeniowych montowanych na pokładach BSP. Odkąd wykazują zdolność przetwarzania w ramach technik zaliczanych do metod uczenia maszynowego, stały się idealnym kandydatem na potrzeby wykrywania uszkodzeń wspomaganego np. sztucznymi sieciami neuronowymi. Choć obecnie większość badaczy wykonuje obliczenia albo na stacjonarnych komputerach osobistych, albo mniejszych komputerach klasy SBC (komputer jednopłytkowy, ang. *Single Board Computer*), to właśnie mikrokontrolery

jawią się jako narzędzie najbardziej perspektywiczne. Dzięki energooszczędności i małej wadze wydają się optymalnym wyborem do praktycznych zastosowań w dziedzinach wykrywania usterek oraz sterowania odpornego na uszkodzenia bezzałogowych pojazdów latających.

1.2 Cel pracy

Obecnie brakuje szybkich i skutecznych rozwiązań realizujących detekcję uszkodzeń elementów wykonawczych BSP w trakcie jego lotu. W Instytucie Robotyki i Inteligencji Maszynowej Politechniki Poznańskiej od lat prowadzone są prace obejmujące diagnostykę podzespołów drona. Na bazie wcześniejszych doświadczeń, w szczególności opisanych w dysertacji [15], postanowiono rozwinąć stosowane dotychczas podejścia, opracować własne metody oraz zapewnić unikalne, jak na tego typu zastosowania, procedury przetwarzania danych i realizacji poszczególnych etapów procesu diagnozowania BSP. Badania przedstawione w niniejszej rozprawie bezpośrednio lub pośrednio dążą do wypracowania technik umożliwiających:

- przeprowadzenie w trakcie lotu drona automatycznej diagnostyki (detekcji, lokalizacji i identyfikacji) uszkodzeń z wykorzystaniem dedykowanych urządzeń pomiarowych montowanych na pokładzie dowolnego drona wielowirnikowego,
- wykrywanie w czasie rzeczywistym (pomiędzy kolejnymi pomiarami z sensorów) anomalii i uszkodzeń śmigieł drona, zarówno przed startem, jak i w trakcie lotu,
- określenie rodzaju usterki, jej rozmiaru oraz miejsca wystąpienia,
- dostarczenie informacji o wykrytych nieprawidłowościach niezbędnych do podjęcia decyzji o możliwości kontynuowania lotu operatorowi lub oprogramowaniu wykonującemu zadania FTC,
- implementację zastosowanych metod dla komercyjnych i prototypowych (autorskich) BSP celem niskokosztowego zwiększenia ich niezawodności.

1.3 Teza rozprawy

W wyniku analizy problemu omówionego w podrozdziale 1.1, sformułowano następującą tezę badawczą:

Możliwa jest automatyczna i precyzyjna diagnostyka uszkodzeń śmigieł bezzałogowego statku powietrznego podczas lotu, prowadzona w czasie rzeczywistym, oparta na uczeniu maszynowym wykorzystanym do przetwarzania sygnałów pomiarowych w dziedzinie czasu, z użyciem jedynie podstawowego mikrokontrolera.

Zdefiniowano trzy tezy szczegółowe, których potwierdzenie wykaże prawdziwość tezy głównej.

- Skuteczna klasyfikacja uszkodzeń różnych typów śmigieł wielowirnikowych dronów może być realizowana w czasie rzeczywistym przy użyciu metod automatycznego przetwarzania cyfrowych sygnałów w dziedzinie czasu z dodatkowego układu sensorycznego.
- W diagnostyce stanu elementów wykonawczych drona można zapewnić przetwarzanie sygnałów, dające informację o lokalizacji i rozmiarze uszkodzenia, realizowane w trybie ciągłym, na bazie jedynie mikrokontrolera.
- Wiarygodną ocenę skuteczności klasyfikatorów w określonych warunkach pracy można przeprowadzić, wykorzystując nowe, wielokryterialne wskaźniki oceny.

1.4 Podstawowe pojęcia

Uszkodzenie to każde zdarzenie eksploatacyjne, które w sposób bezpośredni lub pośredni powoduje pogorszenie jakości działania obiektu i jako takie powinno być wykrywane w procesie diagnozowania [71]. W rozprawie stosowane są również zamiennie synonimy, takie jak anomalia, awaria, defekt czy usterka.

Diagnostyka (lub diagnozowanie) uszkodzeń to ogół czynności mających na celu ocenę stanu badanego obiektu [71]. Zwykle proces ten dzieli się na trzy etapy:

- detekcja – wykrycie usterki i określenie chwili jej wystąpienia,
- lokalizacja – wskazanie rodzaju (typu) i miejsca uszkodzenia,
- identyfikacja – określenie rozmiaru, charakteru i wpływu nieprawidłowości na działanie systemu.

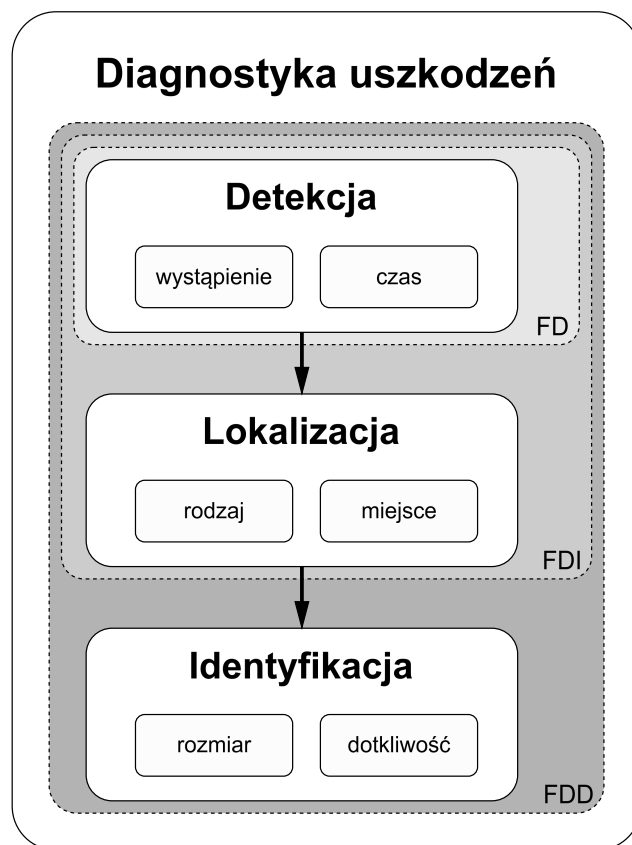
Podział ten został przyjęty przez Komitet Techniczny SAFEPROCESS międzynarodowej organizacji IFAC (*International Federation of Automatic Control*) w 1996 roku [62] i jest powszechnie stosowany w diagnostyce technicznej [40, 117, 50, 88, 100].

Standardowa procedura diagnozowania systemów przewiduje, w zależności od potrzeb, przechodzenie kolejno od etapu FD, którego celem jest sama detekcja, poprzez FDI (detekcja i izolacja uszkodzeń, ang. *Fault Detection and Isolation*), przeprowadzający dodatkowo procedurę lokalizacji, aż do FDD (detekcja i diagnostyka uszkodzeń, ang. *Fault Detection and Diagnosis*), który ponadto dokonuje identyfikacji. Poszczególne fazy zagnieżdżają się od najbardziej obszernego FDD, przez FDI, do FD, co graficznie przedstawiono na rysunku 1.1.

W rozprawie poruszane są zagadnienia dotyczące określania aktualnego stanu obiektu. Natomiast procesami opisującymi stany przeszłe zajmuje się genezowanie, a przewidywaniem przyszłych stanów – prognozowanie.

Ilekcioć w tekście pojawi się określenie *czas rzeczywisty*, oznaczać ono będzie takie operacje, które przeprowadzane są w całości pomiędzy pobraniem dwóch wartości sygnału, na podstawie których przeprowadzane jest wnioskowanie. Wszelkie przetwarzanie danych musi zakończyć się, zanim nastąpi kolejne próbkowanie. Zatem systemem czasu rzeczywistego będzie taki system, w którym całkowity czas przetwarzania danych jest krótszy od okresu próbkowania.

Problematyka rozprawy dotyczy uszkodzeń śmigieł, jednak opisywane metody można wykorzystać w szerszej diagnostyce układów napędowych BSP. W związku z tym, w treści pojawiają się odniesienia również do innych



Rys. 1.1: Etapy diagnostyki uszkodzeń (opracowanie na podstawie [100])

komponentów odpowiadających za wytworzenie siły nośnej i wykonywanie manewrów. Najważniejsze z nich to:

- silnik – urządzenie zmieniające energię elektryczną (lub inną) na energię mechaniczną w celu wytworzenia ruchu; w kontekście BSP, silnik napędza wirniki lub śmigła, które generują ciąg lub siłę nośną, umożliwiając lot,
- wirnik – zespół obrotowych łopatek przymocowanych do piasty, który generuje siłę nośną lub ciąg w wyniku ruchu obrotowego,
- rotor – termin ogólny, obejmujący zespół wirujących elementów zapewniających siłę nośną lub ciąg (m.in. wirnik),
- śmigło – zespół składający się z jednej lub wielu łopatek zamontowanych na piastce; służy do generowania siły nośnej w wielowirnikowych BSP oraz ciągu w konstrukcjach stałopłatowych i hybrydowych,
- łopata – pojedynczy element aerodynamiczny, który jest częścią śmigła lub wirnika.

1.5 Przegląd zawartości rozprawy

Rozprawa składa się z dziewięciu rozdziałów poprzedzonych streszczeniem w języku polskim i angielskim, spisu treści, wykazu skrótów i oznaczeń, spisu rysunków oraz spisu tabel.

Pierwszy rozdział obejmuje wprowadzenie do tematyki uszkodzeń i ich detekcji w kontekście BSP. Definiuje znaczenie problematyki badawczej w odniesieniu do opisywanych w literaturze zagrożeń związanych z awariami elementów wykonawczych drona. Nakreśla także możliwości oferowane przez współcześnie dostępną aparaturę pomiarową i obliczeniową. W rozdziale tym określono również cele oraz tezy badawcze, a także wyjaśniono najważniejsze terminy związane z tematyką diagnostyki bezzałogowych statków powietrznych.

W drugim rozdziale nakreślono obecny stan wiedzy dotyczący FD dronów. Przedstawiono najczęściej stosowane podejścia oraz osiągnięcia z ich zastosowaniem. Wskazano także ograniczenia i problemy, z którymi mierzą się naukowcy pracujący nad diagnostyką BSP.

Kolejna część stanowi zwięzłe przedstawienie układów sensorycznych i danych pomiarowych wykorzystywanych w badaniach opisanych w rozprawie. Scharakteryzowano w niej najczęściej spotykane konstrukcje omawianych czujników oraz sposób generowania sygnałów wyjściowych. Omówione zostały akcelerometry, żyroskopy i mikrofony cyfrowe produkowane w technologii MEMS (mikroukład elektromechaniczny, ang. *MicroElectroMechanical System*). Zaprezentowano także możliwości rozszerzenia obszaru stosowalności tych układów w odniesieniu do tematyki rozprawy.

Prezentację i opis systemu akwizycji danych sensorycznych z czujników zamontowanych na pokładzie BSP zawarto w rozdziale czwartym. Wskazano w nim szczegółową budowę autorskiego układu pomiarowego oraz sposób działania systemu. Scharakteryzowano poszczególne etapy działania opracowanego algorytmu, a także wskazano na przeprowadzone modyfikacje, poprawiające parametry pracy układu.

Rozdział piąty prezentuje strukturę i sposób korzystania z ogólnodostępnego repozytorium danych pomiarowych, które powstało dzięki użyciu czujników opisanych w rozdziale trzecim oraz systemu akwizycji przed-

stawionego w rozdziale czwartym. Wskazano w nim parametry statków powietrznych, które uczestniczyły w przeprowadzanych eksperymentach, a także użytych układów pomiarowych. Omówiono procedurę rejestrowania danych inercyjnych i sygnałów akustycznych. Przedstawiono również sposób prezentacji danych w repozytorium. Ostatni podrozdział poświęcony został opisowi zgromadzonych i udostępnionych publicznie pomiarów pod względem parametrów statystycznych.

Szósty rozdział zawiera opis prac nad oprogramowaniem służącym do przetwarzania danych w ramach diagnostyki przeprowadzanej w czasie rzeczywistym. Obejmuje proces trenowania klasyfikatora uszkodzeń oraz jego implementację w systemie wbudowanym. Przedstawia także algorytm działania gotowego układu, wskazując sposób funkcjonowania poszczególnych etapów przetwarzania.

Najbardziej obszerną część stanowi rozdział siódmy. Zaprezentowano w nim powszechnie używane wskaźniki oceny jakości klasyfikatorów. Wskazano niedoskonałości w ich stosowaniu oraz zaproponowano dwie nowe metryki, stanowiące rozwiązanie dla istniejących ograniczeń. W rozdziale tym opisano także sposób przeprowadzania eksperymentów walidacyjnych dla opracowanych klasyfikatorów. Zamieszczono w nim wyniki badań eksperymentalnych różnorodnych modeli SSN (sztuczna sieć neuronowa, ang. *Artificial Neural Network*) dokonujących klasyfikacji dla zaproponowanych scenariuszy testowych.

Rozdział ósmy to analiza uzyskanych wyników eksperymentalnych. W rozdziale przedstawiono wpływ poszczególnych parametrów przetwarzania na skuteczność klasyfikacji oraz na występujące opóźnienia w procesie uzyskiwania jej wyniku. Zamieszczono także porównanie otrzymanych rezultatów z wynikami innych badań dotyczących klasyfikacji uszkodzeń śmigieł BSP.

Rozprawę podsumowano w rozdziale dziewiątym. Dokonano w nim oceny stopnia realizacji założonych celów w kontekście oryginalności uzyskanych wyników na tle obecnego stanu wiedzy. Wskazano kluczowe wnioski, określono kierunki dalszych badań i potencjalne możliwości rozwoju stosowanych metod.

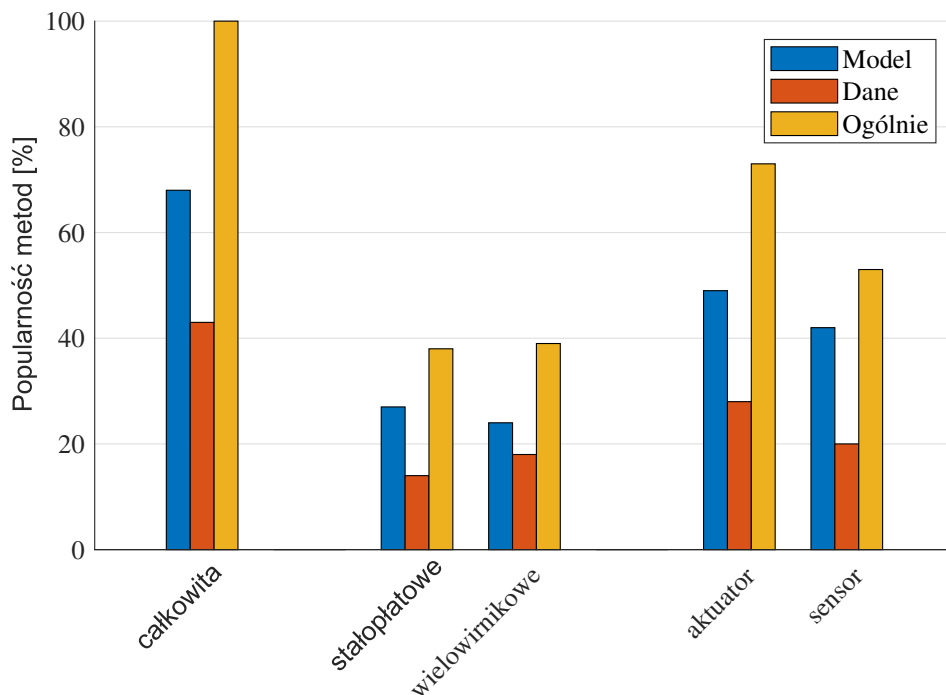
Monitorowanie stanu jednostek napędowych dronów, realizowane w sposób zautomatyzowany i ciągły podczas lotu, staje się coraz bardziej dostrzeżoną koniecznością. Zapewnienie bezpieczeństwa misji jest jednym z kluczowych obszarów zainteresowania badaczy zajmujących się tematyką BSP [113]. Jest to naturalny trend związany z rosnącą popularnością i coraz szerszym zakresem zastosowań tego typu pojazdów. Opracowanie szybkich i skutecznych metod diagnostyki elementów wykonawczych, w kontekście statystyk wypadków z udziałem dronów przytoczonych w rozdziale 1, nabiera charakteru priorytetowej potrzeby. Zaniedbanie tego aspektu może znacząco spowolnić dalszy rozwój technologii bezzałogowych statków powietrznych.

Rozwiązania, które dziś stosuje się w odniesieniu do dronów, są w dużej mierze kontynuacją wcześniejszych badań, które podejmowały problematykę uszkodzeń załogowych statków powietrznych oraz innych maszyn [97, 62]. W pracy [100] z 2022 r. dokonano systematycznego przeglądu literaturowego najnowszych metod detekcji uszkodzeń BSP. Obejmowała ona wyniki badań naukowych opublikowanych w okresie od początku 2016 r. do końca sierpnia 2022 r. Szczegółowo przeanalizowano 4 inne prace przeglądowe oraz 79 artykułów zawierających oryginalne rozwiązania. Wyłania się z nich wyraźny podział na dwie grupy metod: opartych na modelu matematycznym (ang. *model-based*) oraz korzystających z wiedzy bazującej na zebranych danych (ang. *data-driven*). Nie jest to jedyny możliwy podział, ale jawi się jako najbardziej powszechny [138]. W tabeli 2.1 wyszczególniono najpopularniejsze metody stosowane w ramach tych dwóch wyszczególnionych grup.

Tab. 2.1: Klasyczny podział metod detekcji uszkodzeń (opracowanie na podstawie [100]).

Metody oparte na modelu	Metody oparte na danych
estymacja stanu	metody statystyczne
estymacja parametrów	sieci neuronowe
identyfikacja procesu	logika rozmyta
estymacja stanu i parametrów	analiza częstotliwościowa
przestrzeń parzystości	analiza czasowo-częstotliwościowa

W omawianej pracy aż 45 artykułów zakwalifikowano do pierwszej grupy, a tylko 25 do drugiej. 9 publikacji nie udało się przypisać jednoznacznie do żadnej grupy. Opisywane w nich rozwiązania korzystały z mieszanych metod, stosujących tak modele obiektu, jak i używających technik typowo przypisywanych strategiom opartym na danych. Na rysunku 2.1 zaprezentowano w ujęciu procentowym popularność poszczególnych metod na bazie artykułów przedstawionych i przeanalizowanych w [100].



Rys. 2.1: Popularność metod opartych na modelu oraz opartych na danych wśród artykułów zestawionych w [100]

Wykresy przedstawiają tendencję wyboru metody należącej do danej grupy ogólnie dla wszystkich opisanych artykułów, ale także z podziałem na typ BSP oraz rodzaj elementu, którego uszkodzenie było badane. Etykieta *Model* odnosi się do metod opartych na modelu (włącznie z metodami mieszanymi), a etykieta *Dane* dotyczy modeli opartych na danych (oraz mieszanych). Odsetek wszystkich analizowanych prac dotyczących poszczególnej kategorii oznaczono etykietą *Ogólnie*. Ze względu na fakt, iż niektóre prace korzystały zarówno z metod opartych na modelu, jak i metod opartych na danych, a część badań obejmowała tak drony stałopłatowe, jak wielowirnikowe lub dotyczyła jednocześnie usterek aktuatorów i czujników, poszczególne wartości nie sumują się ani do poziomu wskazanego przez etykietę *Ogólnie*, ani do 100%. Pokazują jedynie, jak często badacze podejmują konkretną problematykę lub jak często korzystają z danej grupy metod w badaniach detekcji uszkodzeń dronów.

Okazuje się, że spośród wszystkich omawianych prac korzystających z modeli matematycznych oraz obejmujących metody mieszane, żadna nie podejmuje problemu wykrywania usterek śmigieł z użyciem rzeczywistych dronów wielowirnikowych. Wszystkie badania z tych grup rozpatrujące uszkodzenia rotorów przeprowadzają testy numeryczne lub korzystają z symulowanych awarii. Natomiast wśród prac opisujących metody bazujące na wiedzy znalazło się 6, które demonstrują rzeczywiste eksperymenty na multirotorach, w tym 1 korzystająca z drona na uwięzi.

W artykule [112] Saied i in. przedstawiają aktywne, odporne na uszkodzenia podejście do sterowania wielowirnikowym BSP. Wykrywanie błędów i diagnostyka usterek silników i wirników są wykonywane przy pomocy pomiaru prądów oraz prędkości obrotowych bezszczotkowych silników elektrycznych. Jako klasyfikatora użyto maszyny wektorów nośnych. Eksperymenty przeprowadzono na ośmiowirnikowym dronie o współosiowym układzie silników. Sprawdzone skuteczność proponowanego systemu w dwóch scenariuszach. Pierwszy to jednoczesna awaria dwóch silników, realizowana przez wysłanie komendy do zatrzymania ich pracy. Drugi z kolei dotyczy utraty śmigła, która symulowana jest przez jego fizyczne odłączenie jeszcze przed startem. Próby potwierdziły prawidłowe wykrycie i zlokalizowanie testowanych usterek. Na pokładzie BSP zaimplementowano algorytm SVM (maszyna wektorów nośnych, ang. *Support Vector Machine*) aktualizujący wyniki przewidywania 100 razy na sekundę. Nie przedstawiono skuteczności użytej metody, a jedynie fakt jej prawidłowego zadziałania podczas pojedynczej próby.

Iannace i in. w pracy [58] przedstawili metodę diagnozowania usterek śmigieł czterowirnikowego drona wykorzystującą analizę sygnałów dźwiękowych. Autorzy przeprowadzili detekcję niewyważenia łopat na BSP umocowanym na statywie. W odległości 1,2 m od drona znajdował się mikrofon, który zbierał sygnały akustyczne emitowane przez wirniki. Pomiarów wykonywano z czterech stron pojazdu w każdym z 3 przypadków: wyważone śmigła, śmigła z dodatkowym paskiem papieru na jednej łopacie oraz z dodatkowymi paskami papieru na obu łopatach śmigła. Po zebraniu danych wyodrębniono 31 cech częstotliwościowych dla każdej z trzech klas. Wykorzystano w tym celu zestaw filtrów pasmowo-przepustowych i wyznaczenie poziomu energii dla każdego z badanych pasm widma. Zestaw cech został podzielony na zbiory treningowe i testowe. Zbiór treningowy posłużył do wytrenowania modelu wielowarstwowej jednokierunkowej sieci neuronowej, natomiast

zbiór testowy miał zastosowanie w sprawdzeniu skuteczności wytrenowanego klasyfikatora. Dokładność wyniosła 97,63%. Przedstawiona metoda testowania drona może być stosowana wewnątrz pomieszczeń w procedurze przedstartowej w celu wykrycia wszelkich nieprawidłowości przed rozpoczęciem misji. Nie nadaje się jednak do badania uszkodzeń powstałych podczas wykonywanego lotu.

Z kolei w badaniu [109] Sadhu i in. zaproponowali nowatorską architekturę opartą na CNN (konwolucyjna sieć neuronowa, ang. *Convolutional Neural Network*) i LSTM (długa pamięć krótkich wzorców, ang. *Long Short Term Memory*) do identyfikacji uszkodzeń na podstawie odczytów z czujników inercyjnych. Korzystając z pomiarów z akcelerometru, żyroskopu i magnetometru klasyfikowali awarie śmigieł w 9 różnych przypadkach. Awarie wywoływane były poprzez programowe wymuszenie zerowej wartości prędkości obrotowej silników. Planowane było przetestowanie 15 scenariuszy uszkodzeń, ale ograniczenia oprogramowania wykluczyły niektóre kombinacje. Korzystając z autoenkodera, uzyskano dokładność klasyfikacji na poziomie 99% dla danych symulacyjnych. Podczas testów eksperymentalnych z wykorzystaniem prawdziwego BSP, dokładność spadła do 85%. Czas potrzebny na uzyskanie wyniku klasyfikacji wynosił ok. 100 ms.

Pozostałe trzy prace dotyczą prac przeprowadzonych przez zespół badaczy z Instytutu Robotyki i Inteligencji Maszynowej Politechniki Poznańskiej. W artykule konferencyjnym [18] z 2017 roku przedstawiono metodę wykrywania fizycznych uszkodzeń łopat wirnika, które skutkują zmniejszeniem ciągu BSP klasy mikro. Przetestowano trzy różne techniki przetwarzania sygnałów: FFT (szybka transformacja Fouriera, ang. *Fast Fourier Transform*), WPD (dekompozycja pakietów falkowych, ang. *Wavelet Packet Decomposition*) i pomiar mocy sygnału w poszczególnych pasmach częstotliwości. Do diagnozowania usterek wykorzystano trzy różne klasyfikatory w postaci SVM, których zadaniem jest wykrycie faktu wystąpienia usterki, określenie jej dokliwości oraz rodzaju defektu. W eksperymencie testowano uszkodzoną krawędź i zniekształconą końcówkę łopaty śmigła. W ramach badań najpierw przygotowano stacjonarne stanowisko dynamometryczne z silnikiem BLDC (bezszcotkowy silnik prądu stałego, ang. *Brushless Direct-Current Motor*) i przeprowadzono serię prób z trzema śmigłami o różnych typach i stopniach uszkodzenia. Następnie wykonano 10 lotów eksperymentalnych z wykorzystaniem wielowirnikowca składającego się z czterech jednostek napędowych, każda z współosiową parą wirników. Do analizy wykorzystano dane z dwóch

osi pokładowego akcelerometru. Porównano wpływ długości bufora danych oraz zastosowanej techniki przetwarzania sygnału na skuteczność poszczególnych etapów diagnostyki usterek. Zaproponowana metoda okazała się uniwersalna i stosunkowo prosta w implementacji. Jednak ze względu na centralną lokalizację IMU (czujnik inercyjny, ang. *Inertial Measurement Unit*), niemożliwe okazało się ustalenie, w którym wirniku wystąpiła awaria. Najwyższy osiągnięty współczynnik poprawnego wykrycia uszkodzenia osiągnął wartość poniżej 90%.

W kolejnym badaniu [17] ponownie wykorzystano pomiary z dwóch osi akcelerometru do wykrywania uszkodzeń śmigieł. Tym razem jednak posłużono się czterema jednostkami IMU, co pozwoliło na określenie lokalizacji niesprawnego elementu. Stosując FFT i zawężając spektrum badanych częstotliwości do środkowych wartości uzyskanego widma oraz wykorzystując metodę drzew decyzyjnych, przeprowadzono klasyfikację 7 scenariuszy zawierających – oprócz sprawnego zestawu – 4 pojedyncze oraz 2 podwójne uszkodzenia śmigieł. Klasyfikator uzyskał dokładność wynoszącą 93,9% dla pojedynczych uszkodzeń, ale dużo gorzej poradził sobie z defektami śmigieł występującymi jednocześnie na dwóch ramionach drona. Wskaźnik poprawności samej detekcji uszkodzeń wyniósł ponad 95%, a ogólna dokładność klasyfikacji osiągnęła poziom 86,57%.

Artykuł [19] z 2022 roku bazuje na zupełnie innych danych pomiarowych. Badacze umieścili moduł z 4 mikrofonami na pokładzie BSP. Zbierał on dane akustyczne z rotorów, które wyposażono w śmigła o dwóch typach uszkodzeń: odłamana końcówka oraz wyszczerbiona krawędź łopaty. W sumie wyszczególniono 21 klas zawierających pojedyncze i podwójne uszkodzenia. Korzystając z analizy cepstralnej zebranych danych, zarówno z użyciem klasyfikatora w postaci CNN, jak i LSTM, uzyskano dokładność FDI przewyższającą 98%. Jednakże zastosowaną metodę MFCC (melowo-częstotliwościowe współczynniki cepstralne, ang. *Mel-Frequency Cepstral Coefficients*) cechuje duża złożoność obliczeniowa.

Wśród najnowszych prac, których nie obejmował przegląd [100], pojawia się coraz więcej badań korzystających z rzeczywistych danych. Często jednak prawdziwa usterka w układzie napędowym BSP skutkuje potencjalnie dużym ryzykiem, stąd badacze posiłkują się danymi symulacyjnymi. Przykładem mieszanego podejścia, w którym wykorzystano zarówno pomiary z pokładowego IMU, jak i sztucznie wygenerowane, jest praca [27]. Bada-

niu poddano zjawisko całkowitej utraty mocy w silniku wielowirnikowca o 16 rotorach umieszczonych współosiowo na 8 ramionach. Eksperymenty podzielono na dwa etapy. W pierwszym testowano wykrycie usterki w jednym z 4 regionów, na który został podzielony system napędowy BSP. Każdy region zawierał 4 silniki, których usterki lokalizowano w drugim etapie. Autorzy podają ok. 99% skuteczności zastosowanego klasyfikatora LSTM jako najlepszego wyniku z uzyskanych w pierwszym, jak i drugim etapie testów. Jednakże każdy z nich dotyczył zawsze tylko 4 klas uszkodzeń i nie rozpatrywał wszystkich śmigieł jednocześnie. Autorzy wspominają o potrzebie weryfikacji metody na większej liczbie uszkodzeń oraz wskazują na konieczność zwiększenia zasobów obliczeniowych używanego sprzętu, co sugeruje dużą złożoność stosowanych technik przetwarzania sygnałów.

Naukowcy rozwijają także metody sztucznego rozszerzania rozmiaru dostępnych zbiorów pomiarowych w celu efektywnego wytrenowania klasyfikatora opartego na danych. W artykule [135] autorzy zastosowali hybrydowe podejście korzystające z danych rzeczywistych i wirtualnych. Proponowany model LSTM wykorzystuje prędkość obrotową wirników do oszacowania momentu obrotowego oraz ciągu. Natomiast klasyfikator CNN posłużył do wykrycia oraz oceny stopnia uszkodzenia śmigieł czterowirnikowego drona. W eksperymencie wydzielono 16 klas uszkodzeń i uzyskano dokładność klasyfikacji wynoszącą średnio ponad 80% dla różnych testów. Korzystając jedynie z prawdziwych pomiarów, uzyskano wynik na poziomie 76,32%.

Ciekawe podejście do diagnostyki śmigieł przedstawiono w pracy [10]. Autorzy testowali skuteczność wykrywania uszkodzeń w postaci utraty 5% oraz 10% końcówki łopaty śmigła w sześciowirnikowym dronie. Dane z akcelerometru, w który fabrycznie wyposażono BSP, zostały zebrane podczas lotów na otwartej przestrzeni, co w podobnych badaniach zdarza się rzadko. Jako klasyfikatory wykorzystano binarne drzewa decyzyjne oraz dwa typy liniowych SVM. Dla problemu trójklasowego, klasyfikator w postaci drzew decyzyjnych uzyskał dokładność równą 93,37%, natomiast najlepszy klasyfikator SVM – 98,21%. Choć autorzy podają, że proces ekstrakcji cech oraz przeprowadzenia klasyfikacji odbywa się w czasie rzeczywistym, to dla badanego przykładu zajmował on 5,55 ms. Choć w porównaniu z innymi metodami FDI opisywanymi w literaturze jest on bardzo krótki, to – dla zastosowanej częstotliwości próbkowania wynoszącej 350 Hz – nie spełnia definicji czasu rzeczywistego przedstawionej w podrozdziale 1.4 rozprawy.

W innym badaniu [11] wykorzystano sygnały z akcelerometru do detekcji oraz określenia wielkości utraconej końcówki śmigła w ośmiowirnikowym dronie o czterech ramionach. Zastosowano klasyfikator w postaci BGMM (bayesowski model mieszanin rozkładów Gaussa, ang. *Bayesian Gaussian Mixture Model*) do przeprowadzenia detekcji, której zadaniem było wykrycie jedynie faktu istnienia uszkodzenia. Podczas tej binarnej klasyfikacji uzyskano dokładność 97,57%. Jednakże w przypadku rozpatrywania trzech klas, gdzie rozróżniana była utrata końcówki śmigła na długości 15 mm oraz 35 mm, dokładność spadła do 69,7%.

Pojawia się także coraz więcej opracowań dotyczących wykorzystania sygnałów dźwiękowych do przeprowadzania diagnostyki BSP. Artykuł [123] przedstawia badanie 11 scenariuszy uszkodzeń obejmujących kombinacje od 0 do 4 jednocześnie występujących defektów na śmigłach czterowirnikowca. Pojedyncze łopaty śmigieł były pozbawione końcówki na długości 5 mm (2% długości łopaty) lub 10 mm (4%). Testy przeprowadzono w 3 różnych środowiskach: zwykłe pomieszczenie laboratoryjne, komora bezechowa oraz komora pogłosowa. Dron był unieruchomiony na podłożu, a nad nim umieszczono tzw. kamerę akustyczną ze 112 mikrofonami wykonanymi w technologii MEMS. Posłużono się dwoma różnymi klasyfikatorami, które jako dane wejściowe otrzymywały sygnały przetworzone pod kątem częstotliwościowym. Przedstawiono wyniki dla klasyfikatora liniowego oraz w postaci CNN. Wszystkie kombinacje uszkodzeń zostały posegregowane względem rotora, na którym było zamontowane śmigło i pogrupowane w 3 klasy w zależności od rozmiaru uszkodzenia. Nie były rozróżniane osobno przypadki w odniesieniu do miejsca występowania uszkodzenia. Najlepszy model CNN obejmujący 3 klasy osiągnął dokładność 96,86% podczas badań w komorze akustycznej, a obejmujący 2 klasy – 99,77% dla pomieszczenia laboratoryjnego. Test klasyfikatora liniowego dla 2 klas w komorze pogłosowej pokazał jego dokładność na poziomie 94,41%. Autorzy zaznaczyli, że dla 3 klas wynik znacząco spadł. Procedura pomiarowa ogranicza zastosowanie opisywanej techniki jedynie do inspekcji przedstartowej. Wymaga ona specjalistycznego sprzętu i jest złożona obliczeniowo.

Zauważyć można, iż wiele artykułów dotyczących problematyki BSP wskazuje na kłopoty w pozyskiwaniu odpowiedniej ilości danych zebranych podczas prawdziwych lotów. Jeszcze większy deficyt dotyczy lotów pozalaboratoryjnych, w rzeczywistych warunkach terenowych i atmosferycznych. Często badacze posiłkują się danymi sztucznie powiększonymi lub nawet

w pełni wygenerowanymi symulacyjnie. W odpowiedzi na te ograniczenia, stanowiące przeszkodę przede wszystkim na wczesnym etapie prac nad diagnostyką dronów, pojawiają się repozytoria danych zebranych podczas lotów różnymi konstrukcjami BSP. Prawdziwe pomiary dużo lepiej oddają charakterystykę zaburzeń spowodowanych defektami śmigieł niż jakiegokolwiek sztucznie wytworzone dane. Tabela 2.2 podsumowuje istniejące bazy danych utworzone na potrzeby badań nad detekcją uszkodzeń śmigieł wielowirnikowych BSP.

Tab. 2.2: Porównanie baz danych dotyczących detekcji uszkodzeń dronów wielowirnikowych.

Baza danych	Typ BSP	Liczba BSP	Liczba zmiennych	Liczba uszk.	Liczba scenar.	Liczba lotów	Miejsce lotów
Hexacopter Data [46]	6-wirnikowy	1	>9	1	4	20	klatka zewn.
Acoustic Data [19]	4-wirnikowy	1	4	2	21	36	wewn.
UAV-FD [9]	6-wirnikowy	1	38	2	13	18	zewn.
UAV-RFD [65]	4-wirnikowy	1	b.d.	1	5	100	wewn.
CrazyPAD [86]	4-wirnikowy	1	7	12	13	13	wewn.
PADRE [3]	4-wirnikowy	2	32	4	29	29	wewn./zewn.

W bazie audio [19] dotyczącej czteroramiennego drona *Falcon V5* (wypozażonego w osiem śmigieł zamontowanych współosiowo na podwójnych przeciwbieżnych jednostkach napędowych) udostępniono pliki dźwiękowe *WAV* zawierające czterokanałowe dane akustyczne. W przeciwieństwie do baz danych UAV-RFD (baza danych rzeczywistych usterek drona, ang. *UAV-Realistic-Fault-Dataset*) oraz PADRE (repozytorium danych pomiarowych dotyczących uszkodzeń śmigieł, ang. *Propeller Anomaly Data REpository*), w których odczyty dźwiękowe są jednymi z wielu udostępnionych parametrów, zawiera ona wyłącznie sygnały foniczne. Niewątpliwą zaletą bazy danych PADRE jest zamieszczenie pomiarów z dwóch bezzałogowych statków powietrznych, a o kolejne baza będzie sukcesywnie rozbudowywana. Co prawda w repozytorium CrazyPAD (dane anomalii śmigieł drona Crazyflye, ang. *Crazyflye Propeller Anomaly Data*) autorzy również korzystają z dwóch dronów, ale są to dwa identyczne czterowirnikowe *Crazyflye*. Użycie drugiego BSP tego samego typu było podyktowane jedynie potrzebą zapewnienia, iż ewentualne różnice w wydajności powiązane są wyłącznie z badanymi uszkodzeniami i są niezależne od czynników zewnętrznych. Z kolei w repozytorium [46], [9] i [86] zawarto szeroką gamę parametrów dostarczanych przez kontroler lotu. Rozszerza to objętość danych i potencjalnie zwiększa atrakcyjność samej bazy, ale jednocześnie czyni ją mniej czytelną dla użytkownika. Nie każdy parametr lotu jest bowiem przydatny w zadaniach wykrywania usterek drona.

W PADRE wszystkie pomiary są przeznaczone na potrzeby FDI. Repozytorium to zawiera dane zebrane podczas lotów o największej liczbie różnych kombinacji śmigieł sprawnych i niesprawnych, obejmując w sumie 29 scenariuszy z dwóch modeli BSP. Umożliwia wykrywanie usterek, wskazywanie miejsc ich występowania i klasyfikowanie według rozmiaru uszkodzenia. Jest też jedyną bazą danych mieszczącą w sobie zarówno dane z lotów wykonywanych w warunkach laboratoryjnych, jak i realizowanych na otwartej przestrzeni. W rozdziale 5 szczegółowo omówiono strukturę oraz cechy ogólnodostępnego repozytorium PADRE.

Przytoczone badania naukowe w oczywisty sposób nie pokrywają wszystkich stosowanych obecnie rozwiązań diagnostyki śmigieł wielowirnikowych BSP. Wyłania się jednak z nich obraz wskazujący na najczęściej występujące problemy i ograniczenia, z którymi borykają się naukowcy. Wśród nich wyróżnić można brak odpowiedniej ilości dobrej jakości danych pomiarowych z czujników. Są one niezbędne w procesie uczenia maszynowego, którego techniki zdominowały dziedzinę FDI w zakresie usterek dronów. Prace, w których jawnie prezentowane są wyniki testów, najczęściej dotyczą prostych zadań, które nie obejmują szerokiego wachlarza możliwych usterek. Uzyskane rezultaty zwykle odnoszą się do samego faktu wykrycia lub klasyfikacji defektu według jednego z dwóch predefiniowanych rozmiarów uszkodzenia. Na szczęście wśród najnowszych badań coraz częściej spotkać można kompleksowe podejście do diagnozowania stanu śmigieł drona. Badają one coraz bardziej zbliżone do rzeczywistych rodzaje i kombinacje możliwych do wystąpienia anomalii w obrębie układu napędowego. Niektóre z nich osiągają wysoką skuteczność klasyfikacji. Zauważana jest także potrzeba przeprowadzania detekcji online, podczas lotu. Choć część naukowców w swoich publikacjach wskazuje, iż stosowane przez nich metody działają w czasie rzeczywistym, to po dogłębnej analizie okazuje się, że nie jest to do końca spełnione, albo kryterium czasu rzeczywistego nie jest tam zdefiniowane w sposób jednoznaczny.

Niniejsza praca stanowi odpowiedź na wyzwania stawiane nowoczesnej diagnostyce uszkodzeń BSP, a zaprojektowany sprzęt i opracowane algorytmy nie posiadają ograniczeń czy wręcz wad opisywanych w przywołanych źródłach literaturowych.

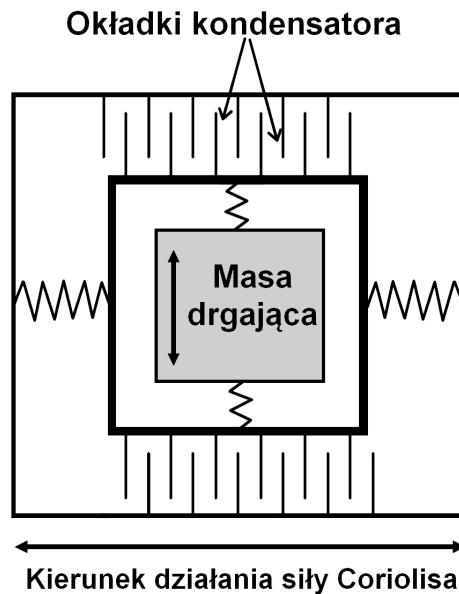
Sygnaly i układy sensoryczne

Uszkodzenia fizyczne elementów odpowiedzialnych za wytwarzanie siły nośnej wywołują zmianę charakterystyki ich pracy. Przerwanie ciągłości śmigła zaburza przepływ powietrza wokół, a utrata części płatu skutkuje brakiem wyważenia całego elementu. Pojawiają się różnice w efektywności działania poszczególnych napędów. Efekty te mogą być wykrywane różnymi typami czujników. Popularnym źródłem pomiarów umożliwiającym wykrycie opisanych nieprawidłowości są przede wszystkim czujniki inercyjne i mikrofony.

3.1 Przyśpieszenia liniowe

Najpowszechniej stosowanym czujnikiem w zadaniach detekcji uszkodzeń jest akcelerometr. Zasada jego działania polega na pomiarze przyśpieszenia liniowego. Jeżeli oś czujnika skierowana jest pionowo, dotyczy to także przyśpieszenia grawitacyjnego. BSP, którego napędy nie pracują w warunkach nominalnych, najczęściej generuje drgania. Nieprawidłowo pracujące śmigło jest przyczyną krótkotrwałych lub ciągłych przechyłów pojazdu latającego (na skutek niezrównoważenia sił ciągu generowanych w poszczególnych jednostkach napędowych na ich elementach wykonawczych, tj. śmigłach). Odpowiednio wyskalowanym akcelerometrem można wykryć chwilowe drgania lub zmierzyć wartość wychylenia płaszczyzny drona od wartości nominalnej. Spotyka się czujniki jedno-, dwu- oraz trójosiowe. Te ostatnie umożliwiają wykrywanie zaburzeń ruchu w każdej płaszczyźnie. Dodatkowo pozwalają na obliczanie kąta przechyłu drona względem płaszczyzny Ziemi.

Obecnie najczęściej stosuje się czujniki inercyjne produkowane w technologii MEMS. Do pomiaru przyśpieszeń wykorzystuje się efekt piezoelektryczny albo – częściej – zmianę pojemności kondensatorów utworzonych wewnątrz struktury czujnika. Centralną część takiego akcelerometru stanowi zawieszona masa z wystającymi płytkami (rysunek 3.1). Razem z podobnymi płytkami umieszczonymi w ramie otaczającej środek czujnika tworzą układ okładek kondensatora. Wraz z nadaniem czujnikowi przyśpieszenia, swobodna masa porusza się względem obudowy sensora, zmieniając odle-

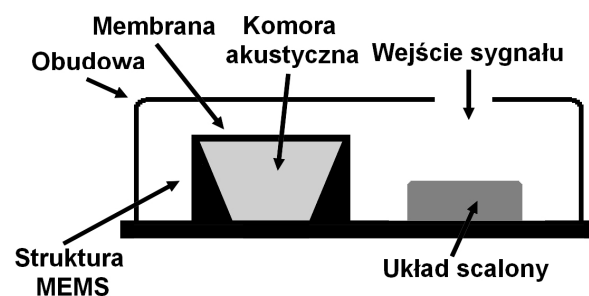


Rys. 3.2: Schemat ideowy jednoosiowego żyroskopu (opracowanie na podstawie [81, 74])

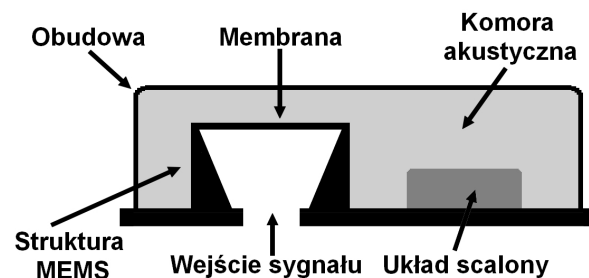
mamy do czynienia ze swobodnie zawieszoną masą oraz płytkami stanowiącymi okładki kondensatorów. Różnica polega na tym, że zawieszona masa jest wprawiona w drgania o określonej częstotliwości i kierunku wychylenia. W czasie przemieszczania czujnika powstaje tzw. efekt Coriolisa. Choć jest to siła pozorna, można ją zmierzyć poprzez pomiar zmian wartości pojemności kondensatorów. Odczytane wartości są w przybliżeniu proporcjonalne do prędkości obrotowej czujnika [121]. Ponieważ dane wyjściowe z żyroskopu obciążone są dryfem, czujnik ten najlepiej sprawdza się przy pomiarach wartości szybko zmieniających się w czasie [75].

3.3 Sygnały akustyczne

Uszkodzenia, o których wspomniano w podrozdziałach 3.1 i 3.2, wywołują także inne efekty. Drgania z konstrukcji drona przenoszone są w powietrzu, emitując sygnały akustyczne. Zaburzenia te można więc mierzyć detektorami dźwięku. Coraz częściej w diagnostyce uszkodzeń stosuje się mikrofony. Dzięki nim możliwy jest pomiar częstotliwości dźwięku (pot. hałasu) wytwarzanego przez silniki i śmigła BSP. Ponieważ defekt silnika lub ostrza śmigła zmienia nieco charakterystykę ich pracy, zmienia się także widmo emitowanej fali akustycznej. Znając rozkład częstotliwości dźwięku podczas normalnej pracy, poprzez znajdowanie różnic w ich widmie możliwe staje się wykrywanie anomalii na podstawie sygnału dźwiękowego.



(a) z otworem dźwiękowym na górze



(b) z otworem dźwiękowym na dole

Rys. 3.3: Budowa mikrofonu MEMS (opracowanie na podstawie [68, 59])

Mikrofon cyfrowy wykonany w technologii MEMS posiada dwa zasadnicze elementy – przetwornik pojemnościowy, w którym funkcję kondensatora o zmieniającej się pod wpływem ciśnienia akustycznego pojemności stanowi para okładek w postaci ruchomej membrany i nieruchomej płytki, a także wyspecjalizowany układ scalony, który te zmiany przekształca na odpowiednie sygnały wyjściowe. Ogólna budowa takiego mikrofonu przedstawiona została na rysunku. 3.3. Otwór wlotowy może znajdować się albo w górnej części obudowy, albo zostać wykonany w technologii PCB (płytki obwodu drukowanego, ang. *Printed Circuit Board*) pod obudową.

Dzięki dużej dostępności tanich czujników inercyjnych oraz mikrofonów omówionych w tym rozdziale, wyposażenie BSP zarówno w redundantne sensory zwiększające niezawodność platformy latającej, jak i rozmieszczenie wielu czujników jednego rodzaju w różnych miejscach drona, nie powinno znacząco zwiększyć kosztu wytworzenia całego systemu. Przyporządkowując niezależny czujnik do każdego z rotorów (lub ramion) wielowirnikowego BSP, możliwe staje się zlokalizowanie potencjalnie wadliwego elementu, a więc przeprowadzenie pełnej procedury FDI. Każdy z przedstawionych typów sensorów może być zastosowany do potwierdzenia tez badawczych postawionych w podrozdziale 1.3.

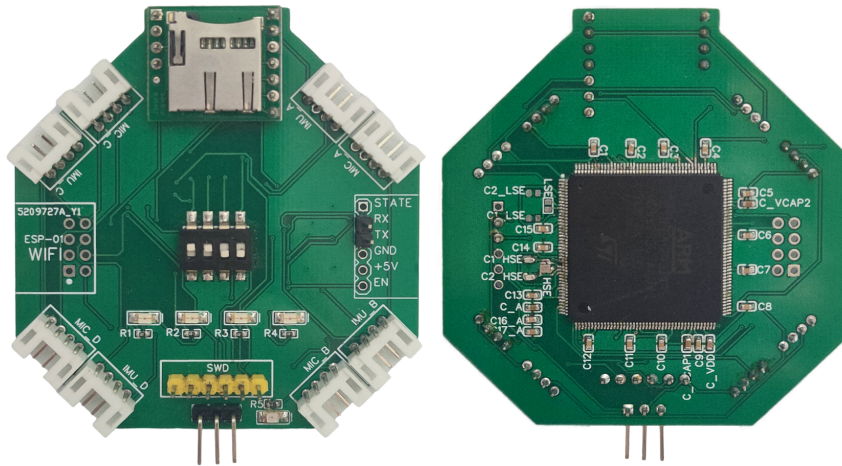
System akwizycji danych sensorycznych

Bazując na założeniach z rozdziału 3, postanowiono zbudować układ umożliwiający wykonywanie pomiarów z opisanych typów czujników. Realizując koncepcję niewielkiego systemu, którego masa i rozmiary nie będą znacząco wpływać na parametry lotu nawet małego BSP, zaprojektowano platformę sprzętowo-programową. Jej zadaniem jest pobieranie i trwały zapis odczytów sensorycznych, które można wykorzystać w zadaniach diagnostyki śmigieł wielowirnikowych dronów.

4.1 Budowa układu pomiarowego

Komercyjne bezzałogowe pojazdy latające zawierają zwykle zestaw różnych czujników rozmieszczonych w okolicy środkowej części kadłuba. Dane z sensorów nie zawsze są w pełni dostępne dla użytkownika sprzętu. Często cechuje je również niska częstotliwość próbkowania odczytów. Popularne modele BSP mają ograniczony potencjał odczytu i zapisu parametrów lotu. Nawet jeżeli dron posiada takie możliwości, stosowane standardy nie są jednolite. Spotyka się także ograniczenia parametrów odczytywanych danych, które są celowo nałożone przez producenta sprzętu lub oprogramowania. Z tego powodu powstała koncepcja opracowania własnego, niezależnego od systemów BSP układu zdolnego do akwizycji danych z czujników [104]. W badaniach nad detekcją uszkodzeń bezzałogowych statków powietrznych postanowiono skupić się na metodach opartych na danych. Wymagają one dużej liczby pomiarów, na podstawie których można przeprowadzać wnioskowanie dotyczące występowania usterek. Podczas gdy jedna centralna inercyjna jednostka pomiarowa o niskiej częstotliwości próbkowania jest zwykle wystarczająca do określenia orientacji drona, nie pozwala na skuteczne wykrywanie usterek w jego jednostkach napędowych. Aby określić lokalizację i rodzaj usterki w silniku lub śmigle, czujniki muszą być umieszczone blisko nich. Aby sprostać tym wymaganiom, zaprojektowano niewielki układ elektroniczny (rysunek 4.1), którego pracą steruje mikrokontroler. Wybrano model STM32H743IIT6, którego potencjał postanowiono wykorzystać do wydajnej akwizycji, wstępnego przetwarzania i zapisu pomiarów z czujników

zamontowanych na ramionach BSP. Jego najważniejsze parametry przedstawia tabela 4.1. Wymiary układu bez uwzględnienia podłączanych czujników i akumulatora to w przybliżeniu 66 mm szerokości oraz 66 mm długości, a masa wynosi ok. 15 g.



Rys. 4.1: Widok górnej i dolnej warstwy systemu do akwizycji danych z czujników

Tab. 4.1: Parametry mikrokontrolera STM32H743IIT6 [129].

Parametr	Wartość
Architektura	ARM Cortex-M7
Taktowanie rdzenia	480 MHz
Pamięć flash	2 MiB ¹
Pamięć RAM	1 MiB
Obudowa	LQFP176
Napięcie zasilania	1.62–3.6 V

Z układu wyprowadzono cztery złącza I²C (zintegrowana magistrala szeregową, ang. *Inter-Integrated Circuit bus*), każde połączone z niezależnym interfejsem w mikrokontrolerze, a także cztery złącza SAI (interfejs szeregowy audio, ang. *Serial Audio Interface*) [130]. Konektory I²C służą do podłączania czujników wyposażonych w taki interfejs, przede wszystkim czujników inercyjnych, jak akcelerometry i żyroskopy. Magistrala SAI pozwala z kolei podłączyć do układu cyfrowe mikrofony MEMS z wyjściem PDM (modulacja gęstością impulsów, ang. *Pulse-Density Modulation*) [126]. Na płycie PCB umieszczony został moduł μ SD (karta pamięci mikro-SD, ang. *micro Secure Digital*) podłączony za pomocą 4-bitowego interfejsu SDIO (interfejs wejścia-wyjścia kart pamięci SD, ang. *Secure Digital Input Output*) [131].

¹W rozprawie, zgodnie z zaleceniami IEC (Międzynarodowa Komisja Elektrotechniczna, ang. *International Electrotechnical Commission*), do określania pojemności pamięci stosuje się przedrostki dwójkowe, w których 1 KiB = 2¹⁰ B, 1 MiB = 2²⁰ B itd.

Służy on do zapisywania pomiarów pobranych z czujników. Dodatkowo układ wyposażono w prosty interfejs użytkownika w postaci mikroprzełączników oraz diod elektroluminescencyjnych (LEDs). Wyprowadzone zostały także złącza zasilania i złącze programatora SWD (szeregowy interfejs programowania, ang. *Serial Wire Debug*) [128]. Na płytce umieszczono również konektory do podłączenia modułów WiFi oraz Bluetooth, jednak – z uwagi na opóźnienia i ograniczenia przepustowości transmisji bezprzewodowej – w opisywanych w niniejszej rozprawie badaniach nie zostały one wykorzystywane. Zamiast transmisji bezprzewodowej odczytów, zdecydowano się na bardziej niezawodny i szybszy proces zapisywania wszystkich pomiarów na kartę pamięci.

Zasadniczym zadaniem opisywanego układu jest pobieranie i trwałe zapisywanie danych z czujników podłączonych do systemu w czasie lotu BSP. Jako punkt wyjścia w wyborze liczby wyprowadzeń czujników, z uwagi na powszechne zastosowanie, posłużył dron czterowirnikowy. Z tego powodu na PCB umieszczono cztery wyprowadzenia I²C. Użyty mikrokontroler STM32H743II także posiada 4 niezależne interfejsy I²C, dzięki czemu możliwa jest szybka akwizycja danych z każdego z czterech podłączonych czujników inercyjnych. Oczywiście ten typ interfejsu umożliwia podłączanie wielu czujników do jednej magistrali, ale wiąże się to z opóźnieniami odczytu. Niemniej, poprzez zastosowanie odpowiedniego adresowania, istnieje możliwość zwiększenia liczby sensorów obsługiwanych przez układ. Choć na etapie projektowania rozważano obsługę głównie akcelerometrów i żyroskopów, praktycznie każdy czujnik wyposażony w interfejs I²C i zasilany napięciem dostępnym w układzie może zostać wykorzystany.

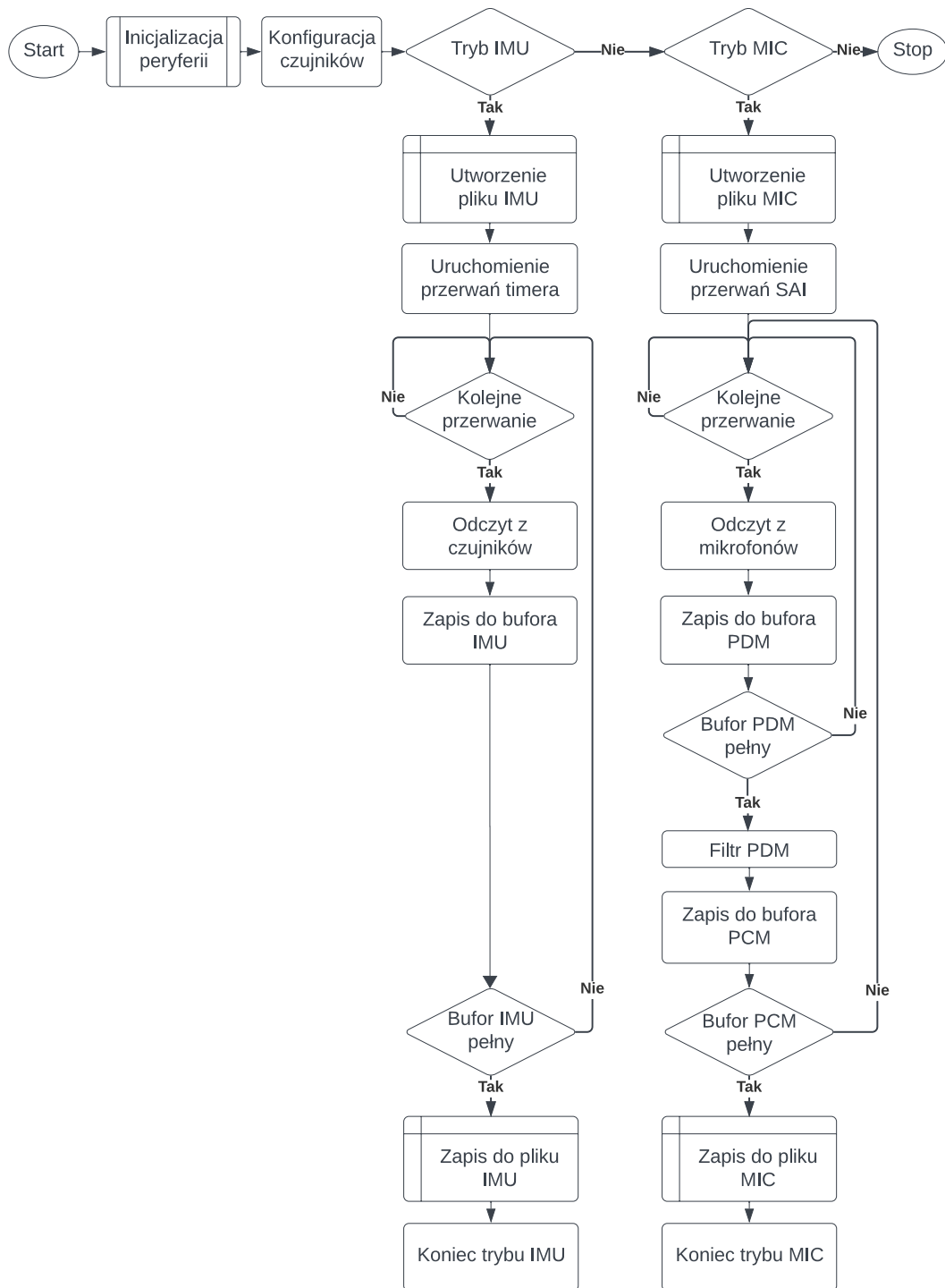
Analogicznie układ wyposażono w 4 złącza mikrofonowe obsługiwane przez interfejs SAI. Umożliwiają one podłączenie mikrofonów cyfrowych kodujących sygnały w standardzie PDM.

Zasilanie mikrokontrolera oraz wszystkich układów peryferyjnych także zrealizowano niezależnie od używanej jednostki latającej. Wykorzystano akumulator Li-Po o napięciu 3,7 V i pojemności 250 mAh oraz stabilizator utrzymujący stałe napięcie, którym zasilany jest mikrokontroler, moduł karty pamięci i czujniki.

4.2 Działanie systemu akwizycji

Uproszczone działanie programu do akwizycji danych z czujników inercyjnych oraz mikrofonów przedstawia rysunek 4.2. Po inicjalizacji niezbędnych funkcji systemowych następuje sprawdzenie ustawienia wszystkich dźwigni poczwórnego mikroprzełącznika. Wskazuje on, które śmigło lub śmigła są uszkodzone w czasie danego lotu. Funkcja ta używana jest jedynie na etapie akwizycji danych w celu przypisania odpowiedniej etykiety (oraz nadania odpowiedniej nazwy plikom z pomiarami) dla każdego z rozpatrywanych scenariuszy uszkodzeń. Po zmianie konfiguracji śmigieł, ustawienie przełącznika należy zmienić na odpowiadające aktualnej kondycji każdej z jednostek napędowych. Zostają uruchomione funkcje odpowiedzialne za ustawienie zakresu działania i czułości sensorów oraz sprawdzenie liczby i rodzajów podłączonych czujników. Na bazie tych ustawień, na karcie pamięci tworzony jest plik CSV dla danych z akcelerometrów, żyroskopów i barometrów oraz osobny plik CSV dla danych z mikrofonów. W nazwach zawarto także inne przydatne informacje. Przykładowo plik o nazwie *Solo_IMU_16g_2kdps_16bit_0001.csv* oznacza, że akwizycji dokonano na dronie *3DR Solo*, zakres pomiarowy akcelerometru wynosi ± 16 g, zakres pomiarowy żyroskopu to ± 2000 dps, czułość barometru ustawiono na 16 bitów, a śmigło *D* jest uszkodzone. Dla danych akustycznych przykładowa nazwa pliku *Solo_MIC_48kHz_0010.csv* wskazuje szybkość akwizycji dźwięku na 48000 próbek na sekundę, a niesprawne śmigło znajduje się na ramieniu oznaczonym literą *C* multirotora *3DR Solo*.

W pierwszej kolejności aktywowany jest tzw. *tryb IMU*, który odpowiada za akwizycję danych z czujników podłączonych do interfejsu I²C. Po utworzeniu oraz otwarciu pliku CSV uruchamiane są przerwania timera. Odpowiadają one za regularne odstępy czasowe kolejnych pomiarów. Na przykład, jeżeli wybrana jest akwizycja przeprowadzana 500 razy na sekundę, przerwania są uruchamiane w odstępie 2 ms. Po wystąpieniu każdego przerwania dokonywany jest odczyt danych z rejestrów pomiarowych wszystkich używanych czujników I²C. Proces powtarzany jest do momentu zapełniania wcześniej zdefiniowanego rozmiaru bufora, który ustalono na 5849088 bajtów, czyli 86016 niezależnych pomiarów, każdy po 68 wartości typu *uint8_t*. Rejestry czujników inercyjnych odpowiedzialne za przechowywanie danych pomiarowych są 8-bitowe. Akwizycji podlegały dane dla trzech osi akcelerometru oraz trzech osi żyroskopu, każda zajmująca 2 rejestry, co daje w sumie słowo o długości 12 bajtów. Jednakże pomiędzy tymi danymi umieszczono dwa rejestry przechowujące wartość temperatury. W celu zoptymalizowania po-



Rys. 4.2: Tryb akwizycji (opracowanie na podstawie [103])

miarów pod kątem szybkości działania, także te rejestry były odczytywane, aby za jednym pomiarem pobrać wszystkie wartości podlegające akwizycji. W dalszym przetwarzaniu dane z termometru nie brały udziału, ponieważ nie są one skorelowane ze stanem śmigieł. Uwzględnione zostały tylko dane dotyczące przyspieszeń liniowych i prędkości kątowych. Zatem odczyty z każdego akcelerometru i żyroskopu (wraz z termometrem) zajmują 14 bajtów. Trzy kolejne bajty przeznaczone są dla pomiarów z barometru. Dla 4 modułów GY-91 daje to w sumie $4 \cdot 17 = 68$ bajtów. Przy $f_s = 500$ Hz, 86016 pomiarów odpowiada 172,032 s lotu. Dla uproszczenia, w związku z późniejszą formą, w jakiej te dane zostały upublicznione, pomiary ze wszystkich rejestrów dokonywane w danym przerwaniu timera określane będą mianem „linii danych” lub po prostu „wierszem”. Zatem z każdego lotu rejestrowano 86016 linii danych pomiarowych.

W rzeczywistości dane te są spójne dla każdych kolejnych 4096 linii, licząc od pierwszej. W związku z ograniczeniami pamięci mikrokontrolera, cały lot jest podzielony na 21 etapów trwających 8,192 sekund. Na koniec każdego etapu pobrane wartości są dopisywane do pliku CSV. Tak więc pomiary z linii 1–4096, 4097–8192 itp. są odczytywane co 2 ms, ale pomiary z linii 4096 i 4097 dzieli większy odstęp, w czasie którego dokonuje się utrwalenia danych na karcie pamięci. W związku z powyższym, rozpatrując te dane – szczególnie w dziedzinie częstotliwości – zaleca się korzystanie z „paczek” nie większych niż 4096 kolejnych linii. Jednakże w praktyce tak duży zbiór okazuje się nadmiarowy, zatem przedstawiony sposób podziału danych nie stanowi żadnego ograniczenia. Po zapisaniu na karcie pamięci ostatniej części pomiarów, plik z danymi zostaje zamknięty, zatrzymywane są przerwanie timera, a *tryb IMU* zakończony.

Następnie program przechodzi do działania w *trybie MIC*. Odpowiada on za akwizycję danych z mikrofonów. Ogólny przepływ danych jest analogiczny do *trybu IMU*. Otwierany jest nowy plik CSV przeznaczony dla sygnałów akustycznych oraz uruchamiana jest funkcja odpowiedzialna za odczyt danych poprzez interfejs SAI. Do utworzonego bufora pamięci zapisywane są kolejne próbki dźwięku w formacie PDM ze wszystkich mikrofonów. Po wypełnieniu bufora PDM uruchamiane są procedury konwersji danych do postaci PCM (modulacja impulsowo-kodowa, ang. *Pulse-Code Modulation*) [13, 125]. Sygnały takie trafiają na kartę pamięci w formacie *uint16_t*. Bufor mieści 4096 niezależnych pomiarów z każdego śmigła, więc w przypadku mikrofonów ciągłość danych jest zapewniona – podobnie jak przy danych

inercyjnych – dla każdego kolejnych 4096 pomiarów. W związku z dużo większą szybkością akwizycji pomiarów z mikrofonów (48 kHz) w porównaniu z czujnikami inercyjnymi (500 Hz), objętość danych akustycznych również jest znacząco większa. Z tego względu czas działania trybu MIC zmniejszono do 43,008 s. W tym przypadku potrzeba aż 504 cykli odczytu i zapisu, pokrywających odcinki ok. 85 ms spójnych danych.

4.3 Rozwój i modyfikacje systemu

Wraz z użytkowaniem omawianego w tym rozdziale systemu pojawiały się pomysły dotyczące kolejnych usprawnień. Najważniejsze z nich obejmują modyfikacje mające na celu zapewnienie niezawodności oraz zwiększenie szybkości działania procesu akwizycji danych.

Podczas wielogodzinnych eksperymentów zdarzało się, że akumulator zasilający system akwizycji ulegał rozładowaniu podczas lotu. W takim przypadku, jeżeli nie zdążył wykonać się zapis danych na kartę pamięci, pomiary z całego lotu były tracone. Wówczas całą procedurę trwającą ok. trzech minut należało powtórzyć. Aby zmniejszyć tę uciążliwość, zastosowano zabieg polegający na zapisywaniu na kartę fragmentarycznych danych. Lot podzielony został na jednominutowe odcinki, po których następowało utrwalenie pomiarów. Dzięki temu ewentualne zaniki zasilania powodowały utratę jedynie części danych, które były rejestrowane w czasie maksymalnie jednej minuty.

Ważniejszym usprawnieniem, wpływającym na poprawę szybkości procesu zapisywania, była rezygnacja z konwersji danych na postać szesnastkową. Okazało się, że jest to jedna z najbardziej czasochłonnych procedur biorących udział w przetwarzaniu sygnałów. Zrezygnowano więc całkowicie z tego przekształcenia i na kartę zapisywano surowe dane w postaci binarnej. Dodatkowa oszczędność czasu wynikała z dwukrotnie mniejszej objętości danych tego typu w porównaniu z danymi heksadecymalnymi zapisywanymi przy użyciu kodu ASCII [29].

Podczas pierwszych prób zapisu w czasie jednego lotu zarówno danych z czujnika inercyjnego, jak i mikrofonów, okazało się, że dostępna pamięć RAM (pamięć o dostępie swobodnym, ang. *Random-Access Memory*) oferowana przez mikrokontroler jest mocno ograniczona. Można było albo dostępne zasoby rozdzielić pomiędzy używane typy czujników, albo prze-

```
union IMU_MIC {
    uint8_t IMU_bufor [liczba_odczytow][rozmiar_odczytow];
    uint32_t MIC_bufor [rozmiar_bufora];
} sensor;
```

prowadzać osobne loty dla różnych typów akwizycji. Opisywany problem postanowiono rozwiązać, korzystając z typu danych *unia*.

Zabieg ten pozwolił na wykorzystanie tego samego obszaru pamięci, raz na dane typu *uint8_t* przeznaczone na pomiary inercyjne, a raz na dane typu *uint32_t*, których wymagały sygnały dźwiękowe. W ten sposób podczas jednego lotu można było zapisywać różne typy danych bez potrzeby wgrywania innego programu. Niemożliwa jednak okazała się symultaniczna akwizycja wszystkich sygnałów. Zatem podprogram odpowiedzialny za pomiary mikrofonowe rozpoczął pracę po zakończeniu działania podprogramu obsługującego jednostki IMU (rysunek 4.2). Pomimo tych ograniczeń, zastosowanie *unii* przyniosło dużą oszczędność czasu.

Pozostałe modyfikacje obejmowały przede wszystkim drobne usprawnienia na etapie procedur przygotowawczych wykonywanych przed wzniesieniem drona. Dotyczyły m.in. przeprowadzenia testu komunikacji ze wszystkimi czujnikami. Zaprogramowano układ do wysyłania komunikatów poprzez protokół UART (uniwersalny asynchroniczny nadajnik-odbiornik, ang. *Universal Asynchronous Receiver-Transmitter*) oraz prostych kodów świetlnych z wykorzystaniem zamontowanych diod. Pozwalały one na identyfikację np. błędnie podłączonego czujnika lub ustawienia niewłaściwego zakresu pomiarowego. Informowały także na bieżąco o postępie i poprawności działania programu. Inne usprawnienia, w tym także te wymagające ingerencji sprzętowej, będą implementowane w następnej wersji systemu, nad którym trwają obecnie prace.

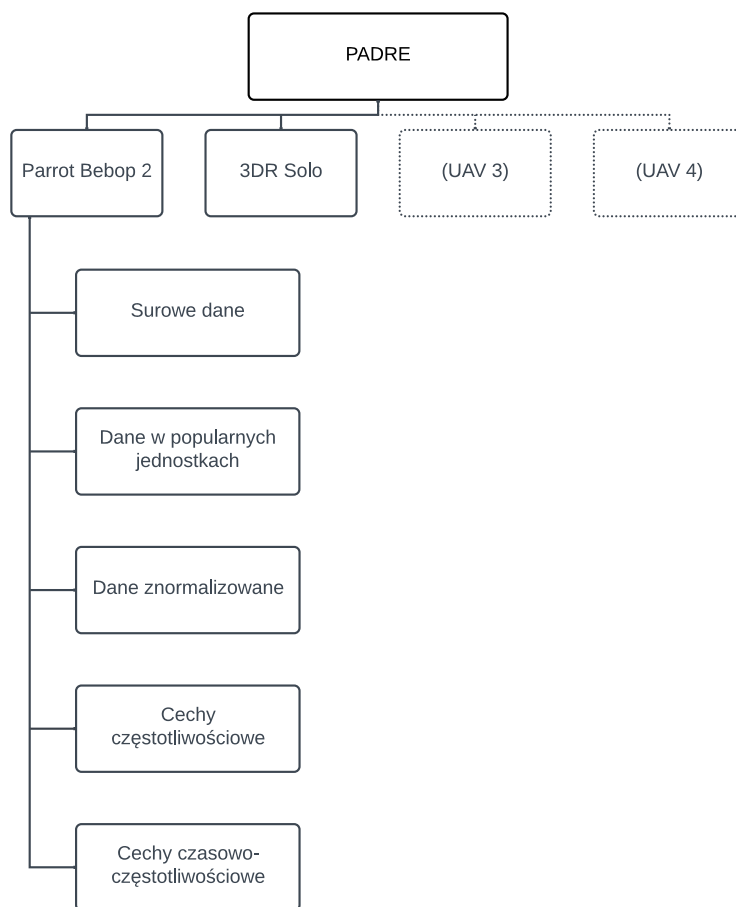
Repozytorium danych sensorycznych

W rozdziale 2 wskazano problemy, z jakimi borykają się naukowcy we wstępnym stadium prac badawczych dotyczących diagnozowania stanu BSP. Metody FDI oparte na wiedzy wymagają dużej ilości danych wysokiej jakości. Ponieważ ich pozyskiwanie i wstępna obróbka są zazwyczaj problematyczne i czasochłonne, wygodnym rozwiązaniem jest korzystanie z gotowych do użycia danych. W początkowej fazie badań nad FDI wykorzystywano dane z drona *Falcon V5* pozyskane na potrzeby badań [15]. Pozwoliło to na szybkie przetestowanie nowych koncepcji bez znużającego zbierania pomiarów o odpowiedniej objętości. Ponieważ taki sposób rozpoczęcia własnych eksperymentów okazał się skuteczny, postanowiono rozwinąć to rozwiązanie. Z uwagi na fakt, iż na kolejnych etapach badań pracowano z danymi zebranymi samodzielnie, zrodził się pomysł, by dane te upublicznić, pomagając innym badaczom. W ten sposób powstała ogólnodostępna baza danych sensorycznych **PADRE** (Propeller Anomaly Data Repository).

Dane pomiarowe zostały zebrane i udostępnione jako repozytorium w serwisie GitHub [3]. Utworzono projekt o nazwie `UAV_measurement_data`, z plikami podzielonymi na kategorie w strukturze hierarchicznej. Udostępnione zostały wyłącznie pomiary z czujników, w które wyposażony był niezależny system akwizycji danych. Nie rejestrowano sygnałów sterujących ani innych danych.

Repozytorium zawiera dane zebrane podczas lotów dronem *Parrot Bebop 2* w *Laboratorium bezzałogowych statków powietrznych AeroLAB* w Instytucie Robotyki i Inteligencji Maszynowej Politechniki Poznańskiej [2] oraz w czasie lotów dronem *3DR Solo* przeprowadzonych w przestrzeni miasta Sydney (w ramach projektu NAWA STER) [4]. Oba pojazdy są czterowirnikowymi dronami w konfiguracji *quad X*. Oczywiście są to tylko przykłady i nie wyczerpują zakresu zastosowań opracowanego systemu akwizycji omówionego w rozdziale 4. Możliwe jest również wykorzystanie systemu do akwizycji danych z dronów o innej liczbie śmigieł. Schematyczną strukturę bazy danych PADRE przedstawia rysunek 5.1. W czasie pisania tego rozdziału w repozyto-

rium upubliczniono dane zebrane z dwóch dronów, jednak planowane jest systematyczne rozbudowywanie kolejnych katalogów.



Rys. 5.1: Repozytorium PADRE (opracowanie na podstawie [103])

5.1 Dane z drona *Parrot Bebop 2*

Pierwszą platformą, na której zamontowano przedstawiony wcześniej system akwizycji, był dron *Parrot Bebop 2*, którego podstawowe parametry zebrano w tabeli 5.1. Konstrukcja BSP pozwala na umieszczenie systemu – z wykorzystaniem uchwyty wykonanego metodą druku 3D – na górze kadłuba. Umożliwia to wygodne poprowadzenie przewodów wzdłuż ramion drona i zamocowanie czujników bezpośrednio pod śmigłami. Wybrano cztery moduły GY-6500, które wyposażone są w 3-osiowe akcelerometry i 3-osiowe żyroskopy. Ich najważniejsze parametry przedstawia tabela 5.2. Wygląd całego zestawu pomiarowego, z użyciem którego wykonywano loty, zamieszczono na rysunku 5.2.

Tab. 5.1: Wybrane parametry drona *Parrot Bebop 2* [95].

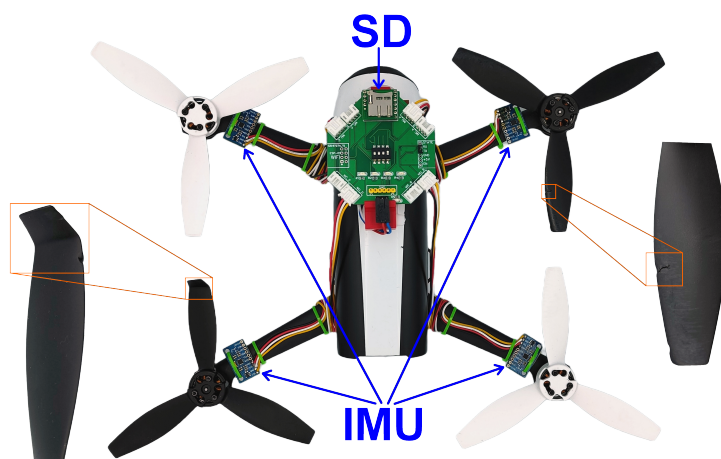
Masa	500 g
Wymiary	328 x 328 x 89 mm
Rozmiar śmigła	15.2 cm
Pojemność baterii	2700 mAh
Czas lotu	do 25 minut
Maksymalna prędkość lotu poziomego	16 m/s
Maksymalna prędkość lotu pionowego	6 m/s
Zasięg łączności radiowej	300 m

Tab. 5.2: Parametry modułu inercyjnego GY-6500 [61].

Układ scalony	MPU-6500
Interfejsy komunikacyjne	I ² C, SPI
Napięcie zasilania	1.17–3.45 V
Akcelerometr	
Rozmiar danych	16 bitów
Liczba osi	3
Zakres	±2, ±4, ±8, ±16 g
Czułość	2048–16384 LSB/g
Szybkość transmisji danych	4000 Hz
Żyroskop	
Rozmiar danych	16 bitów
Liczba osi	3
Zakres	±250, ±500, ±1000, ±2000 dps
Czułość	16.4–131 LSB/dps
Szybkość transmisji danych	8000 Hz

Dane zostały zebrane podczas lotów z różnymi kombinacjami śmigieł sprawnych oraz uszkodzonych. Zastosowano 4-cyfrowe kodowanie: pierwsza cyfra odpowiada śmigłu A, druga cyfra – śmigłu B, trzecia cyfra – śmigłu C, a czwarta cyfra – śmigłu D. Cyfra na każdej pozycji oznacza konkretny typ usterki. Badaniu poddano dwa popularne typy uszkodzeń: 1 – wyszczerbiona krawędź śmigła oraz 2 – wygięta końcówka śmigła (rysunek 5.2). Sprawne śmigło oznaczane jest zawsze cyfrą 0. Wykonano 20 lotów z następującymi klasami uszkodzeń:

- wszystkie śmigła sprawne: 0000,
- pojedyncze uszk.: 1000, 0100, 0010, 0001, 2000, 0200, 0020, 0002,
- podwójne uszk.: 1100, 1020, 1002, 0120, 0102, 0022,
- potrójne uszk.: 1120, 1102, 1022, 0122,
- poczwórne uszk.: 1122.



Rys. 5.2: Parrot Bebop 2 z zamontowanym na kadłubie systemem akwizycji [103]

Wszystkie loty wykonano w laboratorium AeroLAB w trybie manualnym. Podczas każdej próby (lotu trwającego ok. trzech minut), dron wykonywał m.in. następujące manewry: wznoszenie, opadanie, lot w zawisie, do przodu, do tyłu, w prawo, w lewo, po skosie, obrót zgodnie i przeciwnie do ruchu wskazówek zegara, lot po ósemce oraz wzdłuż spirali wznoszącej i opadającej. Manewry przeprowadzano z różną prędkością oraz dynamiką zmian kierunku.

Akwizycji z szybkością 500 pomiarów na sekundę podlegały dane obejmujące wszystkie osie akcelerometrów i żyroskopów pracujących w zakresach odpowiednio ± 16 g oraz ± 1000 dps. Pomiaru uruchamiane były w momencie, gdy dron znajdował się już w zawisie i kończyły się przed rozpoczęciem procedury lądowania.

Z każdego lotu na karcie pamięci zapisywany był plik CSV obejmujący 86016 wierszy. Każdy z wierszy zawiera 96 znaków reprezentowanych w kodzie ASCII, które są wartościami rejestrów pomiarowych zapisanymi w postaci heksadecymalnej. Każdy bajt danych zajmuje dwa symbole *HEX*. Pomiaru z poszczególnych osi czujników zajmują dwa bajty danych: *H* – najbardziej znaczący bajt, *L* – najmniej znaczący bajt. Wartości w każdym wierszu są ułożone w następującej kolejności:

A_ax_H, A_ax_L, A_ay_H, A_ay_L, A_az_H, A_az_L,
 A_gx_H, A_gx_L, A_gy_H, A_gy_L, A_gz_H, A_gz_L,
 B_ax_H, B_ax_L, B_ay_H, B_ay_L, B_az_H, B_az_L,
 B_gx_H, B_gx_L, B_gy_H, B_gy_L, B_gz_H, B_gz_L,
 C_ax_H, C_ax_L, C_ay_H, C_ay_L, C_az_H, C_az_L,
 C_gx_H, C_gx_L, C_gy_H, C_gy_L, C_gz_H, C_gz_L,

D_ax_H, D_ax_L, D_ay_H, D_ay_L, D_az_H, D_az_L,
D_gx_H, D_gx_L, D_gy_H, D_gy_L, D_gz_H, D_gz_L.

Pierwsza litera oznacza konkretne śmigło, mała litera *a* to dane z akcelero-
metru, *g* to dane z żyroskopu, natomiast *x*, *y* i *z* to poszczególne osie czujnika
w lokalnym układzie odniesienia drona.

Tak przygotowanych 20 plików, odpowiadających 20 scenariuszom uszko-
dzeń śmigieł, stanowi zbiór surowych danych w repozytorium PADRE. Jest
to najważniejszy z punktu widzenia użytkownika katalog. Wszystkie pozos-
tałe foldery zawierają dane będące przetworzonymi sygnałami z katalogu
Raw_data.

W folderze Range_data zamieszczono pliki, w których surowe dane prze-
kształcono w taki sposób, by zostały wyskalowane następująco: pomiary
z akcelometru w jednostkach przyspieszenia ziemskiego *g*, a pomiary z ży-
roskopu w stopniach na sekundę dps. Tego typu dane najlepiej nadają się
do analizy mierzonych parametrów¹. Szczególnie przydatne są także do spor-
ządzania wykresów czasowych danych pomiarowych z czujników. Zasad-
nicza różnica struktury tych plików, w porównaniu do plików z katalogu
Raw_data, polega na dwukrotnym zmniejszeniu ilości niezależnych danych,
co spowodowane jest przeprowadzoną konwersją, która dwa bajty HEX spro-
wadza do jednej liczby zmiennoprzecinkowej ze znakiem:

A_ax, A_ay, A_az, A_gx, A_gy, A_gz,
B_ax, B_ay, B_az, B_gx, B_gy, B_gz,
C_ax, C_ay, C_az, C_gx, C_gy, C_gz,
D_ax, D_ay, D_az, D_gx, D_gy, D_gz.

Katalog Normalized_data składa się z plików, których struktura jest
analogiczna do poprzednio omówionych, ale zawiera dane zmiennoprzecin-
kowe o wartościach od -1 do $+1$. Pomiary zostały znormalizowane w taki
sposób, że maksymalne mierzone przyspieszenie liniowe o wartości -16 g
odpowiada liczbie -1 , z kolei $+16$ g to $+1$. W przypadku żyroskopu skrajne
wartości dotyczą prędkości kątowych wynoszących odpowiednio -1000 dps
oraz $+1000$ dps. Tak przygotowane pliki są przydatne w technikach uczenia

¹Pomiary te, z uwagi na zawarte w swojej strukturze łatwe do interpretacji wartości przy-
śpieszeń i prędkości, dają bezpośredni wgląd w występujące amplitudy tych wielkości.
Z tego względu najlepiej nadają się do analizy przyspieszeń liniowych i prędkości kąt-
owych oraz intensywności drgań w poszczególnych osiach konkretnego ramienia drona.

maszynowego, szczególnie w zastosowaniu sztucznych sieci neuronowych, które jako dane wejściowe przyjmują znormalizowane wartości [22].

Najbardziej złożony pod względem liczby plików jest katalog `FFT_data`. Zawiera on – w przeciwieństwie do dotychczas opisanych folderów – dane w dziedzinie częstotliwości. W kolejnych podfolderach zamieszczono pomiary przetworzone przy użyciu algorytmu FFT zgodnie ze wzorem (5.1):

$$X(k) = \frac{2}{N} \left| \sum_{n=1}^N x(n) \cdot \exp \left(-2\pi j(k-1) \frac{(n-1)}{N} \right) \right|, \quad \text{dla } 1 \leq k \leq N, \quad (5.1)$$

gdzie k to numer prążka odpowiadający określonej częstotliwości, N to liczba przetwarzanych próbek w oknie pomiarowym, a n to numer próbki w sygnale dziedziny czasu.

W rzeczywistości przez współczynnik równy 2 mnożone są wartości dla wszystkich prążków z wyjątkiem pierwszego, który odpowiada częstotliwości 0 Hz, oraz ostatniego rzeczywistego, który jest równy częstotliwości Nyquista [87]. Numeracja kolejnych prążków zaczyna się od 1, co odpowiada sposobowi indeksowania w środowisku *MATLAB*.

W celu umożliwienia późniejszego testowania zebranych danych na potrzeby znalezienia optymalnych (pod kątem dokładności i szybkości klasyfikacji) ustawień, wykorzystano rozmaite parametry procesu przetwarzania. W szczególności stosowano różne długości okna pomiarowego (co skutkowało innymi rozdzielczościami sygnału w dziedzinie częstotliwości) oraz różne funkcje okna. Wycinano także różne fragmenty z całego spektrum uzyskanych częstotliwości, odrzucając pozostałe wartości. Przykładowa nazwa podfolderu `128_Hann_16_36` oznacza, że długość okna wynosiła 128 próbek, zastosowano okno Hann, a pliki zawierają dane ze środkowego spektrum częstotliwości od prążka nr 16 do 36. Poszczególne próbki sygnału wymnażane są przez funkcję okna Hann zgodnie ze wzorem (5.2):

$$x_w(n) = x(n) \cdot 0,5 \cdot \left(1 - \cos \left(\frac{2\pi n}{N-1} \right) \right). \quad (5.2)$$

Wartości zgodne z dolną i górną częstotliwością odcięcia są wyliczane ze wzorów (5.3) oraz (5.4):

$$f_D = \frac{f_s}{N}(D-1), \quad (5.3)$$

$$f_G = \frac{f_s}{N}(G - 1), \quad (5.4)$$

gdzie D oznacza liczbę całkowitą odpowiadającą numerowi dolnego prążka, a G – górnego.

Liczba wszystkich prążków w przypadku omawianego podfolderu wynosi $128/2 + 1 = 65$, natomiast w związku z zastosowaną $f_s = 500$ Hz, częstotliwości zawierają się w przedziale od ok. 58,6 Hz ($500 \cdot (16 - 1)/128$) do ok. 136,7 Hz ($500 \cdot (36 - 1)/128$). W plikach CSV takiego podfolderu (które zostały rozdzielone pod kątem poszczególnych czujników, czyli są osobne pliki dla akcelerometru i osobne dla żyroskopu), w każdym wierszu znajduje się $21 \cdot 4 \cdot 3 = 252$ wartości. Poszczególne wielkości w równaniu odpowiadają kolejno liczbie wykorzystanych prążków częstotliwościowych ($36 - 16 + 1$), liczbie czujników oraz liczbie osi każdego czujnika. Z kolei liczba wierszy wynosi $86016/(128 \cdot 2) + 1 = 337$. Dla danych z akcelerometrów w każdym wierszu znajdują się pomiary uporządkowane w następujący sposób:

A_ax16 ... A_ax36, A_ay16 ... A_ay36, A_az16 ... A_az36,
 B_ax16 ... B_ax36, B_ay16 ... B_ay36, B_az16 ... B_az36,
 C_ax16 ... C_ax36, C_ay16 ... C_ay36, C_az16 ... C_az36,
 D_ax16 ... D_ax36, D_ay16 ... D_ay36, D_az16 ... D_az36.

W miejscu wielokropka znajdują się wartości odpowiadające kolejnym prążkom o indeksach od 17 do 35. Są to dodatnie liczby zmiennoprzecinkowe, które reprezentują moc sygnału dla konkretnych częstotliwości. Ponieważ ich wartości nie przekraczają 1, nie było potrzeby przeprowadzania ich jawnej normalizacji.

Katalog o nazwie DWT_data jest zbiorem plików, które są wynikiem przekształcenia DWT (dyskretna transformacja falkowa, ang. *Discrete Wavelet Transform*). Pozwala ono na uzyskanie informacji nie tylko o zawartości widmowej badanego sygnału, ale także o charakterze jego zmian w dziedzinie czasu. Pliki podzielono pod względem używanego czujnika oraz klasy uszkodzenia. 264 wartości w każdym wierszu to część uproszczona (*approx*) oraz szczegółowa (*det*) dla każdej osi czujnika ($((11 + 11) \cdot 3 \cdot 4)$). Przykładowy plik o nazwie *Bebop_DWT_ACCEL_XYZ_16g_16_db4_1_22_0000.csv* zawiera dane czasowo-częstotliwościowe ze wszystkich osi akcelerometrów o zakresie ± 16 g, a transformację DWT z zastosowaniem falek *db4* przeprowadzono na danych o 16-elementowej szerokości okna pomiarowego zarejestrowa-

nych podczas lotu dronem *Parrot Bebop 2* ze wszystkimi śmigłami sprawnymi. Rozkład poszczególnych wartości w omawianym wierszu prezentuje się w następujący sposób:

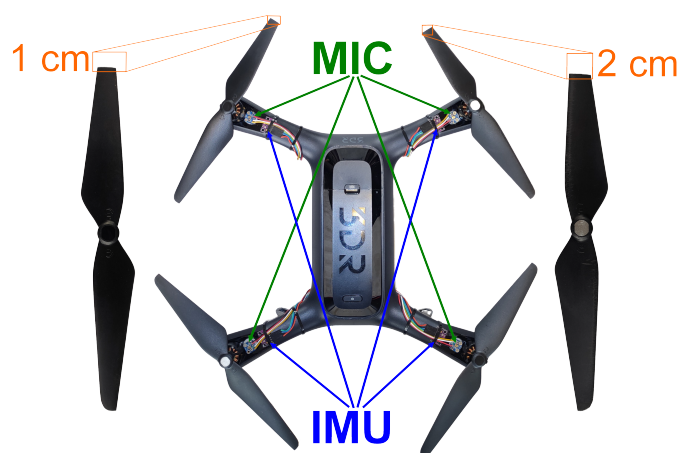
A_ax_approx1 ... A_ax_approx11, A_ax_det1 ... A_ax_det11,
 A_ay_approx1 ... A_ay_approx11, A_ay_det1 ... A_ay_det11,
 A_az_approx1 ... A_az_approx11, A_az_det1 ... A_az_det11,
 B_ax_approx1 ... B_ax_approx11, B_ax_det1 ... B_ax_det11,
 B_ay_approx1 ... B_ay_approx11, B_ay_det1 ... B_ay_det11,
 B_az_approx1 ... B_az_approx11, B_az_det1 ... B_az_det11,
 C_ax_approx1 ... C_ax_approx11, C_ax_det1 ... C_ax_det11,
 C_ay_approx1 ... C_ay_approx11, C_ay_det1 ... C_ay_det11,
 C_az_approx1 ... C_az_approx11, C_az_det1 ... C_az_det11,
 D_ax_approx1 ... D_ax_approx11, D_ax_det1 ... D_ax_det11,
 D_ay_approx1 ... D_ay_approx11, D_ay_det1 ... D_ay_det11,
 D_az_approx1 ... D_az_approx11, D_az_det1 ... D_az_det11.

5.2 Dane z drona *3DR Solo*

Kolejnym dronem, który wykorzystano do zebrania pomiarów z lotów w różnych scenariuszach uszkodzeń, był czterowirnikowy *3DR Solo*. Jego najważniejsze dane techniczne przedstawia tabela 5.3. W tym BSP system akwizycji został podwieszony pod kadłubem. Na ramionach zamontowano czujniki inercyjne oraz mikrofony, co przedstawia rysunek 5.3. Wybrano moduły GY-91 zawierające 3-osiowe akcelerometry, 3-osiowe żyroskopy i barometry (tabela 5.4) oraz układy Adafruit 3492 wyposażone w cyfrowe mikrofony PDM (tabela 5.5).

Tab. 5.3: Wybrane parametry drona *3DR Solo* [1].

Masa	1500 g
Wymiary	250 x 250 x 260 mm
Rozmiar śmigła	25 cm
Pojemność baterii	5200 mAh
Czas lotu	do 25 minut
Maksymalna prędkość lotu poziomego	25,5 m/s
Maksymalna prędkość lotu pionowego	5 m/s
Zasięg łączności radiowej	800 m



Rys. 5.3: 3DR Solo z zamontowanym pod kadłubem systemem akwizycji [103]

Badaniu podlegały śmigła, w których jedna łopata była pozbawiona końcówki o długości 1 cm (uszkodzenie nr 1) lub 2 cm (uszkodzenie nr 2). Zatem w tym przypadku badano jeden rodzaj niesprawności, ale o różnych rozmiarach uszkodzenia. Przeprowadzono 9 lotów z następującymi kombinacjami śmigieł sprawnych (0) i uszkodzonych:

- wszystkie śmigła sprawne: 0000,
- pojedyncze uszkodzenie: 2000, 0200, 0010, 0001,
- podwójne uszkodzenie: 2010, 2001, 0210, 0201.

Loty przeprowadzono na otwartej przestrzeni w ciepły i słoneczny dzień przy zmiennej sile i kierunku wiatru, ale nie powodującym utrudnień w sterowaniu dronem za pomocą dedykowanego kontrolera bezprzewodowego. Poza innymi warunkami środowiskowymi, pozostałe parametry lotu były analogiczne do eksperymentów z *Parrot Bebop 2*. Wykonywane manewry oraz czas akwizycji danych inercyjnych pozostały bez zmian. Każdy lot był dodatkowo wydłużony o ok. 43 sekundy, podczas których przeprowadzano akwizycję danych z mikrofonów. Z uwagi na duże zapotrzebowanie mocy obliczeniowej oraz objętość danych akustycznych, zrezygnowano z równoległej akwizycji ze wszystkich czujników. Sygnały dźwiękowe były zapisywane po zakończeniu akwizycji danych inercyjnych (rysunek 4.2), a czas pomiaru tych sygnałów skrócono czterokrotnie w porównaniu do sygnałów inercyjnych.

Dane z GY-91 pobierano 500 razy na sekundę. Ponieważ barometr ma mniejszą szybkość aktualizacji wartości rejestrów, kilka kolejnych pomiarów (w zależności od ustawień) miało taką samą wartość. Czułość akcelerometru pozostawiono na maksymalnym poziomie ± 16 g, a czułość żyro-

Tab. 5.4: Parametry modułu inercyjnego GY-91 [60].

Układ scalony	MPU-9500	BMP-280
Interfejsy komunikacyjne	I ² C, SPI	I ² C, SPI
Napięcie zasilania	2, 4–3, 6 V	1, 71–3, 6 V
Akcelerometr		
Rozmiar danych	16 bitów	
Liczba osi	3	
Zakres	±2, ±4, ±8, ±16 g	
Czułość	2048–16384 LSB/g	
Szybkość transmisji danych	4000 Hz	
Żyroskop		
Rozmiar danych	16 bitów	
Liczba osi	3	
Zakres	±250, ±500, ±1000, ±2000 dps	
Czułość	16, 4–131 LSB/dps	
Szybkość transmisji danych	8000 Hz	
Magnetometr		
Rozmiar danych	14 bitów	
Liczba osi	3	
Zakres	±4800 μT	
Czułość	0, 6 μT/LSB	
Barometr		
Rozmiar danych		16–20 bitów
Liczba osi		1
Zakres		300–1100 hPa
Szybkość transmisji danych		26, 7–181, 8 Hz

skopu zwiększono do ±2000 dps. Podyktowane to było faktem, iż – pomimo przeprowadzonych wstępnie testów – w niektórych lotach *Bebop 2* odczyty z żyroskopu przez wiele kolejnych pomiarów utrzymywały się w skrajnych wartościach, zatem tracono informację o szybkościach kątowych większych od 1000 *dsp*. Barometr, w celu zmaksymalizowania szybkości akwizycji, ustawiono w tryb 16-bitowy. Pozwoliło to na uzyskanie niezależnych wartości średnio co 3 pomiary.

Podczas każdego lotu na karcie pamięci zapisywane były dwa pliki CSV, jeden dla pomiarów z modułu inercyjnego i jeden dla sygnałów akustycznych. W celu zoptymalizowania czasu przetwarzania danych w mikrokontrolerze oraz zminimalizowania wielkości plików oraz czasu ich zapisu na kartę pamięci, zrezygnowano z konwersji danych do postaci szesnastkowej. Zamiast tego pliki były zapisywane w postaci binarnej (zob. podrozdział 4.3). Takie

Tab. 5.5: Parametry modułu mikrofonowego Adafruit 3492 [124].

Układ scalony	MP34DT01-M
Napięcie zasilania	1, 64–3, 6 V (moduł 1, 8–3, 3 V)
Taktowanie zegara	1–3, 25 MHz
SNR	61 dB
Czułość	–26 dBFS

podejście wymaga dodatkowego etapu przetwarzania podczas późniejszej obróbki w komputerze, ale w zamian odciąża mikroprocesor i kartę pamięci.

Struktura danych z czujników inercyjnych w plikach katalogu `Raw_data` przedstawia się analogicznie do tej opisanej w podrozdziale 5.1:

A_ax_H, A_ax_L, A_ay_H, A_ay_L, A_az_H, A_az_L,
A_gx_H, A_gx_L, A_gy_H, A_gy_L, A_gz_H, A_gz_L,
B_ax_H, B_ax_L, B_ay_H, B_ay_L, B_az_H, B_az_L,
B_gx_H, B_gx_L, B_gy_H, B_gy_L, B_gz_H, B_gz_L,
C_ax_H, C_ax_L, C_ay_H, C_ay_L, C_az_H, C_az_L,
C_gx_H, C_gx_L, C_gy_H, C_gy_L, C_gz_H, C_gz_L,
D_ax_H, D_ax_L, D_ay_H, D_ay_L, D_az_H, D_az_L,
D_gx_H, D_gx_L, D_gy_H, D_gy_L, D_gz_H, D_gz_L,
A_bar_H, A_bar_L, A_bar_X,
B_bar_H, B_bar_L, B_bar_X,
C_bar_H, C_bar_L, C_bar_X,
D_bar_H, D_bar_L, D_bar_X.

Aby pozostawić jak największą zgodność ze strukturą danych z drona *Bebop 2*, wszystkie pomiary z barometru umieszczono na końcu każdego wiersza. Liczba wierszy pozostała taka sama. Dane z barometrów mogą być pozyskiwane z rozdzielczością od 16 do 20 bitów. Pierwsze dwa rejestry barometru są zawsze używane. Podczas pobierania 16-bitowych danych z barometru, rejestr `press_xlsb` (`bar_X`) nie jest wykorzystywany. Dla danych 17-bitowych używany jest tylko najbardziej znaczący bit rejestru `press_xlsb`, dla danych 18-bitowych – dwa bity, dla danych 19-bitowych – trzy bity, a dla danych 20-bitowych – cztery bity. Jednakże, niezależnie od używanej rozdzielczości, wszystkie trzy 8-bitowe rejestry czujnika ciśnienia są zawsze zapisywane na karcie, nawet jeśli ustawiono 16-bitowy rozmiar danych.

Dodatkowo w tym katalogu zamieszczono 9 plików z danymi pobranymi z mikrofonów cyfrowych. Mają one postać 16-bitowych liczb całkowitych bez znaku [125] i również są zapisane w postaci binarnej. Każdy plik zawiera 2064384 wierszy, co dla częstotliwości próbkowania wynoszącej 48 kHz odpowiada dokładnie 43,008 sekundom zarejestrowanych danych akustycznych. Dane w wierszach ułożone są według kolejności oznaczenia śmigieł:

A_mic , B_mic , C_mic , D_mic .

Dane z akcelerometrów oraz żyroskopów zostały także przeskalowane do powszechnie używanych jednostek, natomiast konwersji na liczby zmienno-przecinkowe i normalizacji do zakresu poddano wszystkie dane. Z uwagi na charakter danych z barometru, pomiary ciśnienia przeskalowano do wartości $\langle 0; 1 \rangle$, natomiast pozostałe odczyty, podobnie jak w podrozdziale 5.1, do zakresu $\langle -1; 1 \rangle$. Oczywiście, w przypadku *Solo* skrajne wartości znormalizowanych danych z żyroskopów odpowiadają ± 2000 dps, a nie ± 1000 dps, jak miało to miejsce przy pomiarach z *Bebop 2*. Rozmieszczenie danych w folderze Range_data przedstawia się następująco:

A_ax , A_ay , A_az , A_gx , A_gy , A_gz ,
B_ax , B_ay , B_az , B_gx , B_gy , B_gz ,
C_ax , C_ay , C_az , C_gx , C_gy , C_gz ,
D_ax , D_ay , D_az , D_gx , D_gy , D_gz .

Natomiast katalog Normalized_data zawiera pomiary z modułów GY-91 w następującej postaci:

A_ax , A_ay , A_az , A_gx , A_gy , A_gz , A_bar ,
B_ax , B_ay , B_az , B_gx , B_gy , B_gz , B_bar ,
C_ax , C_ay , C_az , C_gx , C_gy , C_gz , C_bar ,
D_ax , D_ay , D_az , D_gx , D_gy , D_gz , D_bar

oraz sygnały akustyczne w niezmienionej w stosunku do surowych danych kolejności:

A_mic , B_mic , C_mic , D_mic .

W przypadku danych częstotliwościowych, ponownie wykorzystano algorytm FFT. Z uwagi na zupełnie inny charakter oraz częstotliwość próbkowania pomiarów bezwładnościowych i sygnałów akustycznych, ich reprezentacje w dziedzinie częstotliwości zawsze są zawarte w osobnych plikach. Natomiast dane z akcelerometru, żyroskopu i barometru były przygotowywane

w różnych kombinacjach. Niektóre pliki dotyczą tylko jednego rodzaju czujnika lub nawet konkretnej jego osi, inne zawierają odczyty z dwóch lub trzech czujników w różnych konfiguracjach poszczególnych osi pomiarowych. Ponownie zastosowano różne parametry przetwarzania sygnałów, o czym świadczą nazwy poszczególnych podfolderów. Podstawową różnicą w stosunku do analogicznych danych z *Parrot Bebop 2* było zastosowanie tylko jednej, dobrze przetestowanej funkcji okna lub całkowite zrezygnowanie z jej używania. Postanowiono także ominąć etap wycinania poszczególnych częstotliwości z całego spektrum sygnału. Było to podyktowane próbami optymalizacji czasowej przetwarzania sygnałów. Metodyka przeprowadzania testów, ich wyniki oraz przyczyny podjęcia decyzji o zastosowaniu bądź rezygnacji z poszczególnych etapów przetwarzania, szerzej zostały opisane w rozdziale 7.

Także w przypadku ekstrakcji cech mających mieszany charakter dziedzin czasu i częstotliwości, wykorzystano dane z akcelerometrów i żyroskopów. Oprócz DWT opisanego w podrozdziale 5.1, posłużono się tutaj także innymi, bardziej zaawansowanymi technikami. Pierwszą z nich jest EMD (empiryczna analiza modów, ang. *Empirical Mode Decomposition*), która znajduje zastosowanie w analizie sygnałów niestacjonarnych i redukcji wpływu szumów. W folderze EMD_data znajdują się osobno pliki dla akcelerometrów i żyroskopów. W nazwach plików zawarto informacje o zakresie pomiarowym czujnika, szerokości okna pomiarowego, liczbie modów IMF (wewnętrzna funkcja modów, ang. *Intrinsic Mode Function*) oraz liczbie parametrów przypadających na każdą oś czujnika.

Kolejnym przekształceniem pozwalającym na ekstrakcję cech czasowo-częstotliwościowych jest HHT (transformacja Hilberta-Huanga, ang. *Hilbert-Huang Transform*). Dane przetworzone z użyciem tej techniki, zawarto w katalogu HHT_data. Struktura plików zawartych w tym folderze jest analogiczna do danych z EMD_data. Poza nieco inną zawartością informacyjną, każda oś czujnika – niezależnie od szerokości okna pomiarowego – zawiera 101 unikalnych cech.

5.3 Ocena danych pod względem statystycznym

Dane ze wszystkich osi czujników, które podlegały akwizycji, zostały sprawdzone pod kątem ich kompletności i poprawności. Zebrane dane zbadało pod względem statystycznym. W tabeli 5.6 zestawiono podstawowe parametry statystyczne znormalizowanych danych dziedziny czasu zebranych podczas lotów dronem *Parrot Bebop 2*. Uwzględniono pomiary ze wszystkich 20 badanych kombinacji uszkodzeń. Poszczególne klasy nie były w tym przypadku rozpatrywane osobno i wszystkie statystyczne parametry odnoszą się do pomiarów pogrupowanych według poszczególnych osi użytych czujników danego drona, natomiast odczyty z tych samych osi podczas poszczególnych lotów nie są rozdzielone. Kolejne kolumny tabeli przedstawiają liczbę pomiarów każdej osi czujnika, średnią arytmetyczną, odchylenie standardowe, wartość minimalną, pierwszy kwartył, medianę, trzeci kwartył oraz wartość maksymalną.

Histogramy wszystkich odczytów z drona *Parrot Bebop 2* w postaci wykresów KDE (jądrowy estymator gęstości, ang. *Kernel Density Estimation*) [143] zostały przedstawione graficznie na rysunku 5.4. Wykres KDE odwzorowuje kształt funkcji estymatora gęstości jądra opisanej wzorem (5.5):

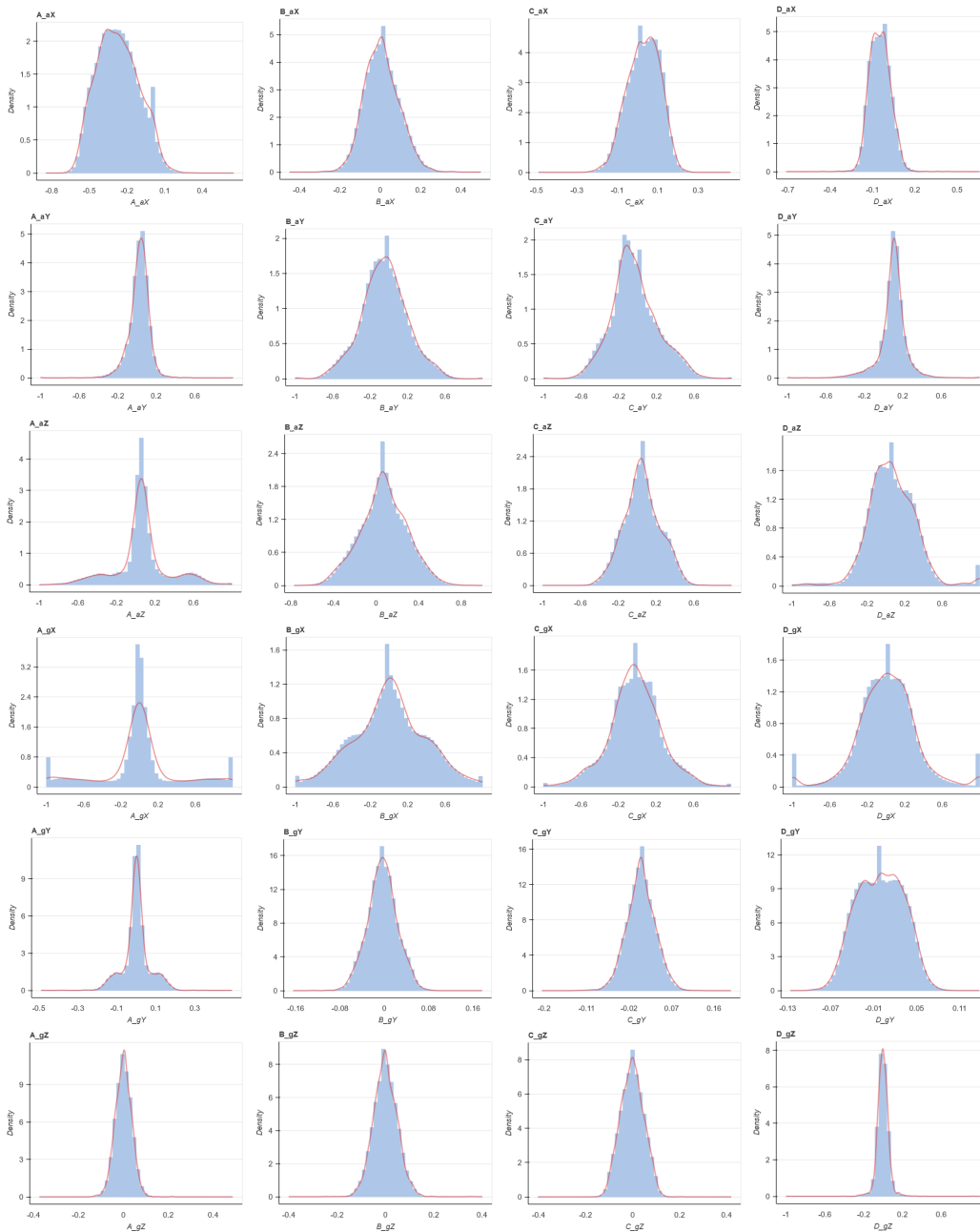
$$\hat{f}_h(x) = \frac{1}{N} \sum_{i=1}^N K_h(x - x_i), \quad (5.5)$$

gdzie x_i to niezależna próbka pobrana z pewnego rozkładu o nieznannej gęstości f w dowolnym punkcie x , natomiast K_h to nieujemna funkcja jądra dla szerokości pasma oznaczonej parametrem h . Osie pionowe, opisane jako *Density*, prezentują, jak często występuje wartość oznaczona na osi poziomej. W każdym wierszu przedstawiono wykresy opisujące jedną oś czujnika, począwszy od osi x akcelerometru, a kończąc na osi z żyroskopu. Kolejne kolumny odpowiadają ramionom drona, na których umieszczony był dany czujnik, od ramienia A z lewej strony rysunku do ramienia D z prawej.

Z kolei korelacja różnych osi czujników zamontowanych na pokładzie *Parrot Bebop 2* jest przedstawiona na rysunku 5.5. Do prezentacji wzajemnych zależności poszczególnych par osi zastosowano współczynnik korelacji

Tab. 5.6: Dane statystyczne z drona *Parrot Bebop 2* (opracowanie na podstawie [103]).

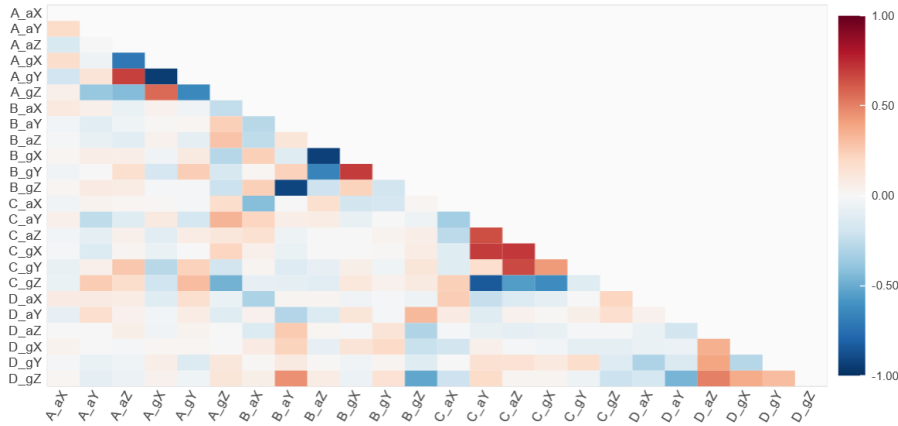
oś	liczba	średnia	odch.	min.	25%	50%	75%	maks.
A_aX	1720320	-0,272	0,163	-0,845	-0,396	-0,280	-0,158	0,650
A_aY	1720320	0,024	0,104	-1,000	-0,022	0,034	0,084	1,000
A_aZ	1720320	0,061	0,270	-1,000	-0,012	0,059	0,130	1,000
A_gX	1720320	-0,001	0,428	-1,000	-0,093	-0,001	0,092	1,000
A_gY	1720320	0,002	0,066	-0,487	-0,022	0,002	0,026	0,491
A_gZ	1720320	0,001	0,037	-0,373	-0,023	0,001	0,026	0,487
B_aX	1720320	0,010	0,085	-0,449	-0,049	0,007	0,065	0,499
B_aY	1720320	-0,049	0,247	-1,000	-0,203	-0,051	0,102	1,000
B_aZ	1720320	0,068	0,217	-0,770	-0,072	0,065	0,209	1,000
B_gX	1720320	-0,001	0,381	-1,000	-0,250	-0,001	0,249	1,000
B_gY	1720320	-0,003	0,027	-0,165	-0,020	-0,003	0,014	0,177
B_gZ	1720320	0,000	0,050	-0,400	-0,032	0,000	0,033	0,401
C_aX	1720320	0,033	0,079	-0,490	-0,023	0,037	0,093	0,460
C_aY	1720320	-0,035	0,248	-1,000	-0,186	-0,058	0,108	1,000
C_aZ	1720320	0,054	0,193	-1,000	-0,072	0,052	0,179	1,000
C_gX	1720320	-0,016	0,276	-1,000	-0,183	-0,016	0,149	1,000
C_gY	1720320	0,005	0,029	-0,205	-0,014	0,005	0,024	0,196
C_gZ	1720320	-0,001	0,049	-0,403	-0,035	-0,001	0,033	0,420
D_aX	1720320	-0,045	0,072	-0,710	-0,099	-0,046	0,003	0,659
D_aY	1720320	0,099	0,139	-1,000	0,052	0,109	0,162	1,000
D_aZ	1720320	0,067	0,254	-1,000	-0,094	0,058	0,225	1,000
D_gX	1720320	0,006	0,312	-1,000	-0,172	0,007	0,184	1,000
D_gY	1720320	-0,001	0,032	-0,126	-0,026	-0,001	0,024	0,138
D_gZ	1720320	-0,001	0,059	-1,000	-0,033	-0,001	0,031	1,000



Rys. 5.4: Wykresy jądrowego estymatora gęstości pomiarów z drona *Parrot Bebop 2* (opracowanie na podstawie [103])

Pearsona [116], którego sposób wyliczania dla dwóch zmiennych x i y przedstawia wzór (5.6):

$$r_{xy} = \frac{1}{N-1} \sum_{i=1}^N \left(\frac{x_i - \bar{x}}{s_x} \right) \left(\frac{y_i - \bar{y}}{s_y} \right). \quad (5.6)$$



Rys. 5.5: Współczynniki korelacji Pearsona poszczególnych par osi czujników z drona *Parrot Bebop 2* (opracowanie na podstawie [103])

Wszystkie dane sensoryczne z *Parrot Bebop 2* są równoliczne i zawierają po 1720320 pomiarów z każdej osi akcelerometrów i żyroskopów. Poza osią x akcelerometru z ramienia A , średnia odczytów jest bardzo zbliżona do wartości 0. Odchylenie standardowe jest największe dla osi x żyroskopu z ramienia oznaczonego literą A . Jednocześnie na wykresie dotyczącym A_gX z rysunku 5.4 widać, że ustawiony zakres ± 1000 dps okazał się niewystarczający i znaczna liczba tych odczytów osiągnęła skrajne wartości. Podobna sytuacja ma miejsce na wykresie D_gX , a także B_gX , choć w mniejszym stopniu. Z rysunku 5.5 wywnioskować można występowanie pozytywnej korelacji pomiędzy następującymi parami osi czujników: A_gY-A_aZ , A_gZ-A_gX , B_gY-B_gX , C_aZ-C_aY , C_gX-C_aY , C_gX-C_aZ oraz C_gY-C_aY . Negatywną korelację widać przede wszystkim w parach A_gY-A_gX , B_gX-B_aZ , B_gZ-B_aY oraz C_gZ-C_aY . Nie powinien dziwić fakt, że wymienione pary zawsze dotyczą tego samego ramienia. Świadczy to o zupełnie zrozumiałym zjawisku wpływu uszkodzenia śmigła na odczyty w różnych osiach akcelerometru i żyroskopu zamontowanych na jednym ramieniu z niesprawnym śmigłem.

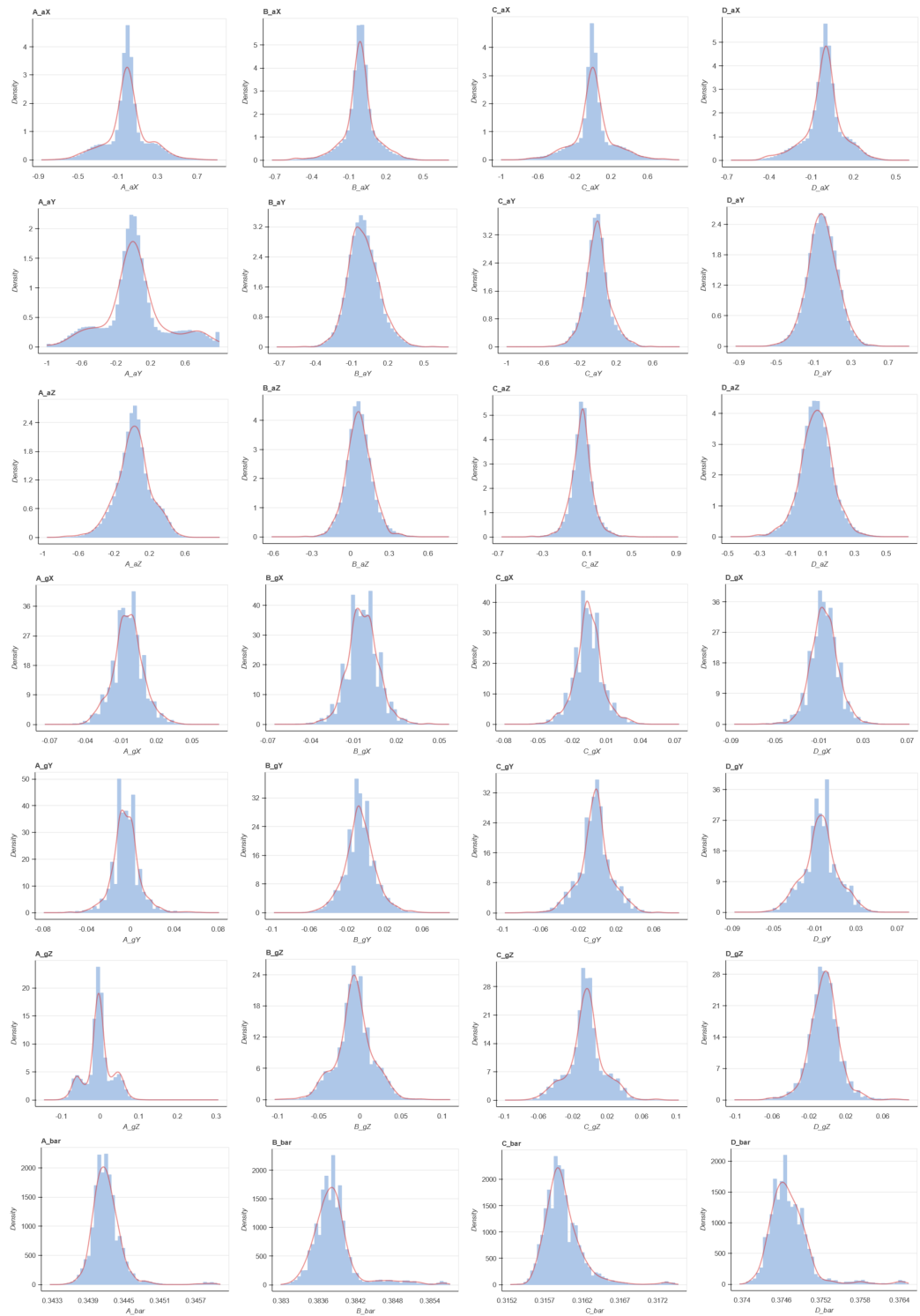
Dane statystyczne odczytów z czujników zamontowanych na dronie *3DR Solo* przedstawiono w tabeli 5.7. Rysunek 5.6 prezentuje histogramy KDE dla pomiarów z akcelerometru, żyroskopu i barometru, zamontowanych na tym

BSP. Ich układ, poza dodanymi na końcu odczytami z czujnika ciśnieniowego, jest analogiczny do histogramów dotyczących odczytów z *Bebop 2*. Wykresy KDE dla odczytów z mikrofonu przedstawiono graficznie na rysunku 5.7. Rysunek 5.8 pokazuje korelacje czujników inercyjnych (w tym barometru) z drona *3DR Solo*, a na rysunku 5.9 przedstawiono korelacje pomiędzy poszczególnymi mikrofonami.

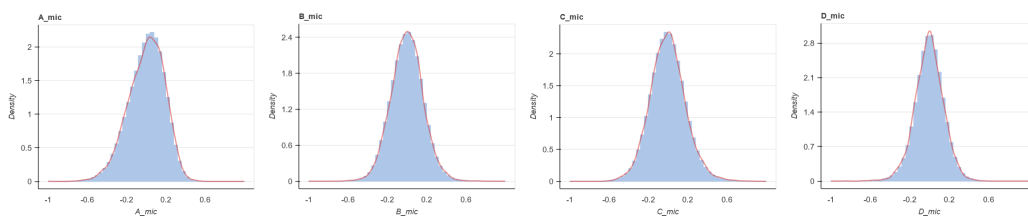
Tab. 5.7: Dane statystyczne z drona *3DR Solo* (opracowanie na podstawie [103]).

oś	liczba	średnia	odch.	min.	25%	50%	75%	maks.
A_aX	774144	-0,004	0,204	-1,000	-0,126	-0,005	0,114	0,993
A_aY	774144	0,027	0,318	-1,000	-0,123	0,018	0,162	0,996
A_aZ	774144	0,004	0,203	-1,000	-0,117	0,016	0,130	0,991
A_gX	774144	-0,004	0,013	-0,120	-0,011	-0,004	0,003	0,081
A_gY	774144	-0,004	0,018	-0,143	-0,016	-0,004	0,007	0,111
A_gZ	774144	-0,004	0,031	-0,254	-0,019	-0,004	0,012	0,306
A_bar	774144	0,343	0,001	0,341	0,342	0,344	0,344	0,346
A_mic	18579456	0,000	0,182	-1,000	-0,116	0,015	0,130	1,000
B_aX	774144	-0,006	0,207	-1,000	-0,117	-0,002	0,106	0,995
B_aY	774144	-0,003	0,368	-1,000	-0,163	-0,009	0,170	0,996
B_aZ	774144	0,067	0,145	-0,823	-0,024	0,059	0,150	0,958
B_gX	774144	-0,004	0,014	-0,111	-0,011	-0,003	0,005	0,105
B_gY	774144	-0,006	0,029	-0,149	-0,023	-0,006	0,011	0,197
B_gZ	774144	-0,006	0,047	-0,211	-0,034	-0,006	0,022	0,330
B_bar	774144	0,383	0,001	0,380	0,382	0,383	0,384	0,386
B_mic	18579456	0,000	0,164	-1,000	-0,109	0,001	0,107	1,000
C_aX	774144	-0,001	0,191	-1,000	-0,098	0,002	0,098	0,992
C_aY	774144	-0,009	0,205	-1,000	-0,129	-0,005	0,126	0,992
C_aZ	774144	0,060	0,121	-0,806	-0,011	0,059	0,131	0,919
C_gX	774144	-0,006	0,014	-0,122	-0,014	-0,006	0,002	0,109
C_gY	774144	-0,002	0,025	-0,118	-0,018	-0,002	0,013	0,135
C_gZ	774144	-0,005	0,026	-0,166	-0,022	-0,004	0,013	0,159
C_bar	774144	0,315	0,001	0,313	0,314	0,316	0,316	0,319
C_mic	18579456	0,000	0,180	-1,000	-0,118	-0,004	0,113	1,000
D_aX	774144	-0,003	0,185	-1,000	-0,108	0,000	0,104	0,996
D_aY	774144	-0,005	0,272	-1,000	-0,175	-0,007	0,157	0,996
D_aZ	774144	0,062	0,147	-0,880	-0,029	0,059	0,152	0,840
D_gX	774144	-0,005	0,016	-0,133	-0,016	-0,005	0,005	0,148
D_gY	774144	-0,004	0,024	-0,149	-0,019	-0,004	0,012	0,113
D_gZ	774144	-0,004	0,026	-0,149	-0,020	-0,005	0,011	0,159
D_bar	774144	0,374	0,001	0,371	0,373	0,374	0,375	0,377
D_mic	18579456	0,000	0,146	-1,000	-0,089	0,002	0,093	1,000

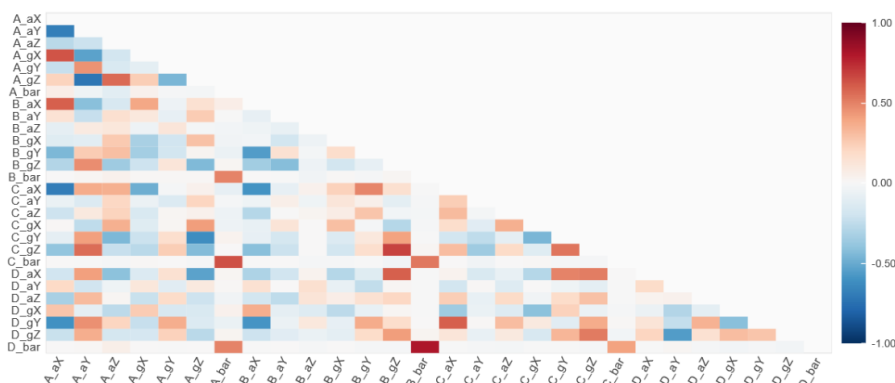
W przypadku drona *3DR Solo* liczba pomiarów z każdej osi akcelerometru, żyroskopu i barometru wynosi 774144 odczytów. Pomiarów z mikrofonów liczą po 18579456 wartości. Poza odczytami barometrycznymi, które mają nieujemne wartości, średnia arytmetyczna pozostałych odczytów oscyluje



Rys. 5.6: Wykresy jądrowego estymatora gęstości odczytów inercyjnych z drona 3DR Solo (opracowanie na podstawie [103])



Rys. 5.7: Wykresy jądrowego estymatora gęstości odczytów mikrofonowych z drona *3DR Solo* (opracowanie na podstawie [103])



Rys. 5.8: Współczynniki korelacji Pearsona poszczególnych par osi czujników inercyjnych z drona *3DR Solo* (opracowanie na podstawie [103])



Rys. 5.9: Współczynniki korelacji Pearsona poszczególnych mikrofonów z drona *3DR Solo* (opracowanie na podstawie [103])

w okolicy 0. Zauważalną różnicą w stosunku do pomiarów z *Bebop 2* jest dużo mniejsze odchylenie oraz maksymalne wartości skrajne danych żyroskopowych. O ile we wcześniejszych eksperymentach zakres żyroskopu był niewystarczający, o tyle w przypadku *Solo* wartość ± 2000 dps jest wyraźnie zbyt duża i mogłaby być przynajmniej dwukrotnie mniejsza. Zapewniłoby to dokładniejsze pomiary z większą rozdzielczością. Jej niedobór widoczny jest na wykresach dotyczących żyroskopów z rysunku 5.6, które są bardziej nieregularne od wykresów z pomiarami przyspieszeń liniowych. Pokazuje

to konieczność indywidualnego dobierania zakresów pomiarowych czujników odpowiednio do używanego drona oraz rodzajów i wielkości badanych uszkodzeń. Współczynniki korelacji osi akcelerometrów i żyroskopów na rysunku 5.8 rozkładają się podobnie jak w przypadku *Bebop 2*, jednak zarówno korelacja pozytywna, jak i negatywna, nie osiągają skrajnych wartości. Dla odczytów mikrofonowych (rysunek 5.9) korelacja praktycznie nie występuje. Może to świadczyć o dobrej izolacji dźwięków z badanego rotora od hałasu generowanego przez pozostałe śmigła.

Diagnostyka uszkodzeń w czasie rzeczywistym

Dotychczas opisane prace dotyczące sprzętu i oprogramowania do akwizycji danych, ich późniejsza obróbka oraz przygotowanie repozytorium PADDRE, były etapami przygotowawczymi do realizacji najważniejszego celu. Jak wskazuje tytuł rozprawy, wszystkie wcześniej wykonane czynności prowadziły do zaprojektowania systemu, który skutecznie będzie przeprowadzał diagnostykę jednostek napędowych bezzałogowego statku powietrznego na potrzeby wykrycia ewentualnych nieprawidłowości w pracy śmigieł BSP. Dodatkowo za cel postawiono zdolność przeprowadzania detekcji w trybie ciągłym, bez zbędnych opóźnień, bezpośrednio podczas wykonywanego lotu. Ponadto system powinien być w stanie określić rodzaj wykrytej usterki, rozmiar uszkodzenia oraz miejsce jego wystąpienia. Dysponując taką wiedzą, operator lub autonomiczny system kontrolujący BSP podczas lotu może podjąć decyzję o dalszych krokach. Jeżeli uszkodzenie jest niewielkie, misja może być kontynuowana, a przegląd i ewentualna naprawa/wymiana niesprawnych elementów zostanie przeprowadzona po wylądowaniu. W przypadku krytycznego defektu, pojazd powinien zostać sprowadzony na ziemię w trybie awaryjnym, aby ograniczyć potencjalne uszkodzenia własne lub otoczenia oraz aby zminimalizować ryzyko zranienia ludzi lub zwierząt, jeśli w ich zasięgu operuje. Najbardziej złożone zadanie czeka system sterujący w przypadku awarii, która umożliwia kontynuowanie lotu, ale w zauważalny sposób zmienia jego parametry. Konieczna może być wtedy kompensacja lub zastąpienie niedziałającego aktuatora przez inne – sprawne. Tymi problemami zajmuje się szeroko pojęta tematyka FTC [108]. Obszerna, bardzo istotna problematyka badawcza FTC wykracza poza ramy tematyczne zagadnień przyjęte w ramach niniejszej rozprawy. Tutaj uwaga skupia się na rozwiązaniach dostarczających możliwie najszybszej i niezawodnej informacji o charakterze występujących uszkodzeń.

6.1 Opracowanie systemu detekcji uszkodzeń

Zaproponowany system, który realizuje zadania detekcji uszkodzeń, jako dane wejściowe przyjmuje pomiary z czujników umieszczonych w pobliżu silników napędzających śmigła osadzone bezpośrednio na jego wale. Ponieważ dokładnie takie same pomiary wykonywał system akwizycji omówiony w rozdziale 4, do celów detekcji uszkodzeń postanowiono wykorzystać ten sam układ. Jego wygląd przedstawiono na rysunku 4.1. Od strony sprzętowej wymagana jest tutaj nawet mniejsza złożoność, ponieważ nie ma potrzeby zapisywania danych, na podstawie których przeprowadzana jest diagnostyka. Jednak gniazdo kart pamięci zostało wykorzystane również tutaj. Posłużyło do zapisywania wyników oraz najważniejszych parametrów detekcji i klasyfikacji uszkodzeń. Zatem cały fizyczny układ z systemu akwizycji został zaadaptowany na potrzeby autodiagnostyki BSP.

6.2 Przygotowanie oprogramowania

Pomimo zastosowania tej samej platformy sprzętowej, co w zadaniu akwizycji danych, z powodu innego przeznaczenia, system detekcji wymagał gruntownej przebudowy oprogramowania. Głównym zadaniem tego systemu jest wykrywanie i określanie najważniejszych parametrów nieprawidłowości w wirnikach BSP. Zasadniczym elementem takiego układu jest zatem algorytm, którego zadanie to określanie stanu pojazdu podczas lotu. Z uwagi na gwałtowny rozwój możliwości mikrokontrolerów w zakresie zastosowania metod uczenia maszynowego [110], zdecydowano się na użycie klasyfikatora w formie SSN. Jego rola to przypisanie aktualnego stanu śmigieł do jednej ze zdefiniowanych klas, określających rodzaj wykrytego uszkodzenia, jego rozmiar i miejsce wystąpienia usterki.

6.2.1 Trenowanie klasyfikatora

Wcześniej zebrane dane, udostępnione w ramach repozytorium PADRE, stanowiły informacje, na bazie których przeprowadzono proces przygotowania klasyfikatorów [102]. Skorzystano ze znormalizowanych danych, które są najlepszym wyborem w przypadku korzystania z sieci neuronowych [22].

Wykorzystano dane ze wszystkich lotów i przydzielono odpowiednie klasy uszkodzeń poszczególnym scenariuszom lotu. W ten sposób uzyskano 20 klas dla *Parrot Bebop 2* oraz 9 klas dla *3DR Solo*. Do przygotowania modelu SSN kompatybilnego z 32-bitowym mikrokontrolerem *STM* wykorzystano środowisko *Anaconda* oraz notatnik *JupyterLab* i język programowania *Python*. Głównymi bibliotekami związanymi z uczeniem maszynowym były *Keras* i *TensorFlow*.

Niezależnie od tego, czy korzystano z danych w dziedzinie czasu, czy częstotliwości, pomiarów obejmujących wszystkie lub wybrane czujniki albo tylko niektóre osie pomiarowe, proces trenowania modelu SSN wygląda bardzo podobnie. Po wczytaniu wszystkich potrzebnych plików (np. 20 plików z pomiarami żyroskopowymi z podfolderu `FFT_128_Hann_16_36` drona *Parrot Bebop 2*), na końcu każdego wiersza jest dopisywana etykieta w postaci liczby od 0 do 19. Są to kolejne numery poszczególnych klas uszkodzeń, gdzie 0 odpowiada sprawnym śmigłom, a 19 – poczwórnemu uszkodzeniu. Dane ze wszystkich przygotowanych w ten sposób plików są następnie losowo dzielone na 3 zbiory: **treningowy**, który obejmuje 70% pomiarów, **walidacyjny**, zawierający 15% wierszy oraz **testowy** z pozostałymi 15% danych. Proporcje są równe dla każdego pliku, a ponieważ wszystkie pliki są tej samej wielkości, finalne zbiory zawierają odpowiednio 70%, 15% i 15% wszystkich danych poddawanych przetwarzaniu.

Następny krok jest kluczowy, gdyż następuje w nim ustalenie hiperparametrów sieci neuronowej oraz przeprowadzenie procesu jej trenowania. Na tym etapie ustalana jest architektura sieci, liczba i rodzaje użytych warstw, liczba neuronów w poszczególnych warstwach oraz liczba epok, podczas których algorytm będzie znajdował wagi dające możliwie najlepsze metryki sieci, według wcześniej zdefiniowanych założeń. Jest to najbardziej czasochłonne stadium przygotowywania modelu, które wymaga wielu powtórzeń oraz dopasowywania struktury, hiperparametrów i technik optymalizacji.

Ponieważ opisywany problem dotyczy wieloklasowej klasyfikacji, w której poszczególnym elementom przypisać można tylko jedną etykietę, funkcję kosztu pełni *kategorialna entropia krzyżowa*. Funkcja aktywacji warstw pośrednich to ReLU (rektyfikowana jednostka liniowa, ang. *Rectified Linear Unit*), natomiast aktywację ostatniej warstwy obsługuje *softmax* [22].

Kategorialna (bądź kategoryzacyjna) entropia krzyżowa (ang. *categorical crossentropy*) to funkcja używana w procesie trenowania, którego celem jest

minimalizacja jej wartości. Jest to więc tzw. funkcja straty, której wartość – w idealnym scenariuszu – powinna dążyć do zera. Można ją przedstawić za pomocą równania (6.1) [152, 53, 101]:

$$\mathcal{L}_{CCE} = -\frac{1}{M} \sum_{k=1}^K \sum_{m=1}^M y_m^k \times \log(h_{\theta}(x_m, k)), \quad (6.1)$$

gdzie:

- M liczba zbiorów danych treningowych,
- K liczba predefiniowanych klas,
- y_m^k etykieta docelowa dla treningowego zbioru danych m dla klasy k ,
- x_m dane wejściowe dla treningowego zbioru danych m ,
- h_{θ} model z obliczonymi wagami sieci neuronowej θ .

We wszystkich warstwach poza ostatnią, niezależnie od ich liczby, zastosowano funkcję aktywacji ReLU, której głównym zadaniem jest zerowanie wartości ujemnych. Nie zmienia ona wartości dodatnich. Jej kształt opisuje wzór (6.2):

$$ReLU(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}, \quad (6.2)$$

gdzie x jest wejściem do neuronu.

Z kolei funkcja *softmax*, używana wyłącznie w zastosowaniu do warstwy wyjściowej, jest uogólnieniem logistycznej funkcji sigmoidalnej (6.3):

$$\sigma(x) = \frac{1}{1 + \exp(-a \cdot x)}, \quad (6.3)$$

gdzie a to współczynnik wzrostu logistycznego [14]. Funkcja aktywacji *softmax* działa z wektorami \mathbf{x} o wartościach rzeczywistych (6.4):

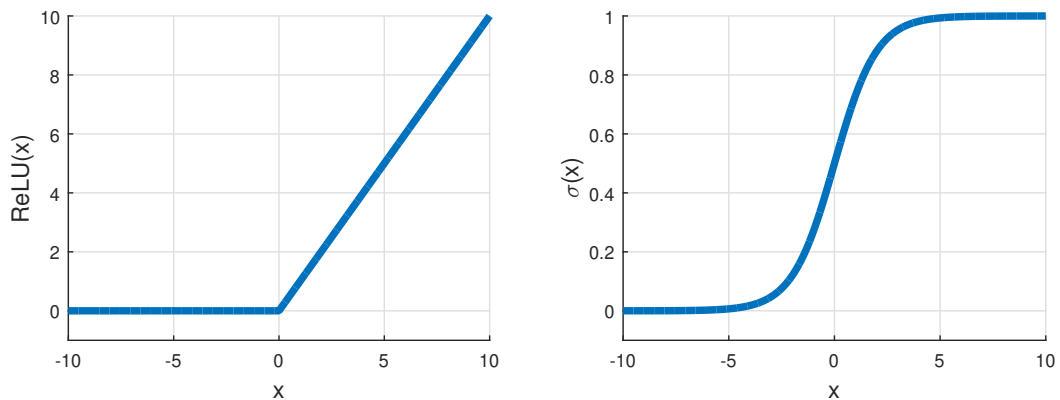
$$softmax(\mathbf{x}) = \frac{1}{\sum_{j=1}^n \exp(x_j)} \begin{bmatrix} \exp(x_1) \\ \exp(x_2) \\ \vdots \\ \exp(x_n) \end{bmatrix}. \quad (6.4)$$

Konwertuje ona wartości na skalary, których suma zawsze wynosi 1. W ten sposób wyniki można interpretować jako prawdopodobieństwo wystąpienia

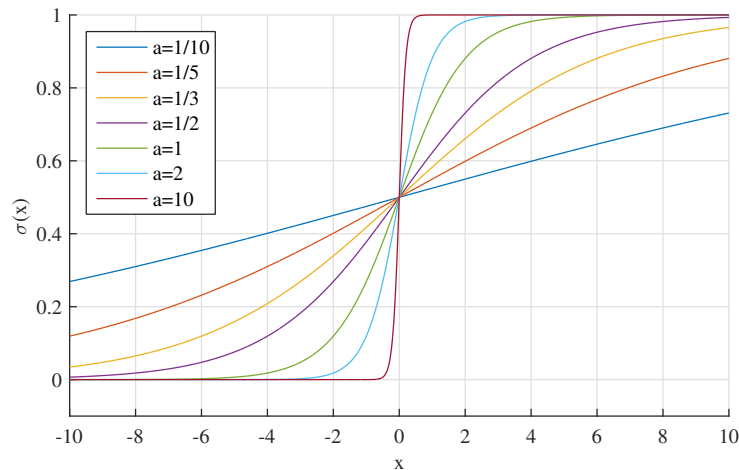
danej klasy. Funkcja ta jest powszechnie definiowana w literaturze [76, 42, 85, 32, 147, 73, 39, 36, 12] jako (6.5):

$$\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_{j=1}^n \exp(x_j)}. \quad (6.5)$$

Na rysunku 6.1 przedstawiono wykresy funkcji ReLU oraz krzywej sigmoidalnej, a rysunek 6.2 prezentuje różnice w kształcie wykresu funkcji logistycznej w zależności od wartości współczynnika wzrostu logistycznego a .



Rys. 6.1: Wykresy funkcji *ReLU* oraz krzywej logistycznej

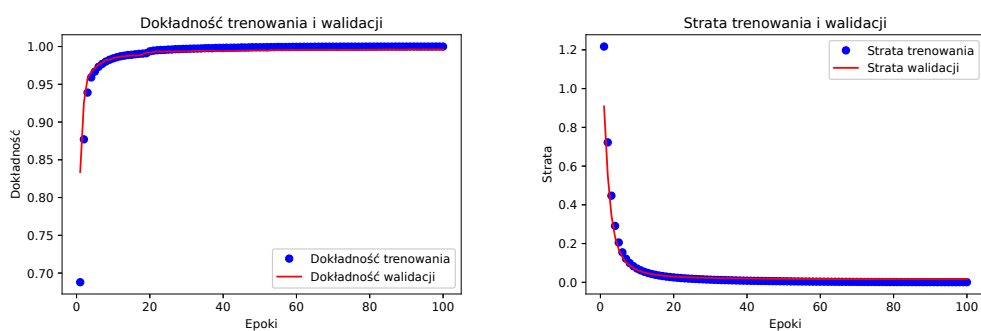


Rys. 6.2: Wykresy funkcji logistycznych z różnymi parametrami a (opracowanie na podstawie [119])

Zastosowano regularyzację wag typu $L2$ o wartości $0,01$, a w zastosowaniu do sieci jednokierunkowych także technikę porzucania *dropout*, której współczynnik ustalony został na $0,5$. Techniki te zmniejszają niepożądany wpływ zjawiska nadmiernego dopasowania sieci do danych treningowych.

Dokładny sposób stosowania gradientu straty w czasie aktualizacji wag sieci określa optymalizator. W badanych modelach korzystano z optymalizatora RMSProp (propagacja średniej kwadratowej, ang. *Root Mean Squared Propagation*). Jest on adaptacyjnym algorytmem optymalizacji szybkości uczenia się, zaprojektowanym w celu rozwiązywania problemów metodą stochastycznego spadku wzdłuż gradientu [7].

Jako metrykę objętą monitorowaniem podczas etapu trenowania wybrano *accuracy*, natomiast w czasie wywołania zwrotnego *ModelCheckpoint*, który w sposób zautomatyzowany zapisuje najlepsze modele na przestrzeni wszystkich epok, monitorowany był *validation_loss*. Wykres zmian dokładności i straty podczas pierwszych 100 epok procesu trenowania i walidacji przykładowego modelu przedstawia rysunek 6.3.



Rys. 6.3: Wykresy dokładności i straty na etapie trenowania i walidacji modelu SSN

Wszystkie pozostałe atrybuty modelu dobierano eksperymentalnie, przeprowadzając kolejne testy, po których porównywano przede wszystkim metryki *test_accuracy* i *test_loss*, w odniesieniu do złożoności obliczeniowej i rozmiaru wynikowego pliku oraz czasu przetwarzania w docelowym mikrokontrolerze. Te ostatnie decydowały o możliwości użycia modelu w systemie wbudowanym oraz zdolności do przeprowadzania klasyfikacji uszkodzeń w czasie rzeczywistym.

Po zakończeniu procesu trenowania, w którym wykorzystywane były zbiory treningowy i walidacyjny, przeprowadzany był test otrzymanego modelu z użyciem wydzielonych danych testowych. Są to zupełnie nowe informacje, do których model dotąd nie miał dostępu. Taki sprawdzian daje możliwość odrzucenia modeli, które są nadmiernie dopasowane do danych treningowych, a gorzej radzą sobie z nowymi danymi wejściowymi. Po przejściu testu, czyli uzyskaniu modelu cechującego się odpowiednią wielkością (umożliwiająca jego wykorzystanie w mikrokontrolerze), wysokim wskaź-

nikiem *accuracy* oraz niskim wskaźnikiem *loss*, plik z modelem zapisywany był na dysku komputera. W ten sposób uzyskiwano klasyfikator uszkodzeń w postaci *SSN* przygotowany techniką uczenia nadzorowanego.

6.2.2 Implementacja klasyfikatora w mikrokontrolerze

Producent *STMicroelectronics* opracował dla swoich najwydajniejszych mikrokontrolerów dodatek *X-CUBE-AI*. Jest to rozszerzenie instalowane w programie *CubeIDE*, które służy do przygotowania oprogramowania dla wspieranych układów. Dodatek ten umożliwia implementację technik SI (sztuczna inteligencja, ang. *Artificial Intelligence*) w niewielkich mikroprocesorach. Po wybraniu zapisanego wcześniej modelu, w kodzie głównym programu napisanym w języku *C* pojawia się nowa funkcja *AI* (SI) odpowiedzialna za przeprowadzanie klasyfikacji. W jej ciele należy zdefiniować zmienne typu *float* odpowiedzialne za poszczególne klasy. Ich wartości po przeprowadzeniu obliczeń będą świadczyć o wykrytej klasie. Po każdym wywołaniu funkcji *AI* na nowo ustalane są te wartości, a ich suma wynosi 1. Zatem można przyjąć, że przypisana wielkość odpowiada prawdopodobieństwu wykrycia konkretnej klasy przy użyciu wgranego modelu *SSN*. *X-CUBE-AI* daje możliwość skontrolowania m.in. zajętości pamięci RAM i Flash oraz złożoności obliczeniowej wyrażonej w jednostkach *MACC* (złożoność mnożenia i akumulacji, ang. *Multiply-and-Accumulate Complexity*), które wskazują poziom kompleksowości modelu głębokiego uczenia z punktu widzenia przetwarzania danych [127].

6.3 Działanie systemu detekcji

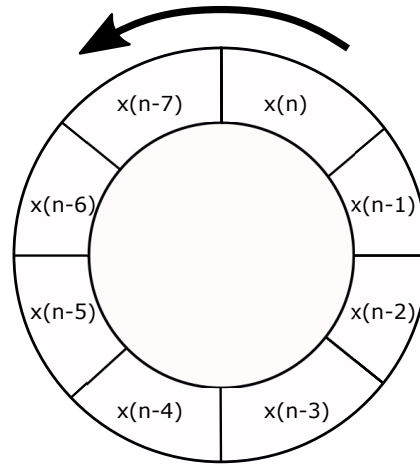
Po uruchomieniu systemu w trybie detekcji, podobnie jak w przypadku trybu akwizycji opisanym w podrozdziale 4.2, następuje inicjalizacja używanych peryferii oraz konfiguracja czujników i uruchomienie przerwań timera decydującego o częstotliwości próbkowania. Na etapie przygotowywania oprogramowania należy wybrać używane czujniki i ewentualnie konkretne osie czujników wieloosiowych. O ile w trybie akwizycji układ pracował zarówno z sygnałami inercyjnymi, jak i dźwiękowymi, to w tym przypadku konieczne jest podjęcie decyzji, z użyciem którego typu odczytów będzie przeprowadzana detekcja uszkodzeń. Choć możliwe jest łączenie pomiarów

z akcelerometrów, żyroskopów i barometrów praktycznie w nieograniczonych kombinacjach, to sygnały akustyczne wymagają separacji od innych pomiarów. Spowodowane jest to różną szybkością próbkowania oraz koniecznością zastosowania innego rodzaju przetwarzania danych, w tym konwersji z PDM do PCM. Po wywołaniu przerwania przeprowadzany jest odczyt danych z rejestrów pomiarowych z czujników. Następnie są one konwertowane na liczby zmiennoprzecinkowe oraz poddawane normalizacji, uwzględniając skrajne zakresy czujników zdefiniowane na etapie ich konfiguracji. Tak przygotowane dane zapisywane są do bufora. Jeżeli przetwarzaniu podlegają sygnały w dziedzinie czasu, następuje ich przekierowanie do funkcji korzystającej z zaimplementowanego modelu *SSN*. Jeżeli analizie poddane mają być dane w innej postaci (np. częstotliwościowej), najpierw przeprowadzane są niezbędne do wykonania takiej transformacji operacje. W zależności od wybranej metody, można zastosować funkcję okna, przetwarzanie FFT oraz wybór konkretnego spektrum widma. Zastosowanie poszczególnych etapów przetwarzania oraz konkretnych ich parametrów, jak np. liczbę przetwarzanych próbek, rodzaj funkcji okna czy szerokość widma, należy wybrać na etapie przygotowywania programu. Powinny być one zgodne z parametrami, których użyto do wytrenowania zaimplementowanego modelu klasyfikatora.

Rysunek 6.4 przedstawia w sposób schematyczny strukturę bufora cyklicznego o 8 komórkach pamięci. Strzałka wskazuje kierunek zmiany miejsca zapisu kolejnego pomiaru. Jeżeli liczba przetwarzanych próbek została ustalona na 8 pomiarów, 7 pierwszych klasyfikacji może dawać nieprawidłowe wartości z powodu niepełnego bufora danych. Od 9. pomiaru najstarsze dane o indeksie $x(n - 7)$ zastępowane są w buforze wartościami najnowszymi, które zawsze otrzymują indeks $x(n)$. Indeksy kolejnych komórek są dekrementowane (np. z $x(n)$ do $x(n - 1)$, a z $x(n - 1)$ do $x(n - 2)$), ale ich zawartość pozostaje bez zmian. Nadpisywane są zawsze tylko komórki o indeksie $x(n - 7)$, a cykl powtarzany jest w nieskończonej pętli (chyba że wcześniej zdefiniowano liczbę przebiegów). W taki sposób w buforze dostępnych jest zawsze 8 ostatnich pomiarów.

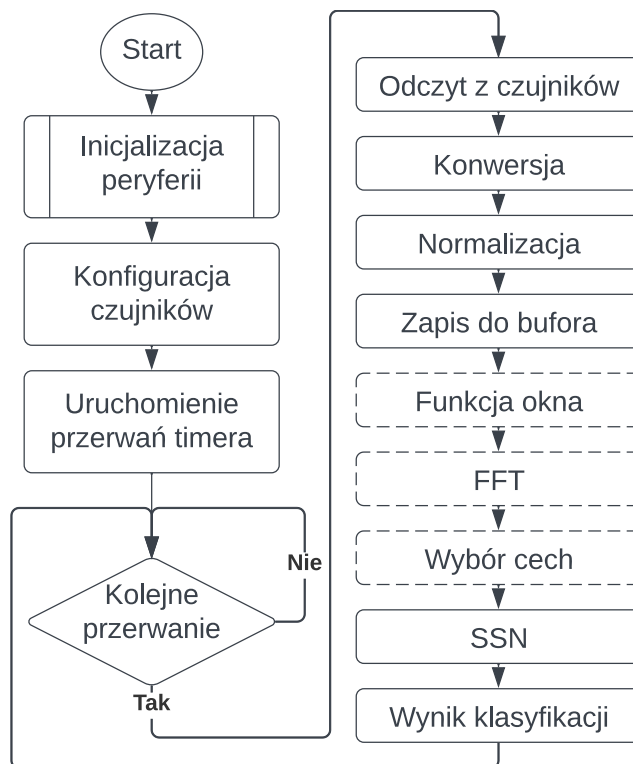
W przypadku systemu niedziałającego w czasie rzeczywistym, najpierw pobierane są wszystkie próbki wymagane do zapełnienia bufora, a następnie przeprowadzana jest diagnostyka. Po jej zakończeniu układ dokonuje akwizycji kolejnej porcji danych z zadaną szybkością próbkowania i proces jest powtarzany. Jednak pomiędzy uzyskaniem wyniku klasyfikacji a wykonaniem kolejnego próbkowania mija czas dłuższy od okresu próbkowania, stąd nie

można mówić o przetwarzaniu w czasie rzeczywistym. Funkcja AI przypisuje wartości poszczególnym klasom uszkodzeń. Klasa, która otrzymała najwyższą wartość liczbowa, uznawana jest za wskazaną przez klasyfikator jako klasa odpowiadająca występującemu uszkodzeniu. Zatem numer takiej klasy odpowiada wynikowi przeprowadzonego wnioskowania.



Rys. 6.4: Schematyczna budowa bufora cyklicznego

Na rysunku 6.5 przedstawiono kolejność zasadniczych operacji wykonywanych podczas działania programu do detekcji i klasyfikacji uszkodzeń w czasie rzeczywistym. Pomimo procesu na pozór dużo prostszego od zadania akwizycji danych (rysunek 4.2), w rzeczywistości realizowanych jest tutaj wiele innych kroków. Między innymi wykorzystywana jest karta pamięci, na której zapisywane są wyniki kolejnych klasyfikacji i czas przetwarzania w poszczególnych etapach działania układu. Wyniki przetwarzania są na bieżąco reprezentowane za pomocą diod elektroluminescencyjnych, ale z powodu dużej szybkości działania systemu (500 klasyfikacji na sekundę), nie jest możliwe zarejestrowanie w ten sposób wszystkich wyników klasyfikacji jedynie za pomocą wzroku. Dlatego są one utrwalane na karcie pamięci, co umożliwia późniejszą ich analizę.



Rys. 6.5: Tryb detekcji (opracowanie na podstawie [103])

Testy skuteczności klasyfikatorów

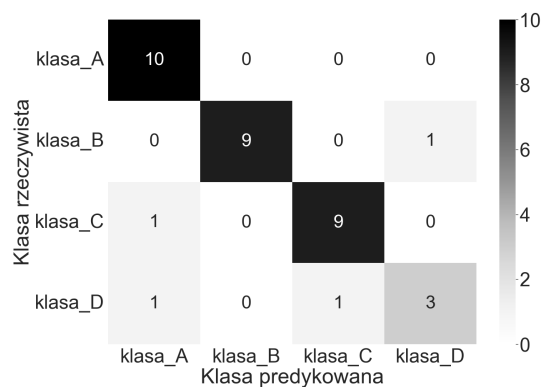
W niniejszym rozdziale przedstawiono wyniki testów walidacyjnych różnych modeli klasyfikatorów, które zostały przygotowane z wykorzystaniem pomiarów zebranych za pomocą systemu akwizycji danych sensorycznych opisanego w rozdziale 4. Testy dotyczą przede wszystkim modeli czasu rzeczywistego, ale omówione zostały także niektóre klasyfikatory, które nie spełniają tego wymogu. W celu dokonania oceny jakości prezentowanych modeli przyjęto jednolity sposób prezentacji wybranych parametrów, ze szczególnym uwzględnieniem określonych metryk, umożliwiających obiektywną analizę każdego klasyfikatora.

7.1 Wskaźniki oceny jakości klasyfikatorów

Aby można było cokolwiek powiedzieć o zdolności klasyfikatora do realizacji postawionego zadania, należy użyć konkretnej miary. Szczególnie w przypadku konieczności porównania dwóch lub większej liczby klasyfikatorów albo w celu stworzenia ich rankingu, potrzebna jest konkretna metryka, która oceni ich jakość dla wybranego kryterium. Przyjmuje się, że dobry klasyfikator to taki, który poprawnie przypisuje badane przypadki do odpowiednich – zwykle z góry zdefiniowanych – klas. Rozumieć należy to jako popełnianie jak najmniejszej liczby błędów w klasyfikacji. W tym przypadku intuicja nie zawsze w pełni odzwierciedla rzeczywistość. W zależności bowiem od rodzaju dokonanego podziału na poszczególne klasy, zupełnie inaczej oceniony zostać może klasyfikator poprawnie wskazujący jedną klasę i popełniający błędy przy drugiej, od klasyfikatora wykazującego dużą dokładność dla klasy drugiej, a popełniającego błędy przy pierwszej. Z tego powodu powstało wiele różnych miar umożliwiających jakościową ocenę konkretnych modeli dokonujących przyporządkowania analizowanych przypadków do zdefiniowanych klas.

Najczęstszym punktem wyjścia do tworzenia konkretnych metryk jakościowych jest tzw. macierz błędów (macierz pomyłek, tablica pomyłek). Pokazuje ona dokładnie, jak wiele przypadków konkretnej klasy zostało przy-

pisanych do klasy prawdziwej, a jak wiele do poszczególnych klas błędnych. Najprostszym rodzajem klasyfikacji jest klasyfikacja binarna, w której występują tylko dwie klasy – najczęściej nazywane pozytywną i negatywną. Ponieważ tematyka rozprawy oprócz wykrycia samego uszkodzenia jako takiego obejmuje również rozpoznawanie umiejscowienia, rodzaju i rozmiaru usterki, wszelkie przeprowadzane eksperymenty miały charakter wieloklasowy. Zatem omówione zostaną tutaj te wskaźniki, które można zastosować do większej liczby klas. Macierz błędów zazwyczaj ma kształt zbliżony do kwadratu – posiada tyle samo kolumn, co wierszy. Spotyka się zarówno konstrukcje przedstawiające klasy rzeczywiste poziomo, a klasy predykowane (prognozowane, rozpoznawane przez klasyfikator) pionowo, jak i w sposób przeciwny. W całej rozprawie stosowana będzie forma, w której klasy rzeczywiste ułożone są wierszami jedna pod drugą, a klasy predykowane stanowią kolejne kolumny. Wygląd przykładowej macierzy błędów przedstawiono na rysunku 7.1.



Rys. 7.1: Przykładowa macierz błędów dla klasyfikatora czteroklasowego

Główna przekątna biegnąca z lewego górnego do prawego dolnego rogu wskazuje poprawnie sklasyfikowane przypadki. To ona zwykle będzie wyróżniona wyraźnie większymi wartościami liczbowymi (lub procentowymi). W rozprawie stosowane jest także ciemniejsze barwienie pól o największych wartościach, co pozwala na szybką ocenę klasyfikatora po kontraście występującym między główną przekątną a pozostałymi polami macierzy.

Dzięki takiej formie reprezentacji poszczególnych klasyfikacji, można szybko zauważyć, że testowy klasyfikator bezbłędnie rozpoznał klasę oznaczoną literą *A*, po jednym błędzie popełnił klasyfikując przypadki *B* i *C*, natomiast klasę *D* dwukrotnie zinterpretował niewłaściwie. Macierz taka pozwoli także wyznaczyć precyzyjniejsze metryki. Do najbardziej popularnych należą:

- **Dokładność** (*accuracy*) to stosunek poprawnie sklasyfikowanych próbek do wszystkich próbek.

$$\text{accuracy} = \frac{\sum_{i=1}^N TP_i}{N}, \quad (7.1)$$

gdzie TP_i to liczba poprawnie sklasyfikowanych próbek klasy i , natomiast N to łączna liczba wszystkich próbek.

- **Precyzja** (*precision*) dla każdej klasy określana jest jako stosunek liczby poprawnie przewidzianych próbek danej klasy do liczby wszystkich próbek, które zostały przewidziane jako ta klasa.

$$\text{precision}_i = \frac{TP_i}{TP_i + FP_i}, \quad (7.2)$$

gdzie TP_i to liczba prawdziwie pozytywnych próbek klasy i , z kolei FP_i to liczba próbek błędnie przyporządkowanych do klasy i .

Dla modeli wieloklasowych stosuje się tzw. *macro average precision* do obliczenia średniej arytmetycznej precyzji ze wszystkich klas:

$$\text{macro average precision} = \frac{1}{K} \sum_{i=1}^K \text{precision}_i, \quad (7.3)$$

micro average precision, uwzględniające globalną wartość precyzji całego modelu:

$$\text{micro average precision} = \frac{\sum_{i=1}^K TP_i}{\sum_{i=1}^K (TP_i + FP_i)}, \quad (7.4)$$

a także *weighted precision*, w której każda klasa wpływa na wynik w stopniu proporcjonalnym do liczby próbek w tej klasie:

$$\text{weighted precision} = \sum_{i=1}^K \frac{n_i}{N} \cdot \text{precision}_i, \quad (7.5)$$

gdzie K to liczba klas, a n_i to liczba próbek w klasie i .

Warto zauważyć, że w klasyfikacji wieloklasowej, gdzie wszystkie błędne predykcje FP (liczba przewidywań fałszywie pozytywnych, ang. *False Positive*) dla jednej klasy stają się jednocześnie FN (liczba przewidywań fałszywie negatywnych, ang. *False Negative*) dla innej klasy, *micro average precision* jest tym samym, co *accuracy*.

- **Czułość** (*recall*) to stosunek liczby poprawnie przewidzianych próbek danej klasy do liczby wszystkich próbek rzeczywiście należących do tej klasy:

$$\text{recall}_i = \frac{TP_i}{TP_i + FN_i}, \quad (7.6)$$

gdzie FN_i to liczba próbek klasy i przypisanych do innej klasy.

Dla modeli wieloklasowych stosuje się tzw. *weighted recall* do obliczenia średniej ważonej czułości ze wszystkich klas:

$$\text{weighted recall} = \sum_{i=1}^K \frac{n_i}{N} \cdot \text{recall}_i. \quad (7.7)$$

- **Metryka F1** (*F1-score*) jest wyrażana jako średnia harmoniczna precyzji i czułości dla każdej klasy:

$$F1_i = 2 \cdot \frac{\text{precision}_i \cdot \text{recall}_i}{\text{precision}_i + \text{recall}_i}. \quad (7.8)$$

Średnia ważona metryki F1 dla klasyfikatora wieloklasowego przyjmuje postać:

$$\text{weighted F1} = \sum_{i=1}^K \frac{n_i}{N} \cdot F1_i. \quad (7.9)$$

- **Współczynnik kappa** (*kappa coefficient*) mierzy zgodność predykcji modelu z rzeczywistymi wartościami z uwzględnieniem zgodności losowej:

$$\kappa = \frac{p_o - p_e}{1 - p_e}, \quad (7.10)$$

gdzie p_o to rzeczywista dokładność (zgodność), natomiast p_e oznacza przewidywaną zgodność dla losowego przypisywania etykiet.

W tabeli 7.1 zestawiono omówione metryki zarówno dla poszczególnych klas, jak i wartości średnie wyliczone różnymi sposobami. Niektóre z nich nie są w praktyce stosowane, ale dla pokazania różnic w wynikach w zależności od stosowanego wzoru oraz zależności pomiędzy poszczególnymi parametrami, uwzględniono wszystkie 35 wartości.

Najczęściej występującą wartością jest 0,8857, która odpowiada m.in. ogólnej (globalnej) dokładności. Jest to najbardziej intuicyjna wielkość, którą wylicza się wg wzoru (7.1), czyli po prostu dzieląc liczbę poprawnych pre-

Tab. 7.1: Popularne metryki dla przykładowej macierzy błędów z rysunku 7.1.

	Dokładność	Precyzja	Czułość	F1	Kappa
klasa_A	0,9429	0,8333	1,0000	0,9091	0,8679
klasa_B	0,9714	1,0000	0,9000	0,9474	0,9278
klasa_C	0,9429	0,9000	0,9000	0,9000	0,8600
klasa_D	0,9143	0,7500	0,6000	0,6667	0,6182
macro avg	0,9429	0,8708	0,8500	0,8558	0,8185
micro avg	0,8857	0,8857	0,8857	0,8857	0,8436
weighted	0,9470	0,8881	0,8857	0,8828	0,8471

dykcji przez liczbę wszystkich predykcji. I to właśnie metryka *dokładność* będzie stanowić podstawę do oceny zdolności klasyfikacji poszczególnych modeli uwzględniającej jedynie poprawność przypisywania badanych przypadków do właściwej klasy. W tabelach prezentujących wyniki uzyskiwane przez testowane modele wartości dokładności podawane są z użyciem zapisu liczbowego, w celu ujednolicenia z innymi metrykami. Natomiast w tekście zwykle stosowana jest wartość procentowa, jako najczęściej spotykany sposób przedstawiania tego wskaźnika.

Oprócz wyżej wymienionych, spotkać można się także z używaniem innych metryk. Zwykle są one stosowane do klasyfikacji binarnej, ale stosując techniki typu *one-vs-rest* [78] lub obliczając wartości osobno dla każdej klasy, a następnie je uśredniając, można posłużyć się nimi także w przypadku modeli wieloklasowych. Do takich metryk należą między innymi: *specyficzność* [84], *AUC-ROC* [137], *logarytmiczna funkcja straty* [28], *top-k accuracy* [77], *MCC* [21], *FMI* [20], czy *indeks Jaccarda* [142]. Każdy z tych wskaźników daje nieco inną perspektywę na jakość modelu, pomagając w różny sposób spojrzeć na efektywność oraz ograniczenia modelu. Ważnym jest zatem dobranie takich metryk, które w najlepszy możliwy sposób będą odpowiadały wymogom zadania realizowanego przez konkretny klasyfikator.

7.1.1 Ocena klasyfikacji niezależnej od czasu przetwarzania

Wszystkie opisane dotychczas wskaźniki wykorzystują konkretną formułę matematyczną, która uwzględnia tylko jeden rodzaj zmiennych – poprawność przypisywania badanych wzorców do konkretnych klas. Co prawda wzory wyliczające wartości średnie zawierają także liczbę klas, w ramach których operuje model, ale wielkość ta nie wpływa w sposób arbitralny na uzyskane wyniki. Przykładowo model binarny popełniający błąd klasyfikacji w co drugim przypadku uzyska taką samą wartość dokładności, co model przypisujący 20 różnych etykiet analizowanym danym, jeśli jest nieomylny w co drugim przypadku. Wiele innych metryk również będzie wskazywać podobną efektywność obu modeli, mimo że zadanie rozróżnienia 2 klas jest dużo prostszym problemem niż rozróżnienie 20 klas. Co prawda bardzo trudno byłoby porównywać modele klasyfikujące zupełnie różne wzorce, np. statyczne obrazy z odczytami z czujników inercyjnych, ale jeśli dane wejściowe są podobnego typu, można zestawiać ze sobą różne klasyfikatory. W przypadku rozpoznawania uszkodzeń tworzone są modele przeprowadzające prostą binarną detekcję, których zadaniem jest wykrycie, że usterka wystąpiła. Bardziej złożone modele zdolne są do wskazania miejsca wystąpienia awarii, jej rozmiaru lub nawet jednego z wielu rodzajów prawdopodobnych anomalii. Aby możliwe było bezpośrednie porównanie takich klasyfikatorów, zaproponowano nową metrykę – Q_{AC} [103]. Formułę do wyznaczania jej wartości przedstawia wzór (7.11):

$$Q_{AC} = A^\alpha \sqrt[\alpha]{\ln C}, \quad (7.11)$$

gdzie A to dokładność modelu wyliczana zgodnie z (7.1), z kolei C to liczba klas, na których operuje model. We wzorze znajduje się także parametr α , który reguluje, jak duży wpływ na wartość Q_{AC} mają wielkości A i C . Im większy parametr α , tym wskaźnik jest bardziej wrażliwy na zmiany dokładności, a mniej wrażliwy na liczbę klas. Aby zachować racjonalny balans między tymi dwiema składowymi, w zadaniach rozpoznawania usterek drona proponuje się stosowanie $\alpha = 3$. Wtedy wzór (7.11) przyjmuje postać $Q_{AC} = A^3 \sqrt[3]{\ln C}$. Natomiast przy porównywaniu różnych klasyfikatorów najistotniejsze jest konsekwentne używanie tej samej wartości α . W tabeli 7.2 zestawiono wartości wskaźnika Q_{AC} dla różnych parametrów dokładności oraz liczby rozpatrywanych klas. Przyjęto wartość parametru α równą 3.

Tab. 7.2: Wartości wskaźnika Q_{AC} dla $\alpha = 3$ oraz zmiennych parametrów A i C .

A	C	Q_{AC}	A	C	Q_{AC}
0,99	6	1,18	0,96	2	0,78
0,97	6	1,11	0,96	4	0,99
0,95	6	1,04	0,96	8	1,13
0,93	6	0,98	0,96	16	1,24
0,91	6	0,92	0,96	32	1,34
0,85	6	0,75	0,96	64	1,42
0,80	6	0,62	0,96	128	1,50
0,70	6	0,42	0,96	256	1,57

Zwykle wskaźnik Q_{AC} przyjmuje wartości z przedziału od 0 do ok. 1, 5. Większe wartości można uzyskać, ale nie są spotykane w praktyce. Na przykład wartość 2 zostanie osiągnięta dla 100% dokładności i liczby klas ok. 2981 (ponieważ $e^8 \approx 2980,96$). Dla sześcioklasowego problemu wskaźnik Q_{AC} przyjmuje wartości większe od 1.0 w modelach, których dokładność wynosi min. 94,8%. Z kolei dla modelu o dokładności 96%, Q_{AC} uzyskuje poziom 1 przy liczbie klas wynoszącej 5 lub więcej. Można przyjąć, że modele o wskaźniku Q_{AC} większym od 1 są dobrymi klasyfikatorami. Jednak uzyskany bezwzględny wynik nie stanowi definitywnej oceny konkretnego modelu. Należy go rozpatrywać w kontekście rozważanego problemu, w szczególności porównując z wynikami uzyskanymi przez inne modele używane w podobnych zadaniach, zwłaszcza gdy różnią się między sobą liczbą rozpoznawanych klas.

Opisywany wskaźnik opracowany został w odpowiedzi na potrzebę porównania różnych klasyfikatorów uszkodzeń śmigieł dronów, które w przeprowadzanych badaniach analizowały problemy złożone zarówno z dwóch, jak i kilkunastu lub nawet kilkudziesięciu klas. Porównanie takich modeli klasycznymi wskaźnikami nie daje wystarczającej informacji o przydatności tych klasyfikatorów w konkretnych problemach.

7.1.2 Ocena klasyfikacji z krytycznym czasem przetwarzania

Podobną niedogodność napotyka się, chcąc porównać ze sobą klasyfikatory pracujące w systemach, w których ważną rolę odgrywa czas przetwarzania. O ile dla modeli dokonujących obliczeń offline nie ma większego

znaczenia, czy wynik uzyskano w 1 ms czy w 20 s, o tyle w systemach bezpieczeństwa pojazdów latających opóźnienie między wystąpieniem usterki a jej detekcją jest krytyczne. Szybka diagnoza może wykryć uszkodzenie na wstępnych etapach degradacji podzespołów, a także umożliwić bezzwłoczną reakcję, np. w postaci awaryjnego lądowania, co może uchronić BSP przed większymi uszkodzeniami spowodowanymi ewentualnym rozbiem. Skoro czas wykrycia nieprawidłowości jest tak istotny, warto go uwzględnić w ocenie jakości klasyfikatorów używanych w tego typu systemach. Wydawać by się mogło, że podobną rolę spełniałoby rozpatrzenie złożoności obliczeniowej wszystkich operacji niezbędnych do uzyskania wyniku klasyfikacji, ale nie byłaby to pełna informacja. Dopiero uwzględnienie sumarycznego czasu przeznaczonego na każdy etap biorący udział w operacjach – od pozyskania niezbędnych danych, poprzez ich przetwarzanie aż po właściwe zaklasyfikowanie – daje rzeczywisty obraz opóźnienia. Ma na to wpływ zastosowany sprzęt, użyte oprogramowanie i techniki przetwarzania. Ze względu na ograniczenia udźwigu i wydajności zasilania konkretnej klasy BSP, problem należy rozpatrywać, biorąc pod uwagę rzeczywiste możliwości zastosowania konkretnej metody przetwarzania. Dlatego posłużenie się rzeczywistym czasem opóźnienia w klasyfikacji usterek jest bardziej praktyczne niż korzystanie ze złożoności obliczeniowej jako parametru. W związku z tym zaleca się korzystanie ze wskaźnika Q_{ACT} [103], który pozwala na porównanie klasyfikatorów pod kątem dokładności klasyfikacji, liczby klas oraz czasu przetwarzania. Wzór (7.12) umożliwia wyznaczenie wartości tego wskaźnika:

$$Q_{ACT} = A^\alpha \sqrt[\alpha]{\frac{\ln C}{\ln(1 + \tau^2)}}, \quad (7.12)$$

gdzie τ (7.13) jest bezwymiarowym parametrem zależnym od rzeczywistego czasu całego procesu uzyskania pojedynczej klasyfikacji T wyskalowanego w milisekundach oraz czasu odniesienia T_0 , również wyrażonego w milisekundach:

$$\tau = \frac{T}{T_0}. \quad (7.13)$$

Proponuje się przyjęcie za T_0 okresu próbkowania danych, czyli, dla przykładu, przy f_s równym 1 kHz, czas odniesienia T_0 wyniesie 1 ms. Dla eksperymentów opisanych w rozdziałach 4 i 6 będą to 2 ms, ponieważ w tych przypadkach częstotliwość akwizycji pomiarów z czujników wynosi 500 Hz. Pomaga to w szybkim określeniu, czy system spełnia wymogi czasu rzeczywistego według definicji z podrozdziału 1.4. W przypadku porównywania klasyfikatorów pracujących z różnymi okresami próbkowania, należy przyjąć

jednakową wartość T_0 . Zaleca się użycie największej wartości ze wszystkich porównywanych modeli. W tabeli 7.3 zestawiono wartości wskaźnika Q_{ACT} dla różnych parametrów dokładności, liczby rozpatrywanych klas oraz czasu przetwarzania. Przyjęto wartość parametru α równą 3, natomiast T_0 wynosi 2 ms.

Tab. 7.3: Wartości wskaźnika Q_{ACT} dla $\alpha = 3$, $T_0 = 2$ ms oraz zmiennych parametrów A , C i T .

A	C	T	Q_{ACT}	A	C	T	Q_{ACT}	A	C	T	Q_{ACT}
0,99	6	2	1,33	0,96	2	2	0,88	0,96	6	0,5	2,74
0,97	6	2	1,25	0,96	4	2	1,11	0,96	6	1	1,77
0,95	6	2	1,18	0,96	8	2	1,28	0,96	6	2	1,21
0,93	6	2	1,10	0,96	16	2	1,40	0,96	6	4	0,92
0,91	6	2	1,03	0,96	32	2	1,51	0,96	6	10	0,72
0,85	6	2	0,84	0,96	64	2	1,61	0,96	6	100	0,54
0,80	6	2	0,70	0,96	128	2	1,69	0,96	6	1000	0,46
0,70	6	2	0,47	0,96	256	2	1,77	0,96	6	10000	0,42

Podobnie jak w przypadku wskaźnika Q_{AC} , tak i dla oceny jakości modeli, w których czas przetwarzania odgrywa istotną rolę, można przyjąć szacunkowo, że dobre klasyfikatory uzyskują Q_{ACT} na poziomie wyższym od 1, jednakże szczegółowa interpretacja wyniku zależy będzie od potrzeb konkretnego układu, w którym dany model będzie realizował zadanie klasyfikacji. Wskaźnik Q_{ACT} jest jednak jak najbardziej odpowiednim narzędziem do porównywania różnych klasyfikatorów przeznaczonych do podobnych zadań, w tym do rozpoznawania usterek śmigieł drona podczas lotu.

7.2 Sposób przeprowadzania eksperymentów

Na etapie przygotowywania klasyfikatora uszkodzeń należało poczynić wstępne założenia dotyczące celu, jaki dany klasyfikator miał osiągnąć. W początkowych etapach badań głównym kryterium było uzyskanie jak najwyższej dokładności przy możliwie jak najmniejszej liczbie fałszywych alarmów. Najważniejszym parametrem była więc globalna dokładność klasyfikacji wszystkich badanych klas uszkodzeń, w dalszej kolejności skuteczność wykrywania klasy zdrowej i jak najwierniejsze rozpoznawanie klasy uszkodzonej. Choć w rozprawie wprost podawany jest wynik kategoryza-

cji wieloklasowej przeprowadzanej w ramach FDI lub FDD (w zależności od rodzajów uszkodzeń), to w pewnym stopniu niejawnie brano pod uwagę także ogólniejszy problem FD. Jeżeli dwa modele uzyskiwały bardzo podobną dokładność (różnica mniejsza niż 0,1 p.p.), za skuteczniejszy był uznawany ten, który osiągał lepszy wynik podziału klas na 2 grupy: *system sprawny* oraz *system uszkodzony*. Gdy jasne stało się, że stosowane metody z powodzeniem można aplikować do systemów pracujących w czasie rzeczywistym, zaczęto stosować także kolejne kryterium – całkowity czas przetwarzania w zastosowanym mikrokontrolerze. Z tego względu nie dla wszystkich modeli ten parametr był mierzony. Jeżeli czas przetwarzania oraz wskaźnik Q_{ACT} nie są podane, oznacza to, że model nie nadaje się do pracy w czasie rzeczywistym. Niezależnie od faktu, czy wielkości te były określane, w tym rozdziale nie wskazywano ich, jeżeli przetwarzanie sygnałów z zastosowaniem bufora cyklicznego zajmowało więcej niż 2 ms.

Ponieważ w repozytorium PADRE, oprócz samych danych zarejestrowanych przez czujniki pomiarowe, znalazły się także dane wstępnie przetworzone, w wielu przypadkach wystarczyło wprost posłużyć się nimi. Jeżeli brakowało jakiegoś rodzaju cech lub potrzebnej kombinacji osi pomiarowych, generowano je w programie *MATLAB*.

Wszystkie modele w pierwszej kolejności poddawane były testom offline z zastosowaniem 15% danych wydzielonych na ten właśnie proces. W testach zawsze wykorzystywano nowe, nieznane wcześniej modelowi dane. Nie brały one udziału we wcześniejszych etapach przygotowywania klasyfikatora. Następnie model był implementowany w mikrokontrolerze. Testy z dronem *Bebop 2* przeprowadzano w laboratorium, a z dronem *Solo* w otwartej przestrzeni miejskiej. Eksperymenty walidacyjne wykonywano w miejscu, w którym pobierane były dane treningowe. Nie testowano w warunkach zewnętrznych modeli trenowanych z użyciem danych z laboratorium, a w laboratorium nie sprawdzano skuteczności modeli przygotowanych na bazie danych zebranych w naturalnych warunkach atmosferycznych. Modele użytych sensorów oraz ich zakresy pomiarowe odpowiadały tym z etapu zbierania danych. Natomiast konkretne egzemplarze czujników lub miejsce ich zamontowania były różne. Podobnie jak w przypadku akwizycji danych, nie przeprowadzano procesu ich kalibracji.

W kolejnych podrozdziałach zamieszczono wyniki uzyskane podczas przeprowadzonych eksperymentów. Posłużono się tabelami, w których wy-

pisano najważniejsze parametry i metryki oraz macierzami błędów, które pozwalają w obrazowy sposób ocenić precyzję testowanego modelu. Choć liczba omawianych modeli może wydawać się spora, to przedstawione wyniki dotyczą tylko klasyfikatorów uznanych za najlepsze w kilku różnych schematach. Eksperymenty podzielono na kilka grup według następujących kryteriów: model BSP, rodzaj uszkodzeń, dziedzina cech sygnałowych, reżim pracy ze względu na czas przetwarzania.

7.3 Sygnały wyrażone w dziedzinie czasu

Akwizycja danych z czujników przeprowadzana była w sposób klasyczny. Dokonano równomiernego próbkowania sygnału, odczytując wartości z rejestrów czujników w stałych odstępach czasu. Zużycie lub uszkodzenie śmigieł BSP powinno wpływać na wartość tych odczytów. Naturalnym pomysłem na przeprowadzenie wnioskowania o stanie wirników nośnych jest zatem bezpośrednie wykorzystanie tego typu sygnałów. Ponieważ jednak pojedynczy odczyt nie niesie za sobą informacji o zmianach wartości sygnału w czasie, konieczne było dokonanie pewnych modyfikacji. Postanowiono odczyty jednego rodzaju pogrupować w pakiety. Liczba takich następujących po sobie pomiarów z jednej osi sensora określana będzie mianem szerokości lub długości okna pomiarowego. Załóżmy, że do dyspozycji są odczyty z osi x oraz osi y akcelerometrów umieszczonych na 4 ramionach drona. Pierwszych 8 pomiarów wyglądać będzie następująco:

1. A_{ax} , A_{ay} , B_{ax} , B_{ay} , C_{ax} , C_{ay} , D_{ax} , D_{ay}
2. A_{ax} , A_{ay} , B_{ax} , B_{ay} , C_{ax} , C_{ay} , D_{ax} , D_{ay}
3. A_{ax} , A_{ay} , B_{ax} , B_{ay} , C_{ax} , C_{ay} , D_{ax} , D_{ay}
4. A_{ax} , A_{ay} , B_{ax} , B_{ay} , C_{ax} , C_{ay} , D_{ax} , D_{ay}
5. A_{ax} , A_{ay} , B_{ax} , B_{ay} , C_{ax} , C_{ay} , D_{ax} , D_{ay}
6. A_{ax} , A_{ay} , B_{ax} , B_{ay} , C_{ax} , C_{ay} , D_{ax} , D_{ay}
7. A_{ax} , A_{ay} , B_{ax} , B_{ay} , C_{ax} , C_{ay} , D_{ax} , D_{ay}
8. A_{ax} , A_{ay} , B_{ax} , B_{ay} , C_{ax} , C_{ay} , D_{ax} , D_{ay} .

W celu przetworzenia danych do użycia na potrzeby przygotowania klasyfikatora uszkodzeń, zostały one posortowane w taki sposób, by kolejne odczyty z danej osi czujnika znajdowały się obok siebie. Jeśli szerokość okna pomiarowego wynosi 4, nowa struktura odczytów ma taką postać:

A_ax1 , A_ax2 , A_ax3 , A_ax4 , A_ay1 , A_ay2 , A_ay3 , A_ay4 ,
B_ax1 , B_ax2 , B_ax3 , B_ax4 , B_ay1 , B_ay2 , B_ay3 , B_ay4 ,
C_ax1 , C_ax2 , C_ax3 , C_ax4 , C_ay1 , C_ay2 , C_ay3 , C_ay4 ,
D_ax1 , D_ax2 , D_ax3 , D_ax4 , D_ay1 , D_ay2 , D_ay3 , D_ay4 .

A_ax5 , A_ax6 , A_ax7 , A_ax8 , A_ay5 , A_ay6 , A_ay7 , A_ay8 ,
B_ax5 , B_ax6 , B_ax7 , B_ax8 , B_ay5 , B_ay6 , B_ay7 , B_ay8 ,
C_ax5 , C_ax6 , C_ax7 , C_ax8 , C_ay5 , C_ay6 , C_ay7 , C_ay8 ,
D_ax5 , D_ax6 , D_ax7 , D_ax8 , D_ay5 , D_ay6 , D_ay7 , D_ay8 .

W ten sposób uzyskano 2 niezależne wiersze danych, każdy zawierający po 4 kolejne pomiary z poszczególnych osi rozpatrywanych czujników (w raporcie, z uwagi na ograniczoną szerokość strony, każdy wiersz zajmuje 4 linie tekstu, natomiast w pliku CSV jest to 1 linia). W podobny sposób przygotowywane są dane dla każdej klasy uszkodzeń. Dzieliąc cały plik zawierający znormalizowane wartości na takie pakiety, uzyskuje się dane gotowe do trenowania modelu SSN.

Przeprowadzonych zostało wiele eksperymentów z użyciem danych z repozytorium PADRE zgodnie z procedurą opisaną w rozdziale 6. Sprawdzano zarówno detekcję przeprowadzaną w czasie rzeczywistym, jak i realizującą klasyfikację usterek w sposób klasyczny. Diagnozowano wszystkie badane rodzaje uszkodzeń oraz mniejsze grupy scenariuszy, np. tylko pojedyncze uszkodzenia albo tylko uszkodzenia jednego rodzaju. Wykorzystywane były różne kombinacje czujników oraz ich poszczególnych osi. W przypadku struktury samej sieci neuronowej, eksperymentalnie dobierano hiperparametry w taki sposób, by uzyskać jak największą wartość metryki *accuracy* oraz jak najmniejszą wartość metryki *loss*. Na etapie przygotowywania modelu dotyczyły one danych walidacyjnych, natomiast gdy model był już wyuczony, oceniano go na podstawie klasyfikacji dokonywanej na danych testowych. I to właśnie te ostatnie dane stanowiły ostateczny sprawdzian dla każdego modelu sieci.

Najprostszym badanym problemem była lokalizacja przeprowadzana dla pięciu klas jednego typu uszkodzeń śmigieł z drona *Parrot Bebop 2*. W eksperymencie użyto danych z lotu, podczas którego wszystkie śmigła były sprawne (typ nr 0) oraz czterech lotów, w czasie których śmigło z wyszczerbioną krawędzią było zamontowane kolejno na każdym ramieniu drona. Ze względu na różne kierunki wirowania poszczególnych rotorów, w rzeczywistości w ba-

daniu brały udział dwa śmigła uszkodzone w taki sam sposób, prawe i lewe. Jednakże w celu uniknięcia niepotrzebnej komplikacji w oznaczeniach, każde z nich było uznawane za śmigło z uszkodzeniem typu 1.

Dane modelu SSN, który uzyskał najwyższą dokładność na zbiorze testowym, przedstawiono w tabeli 7.4. Metryki *precyzja*, *czułość* oraz *F1-score* mają

Tab. 7.4: Parametry modelu dziedziny czasu dla 5 klas z pojedynczymi uszkodzeniami w postaci wyszczerbionej krawędzi śmigła dla szerokości okna wynoszącego 8 próbek.

Parametr	Wartość
model BSP	Parrot Bebop 2
wykorzystane osie akcelerometru	–
wykorzystane osie żyroskopu	XYZ
liczba wykorzystanych osi	12
szerokość okna pomiarowego	8 próbek
typ sieci	jednokierunkowa
rozmiar wsadu	512
liczba warstw ukrytych	1
liczba neuronów w warstwie ukrytej	164
numer epoki	2441
czas przetwarzania	1,13 ms
<i>train_loss</i>	0,0637
<i>train_acc</i>	0,9877
<i>valid_loss</i>	0,0434
<i>valid_acc</i>	0,9926
<i>test_loss</i>	0,0419
dokładność (<i>test_acc</i>)	0,9952
precyzja	0,9952
czułość	0,9952
metryka F1	0,9952
Q_{AC}	1,16
Q_{ACT}	1,77
rozmiar pliku <i>h5</i> z modelem	150 KiB

różne wartości dla poszczególnych klas, co przedstawiono w tabeli 7.5. Jednakże dzięki zastosowaniu równomiernego podziału na poszczególne klasy w ramach zbioru testowego (ale także treningowego i walidacyjnego), zarówno średnie arytmetyczne, jak i średnie ważone tych metryk są praktycznie identyczne z metryką *dokładność*. Różnica tych wartości w przedstawionych modelach wynosi maksymalnie 0,0001, zatem w kolejnych tabelach podawana będzie jedynie wartość dokładności na zbiorze testowym (*test_acc*) jako

Tab. 7.5: Metryki dla poszczególnych klas uszkodzeń modelu dziedziny czasu uwzględniającego 5 klas z pojedynczymi uszkodzeniami w postaci wyszczerbionej krawędzi śmigła dla szerokości okna wynoszącego 8 próbek.

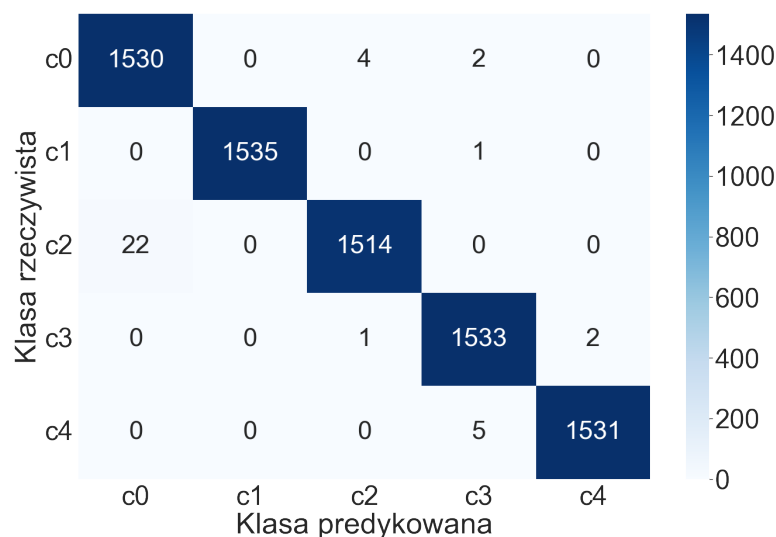
Klasa	Dokładność	Precyzja	Czułość	F1
c0	0,9964	0,9858	0,9961	0,9909
c1	0,9999	1,0000	0,9993	0,9997
c2	0,9965	0,9967	0,9857	0,9912
c3	0,9986	0,9948	0,9980	0,9964
c4	0,9991	0,9987	0,9967	0,9977
średnia	0,9952	0,9952	0,9952	0,9952

powszechnie używany wskaźnik oceny jakości klasyfikatora, a także nowe wskaźniki zaproponowane w podrozdziale 7.1.

Warto zauważyć, jak nieoczywiste wyniki daje tradycyjny sposób liczenia dokładności dla poszczególnych klas. Wyznacza się ją ze wzoru $ACC = (TP + TN)/(TP + TN + FP + FN)$, w którym jako wartości TN (liczba przewidywań prawdziwie negatywnych, ang. *True Negative*) brane są wszystkie predykcje niezwiązane z rozpatrywaną klasą (czyli po prostu wszystkie wartości z macierzy błędu nieznajdujące się ani w wierszu, ani w kolumnie dotyczącej klasy, dla której wyliczana jest dokładność). Zatem obliczanie dokładności globalnej jako średniej arytmetycznej dokładności poszczególnych klas może prowadzić do uzyskania zawyżonego wyniku. Taki sposób wyliczania tej metryki dla problemów wieloklasowych należy uznać za niewłaściwy. W przypadku prac naukowych, w których nie podano w sposób jawny wzoru do wyliczenia podanego wyniku, jest to szczególnie szkodliwe. Może wprowadzać w błąd i dawać złudne wyobrażenie wysokich wyników dokładności uzyskiwanych przez przedstawiany klasyfikator. Dlatego rekomendowanym sposobem wyliczania ogólnej (globalnej, średniej) dokładności jest korzystanie ze wzoru (7.1).

Dokładny rozkład klasyfikacji przeprowadzonej przez omawiany model na zbiorze testowym reprezentuje macierz błędu przedstawiona na rysunku 7.2.

Co ciekawe, niewiele gorsze wyniki osiągnął model, który trenowany był na danych o szerokości okna równej zaledwie 2 próbki z każdej z 3 osi żyroskopu. Jego parametry przedstawia tabela 7.6, a wyniki klasyfikacji – rysunek 7.3. Ponieważ długość wiersza danych wejściowych jest cztery razy



Rys. 7.2: Macierz błędów modelu dziedziny czasu dla 5 klas z pojedynczymi uszkodzeniami w postaci wyszczerbionej krawędzi śmigła dla szerokości okna wynoszącego 8 próbek

krótsza, liczba wierszy zwiększa się czterokrotnie. Stąd bierze się większa liczba przeprowadzonych klasyfikacji.

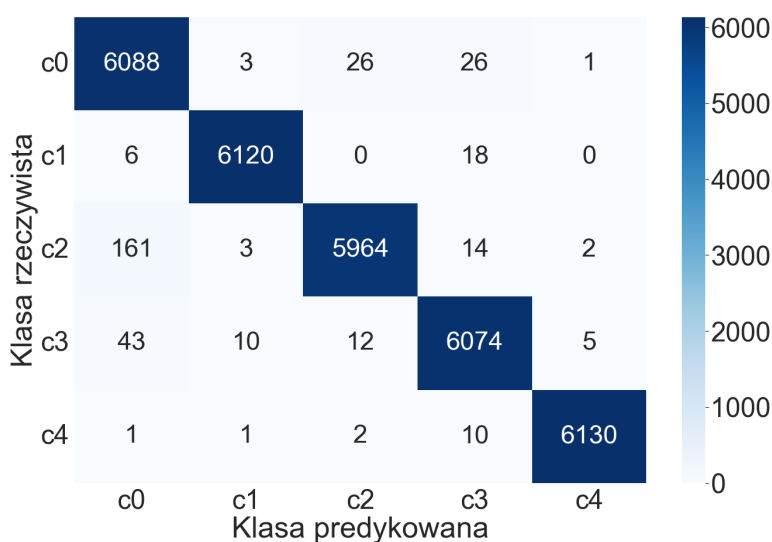
Model ten, dzięki swojej niewielkiej złożoności, podczas klasyfikacji na mikrokontrolerze zrealizował przetwarzanie danych w czasie zaledwie 0,47 ms. Przy zachowaniu wysokiej dokładności pozwoliło to wskaźnikowi Q_{ACT} uzyskać wartość równą 3,00.

Podobne eksperymenty przeprowadzono dla drugiego rodzaju uszkodzeń – wygiętej końcówki śmigła. Własności modelu najlepiej radzącego sobie z problemem klasyfikacji tego typu uszkodzeń przedstawiono w tabeli 7.7. Z kolei wyniki klasyfikacji zobrazowano na rysunku 7.4.

Na podstawie danych z czterowirnikowego *Parrot Bebop 2* sprawdzano skuteczność klasyfikatorów także w lokalizacji i identyfikacji różnych typów uszkodzeń występujących jednocześnie. W mniej złożonym problemie dotyczącym pojedynczych uszkodzeń badano 9 różnych scenariuszy: cztery uszkodzenia pierwszego typu na każdym z ramion, cztery uszkodzenia drugiego typu oraz układ zawierający wszystkie śmigła sprawne. Uzyskana dokładność klasyfikacji wynosi prawie 99%, a przetwarzanie odbywa się w czasie rzeczywistym. W tabeli 7.8 zebrano najważniejsze informacje na temat najlepszego klasyfikatora dla tego problemu, a na rysunku 7.5 zaprezentowano macierz błędów otrzymaną z użyciem tego modelu SSN.

Tab. 7.6: Parametry modelu dziedziny czasu dla 5 klas z pojedynczymi uszkodzeniami w postaci wyszczerbionej krawędzi śmigła dla szerokości okna wynoszącego 2 próbki.

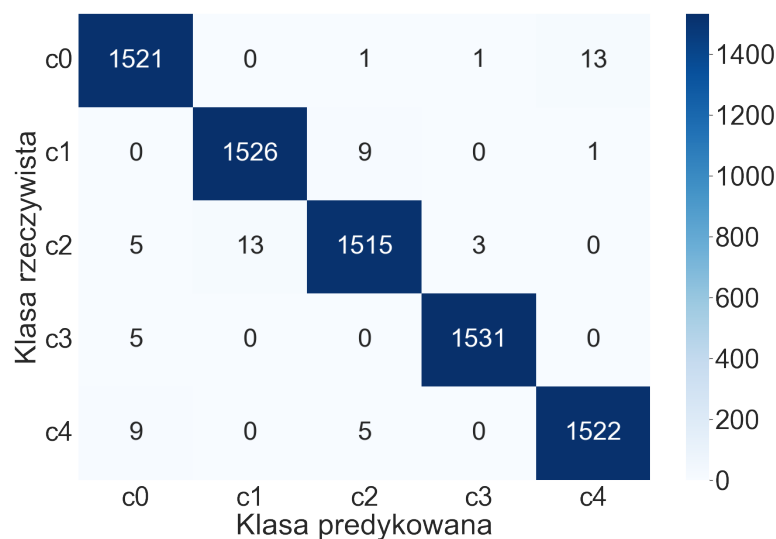
Parametr	Wartość
model BSP	Parrot Bebop 2
wykorzystane osie akcelerometru	–
wykorzystane osie żyroskopu	XYZ
liczba wykorzystanych osi	12
szerokość okna pomiarowego	2 próbki
typ sieci	jednokierunkowa
rozmiar wsadu	512
liczba warstw ukrytych	1
liczba neuronów w warstwie ukrytej	96
numer epoki	3137
czas przetwarzania	0,47 ms
<i>train_loss</i>	0,1291
<i>train_acc</i>	0,9676
<i>valid_loss</i>	0,0690
<i>valid_acc</i>	0,9886
<i>test_loss</i>	0,0643
dokładność (<i>test_acc</i>)	0,9888
Q_{AC}	1,13
Q_{ACT}	3,00
rozmiar pliku <i>h5</i> z modelem	43 KiB



Rys. 7.3: Macierz błędów modelu dziedziny czasu dla 5 klas z pojedynczymi uszkodzeniami w postaci wyszczerbionej krawędzi śmigła dla szerokości okna wynoszącego 2 próbki

Tab. 7.7: Parametry modelu dziedziny czasu dla 5 klas z pojedynczymi uszkodzeniami w postaci wygiętej końcówki śmigła.

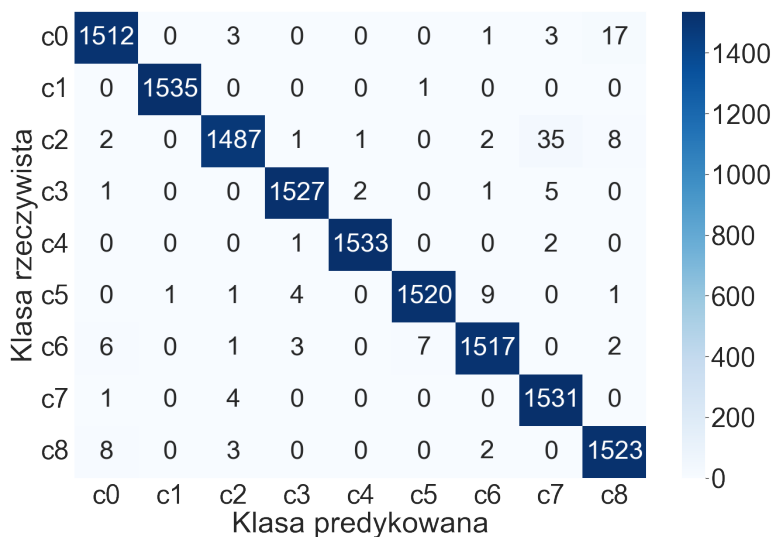
Parametr	Wartość
model BSP	Parrot Bebop 2
wykorzystane osie akcelerometru	–
wykorzystane osie żyroskopu	XYZ
liczba wykorzystanych osi	12
szerokość okna pomiarowego	8 próbek
typ sieci	jednokierunkowa
rozmiar wsadu	512
liczba warstw ukrytych	1
liczba neuronów w warstwie ukrytej	164
numer epoki	4169
czas przetwarzania	1,13 ms
<i>train_loss</i>	0,1994
<i>train_acc</i>	0,9512
<i>valid_loss</i>	0,1067
<i>valid_acc</i>	0,9918
<i>test_loss</i>	0,1036
dokładność (<i>test_acc</i>)	0,9915
Q_{AC}	1,14
Q_{ACT}	1,75
rozmiar pliku <i>h5</i> z modelem	150 KiB



Rys. 7.4: Macierz błędów modelu dziedziny czasu dla 5 klas z pojedynczymi uszkodzeniami w postaci wygiętej końcówki śmigła

Tab. 7.8: Parametry modelu dziedziny czasu dla 9 klas z pojedynczymi uszkodzeniami dwóch rodzajów.

Parametr	Wartość
model BSP	Parrot Bebop 2
wykorzystane osie akcelerometru	–
wykorzystane osie żyroskopu	XYZ
liczba wykorzystanych osi	12
szerokość okna pomiarowego	8 próbek
typ sieci	jednokierunkowa
rozmiar wsadu	512
liczba warstw ukrytych	1
liczba neuronów w warstwie ukrytej	164
numer epoki	4864
czas przetwarzania	1,15 ms
<i>train_loss</i>	0,1874
<i>train_acc</i>	0,9605
<i>valid_loss</i>	0,1018
<i>valid_acc</i>	0,9917
<i>test_loss</i>	0,1073
dokładność (<i>test_acc</i>)	0,9899
Q_{AC}	1,26
Q_{ACT}	1,91
rozmiar pliku <i>h5</i> z modelem	156 KiB



Rys. 7.5: Macierz błędów modelu dziedziny czasu dla 9 klas z pojedynczymi uszkodzeniami dwóch rodzajów

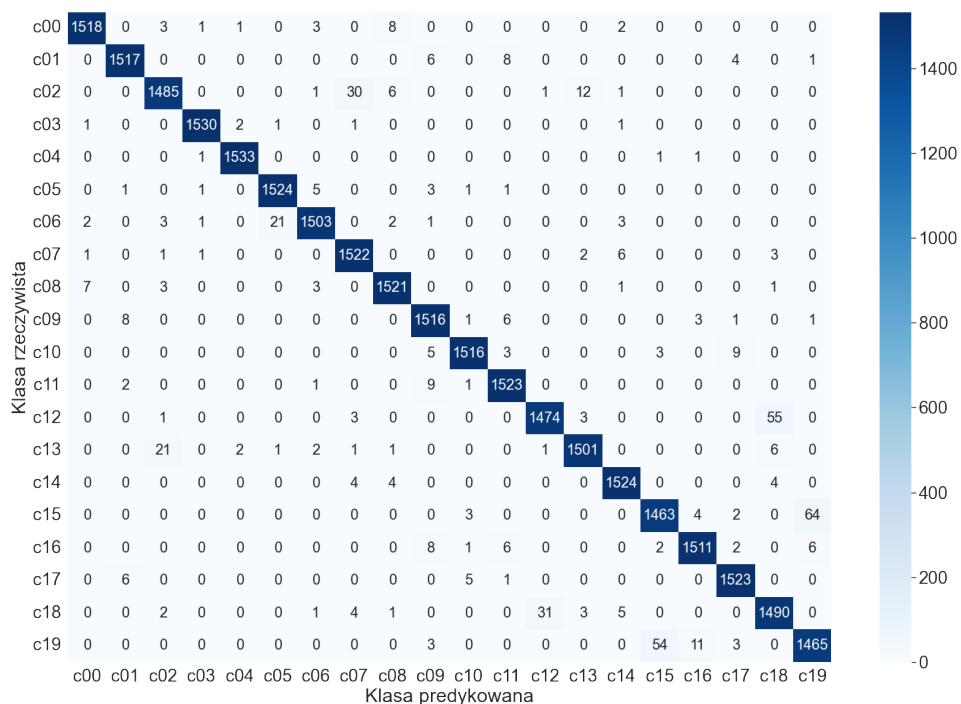
Z kolei najbardziej złożony problem dotyczy różnych kombinacji obu typów nieprawidłowości z różną liczbą uszkodzeń występujących jednocześnie. Pomimo aż 20 różnych klas, najlepszy model osiągnął ponad 98% dokładności. Wszystkie własności modelu przedstawiono w tabeli 7.9, a wyniki poszczególnych klasyfikacji na rysunku 7.6.

Tab. 7.9: Parametry modelu dziedziny czasu dla 20 klas uszkodzeń.

Parametr	Wartość
model BSP	Parrot Bebop 2
wykorzystane osie akcelerometru	–
wykorzystane osie żyroskopu	XYZ
liczba wykorzystanych osi	12
szerokość okna pomiarowego	8 próbek
typ sieci	jednokierunkowa
rozmiar wsadu	2500
liczba warstw ukrytych	1
liczba neuronów w warstwie ukrytej	150
numer epoki	8244
czas przetwarzania	1,17 ms
<i>train_loss</i>	0,3080
<i>train_acc</i>	0,9307
<i>valid_loss</i>	0,1782
<i>valid_acc</i>	0,9813
<i>test_loss</i>	0,1844
dokładność (<i>test_acc</i>)	0,9817
Q_{AC}	1,36
Q_{ACT}	2,05
rozmiar pliku <i>h5</i> z modelem	137 KiB

W przypadku drona *3DR Solo*, testom podlegało 9 scenariuszy, które dotyczyły pojedynczych i podwójnych uszkodzeń. Uszkodzenia to utrata końcówki śmigła o rozmiarze 1 cm lub 2 cm, przy czym rozmiar uszkodzenia również był rozróżniany. Najskuteczniejszy model, którego parametry zestawiono w tabeli 7.10, osiągnął dokładność na poziomie 99,5%, ale z uwagi na zastosowanie okna pomiarowego o szerokości 32 próbek z 3 osi akcelerometru, nie spełnia on wymogów stawianych klasyfikatorom działającym w czasie rzeczywistym. Czas przetwarzania takiej ilości danych jest dłuższy niż okres próbkowania. Macierz błędów tego modelu prezentuje rysunek 7.7.

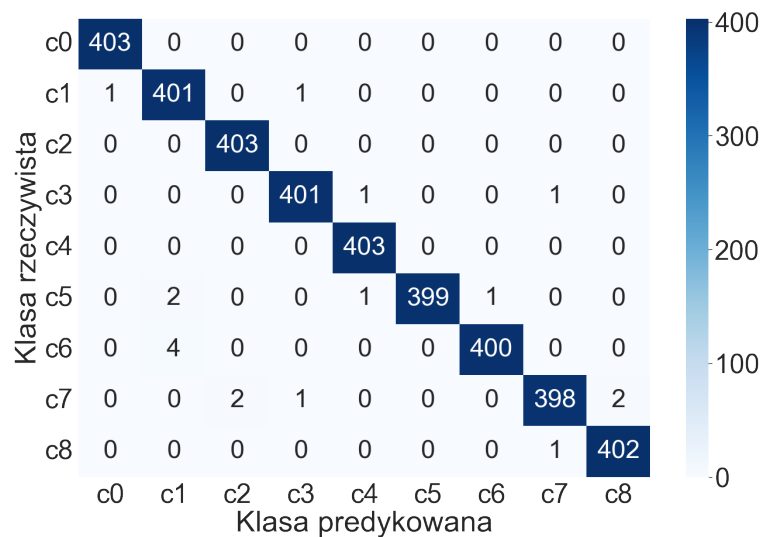
Aby sprostać reżimowi czasu rzeczywistego, zmniejszono szerokość okna pomiarowego do 16 próbek, a sieć konwolucyjną zamieniono na jednokierunkową sieć zbudowaną z pojedynczej, w pełni połączonej warstwy ukrytej (ang.



Rys. 7.6: Macierz błędów modelu dziedziny czasu dla 20 klas uszkodzeń

Tab. 7.10: Parametry modelu dziedziny czasu dla 9 klas uszkodzeń o dwóch rozmiarach.

Parametr	Wartość
model BSP	3DR Solo
wykorzystane osie akcelerometru	XYZ
wykorzystane osie żyroskopu	–
liczba wykorzystanych osi	12
szerokość okna pomiarowego	32 próbki
typ sieci	konwolucyjna
rozmiar wsadu	512
liczba filtrów	3
rozmiar jądra	300
numer epoki	1166
czas przetwarzania	5,13 ms
<i>train_loss</i>	0,0088
<i>train_acc</i>	0,9996
<i>valid_loss</i>	0,0275
<i>valid_acc</i>	0,9928
<i>test_loss</i>	0,0273
dokładność (<i>test_acc</i>)	0,9950
Q_{AC}	1,28
rozmiar pliku h5 z modelem	46 KiB



Rys. 7.7: Macierz błędów modelu dziedziny czasu dla 9 klas uszkodzeń o dwóch rozmiarach

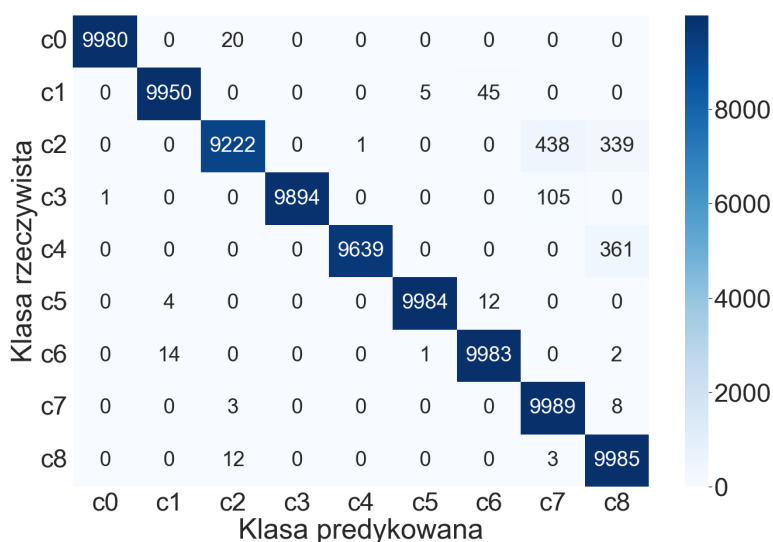
fully connected, dense) [55] o 130 neuronach. Pozwoliło to na skrócenie czasu przetwarzania do 1,42 ms przy obniżeniu dokładności o mniej niż 0,5 punktu procentowego podczas sprawdzianu przeprowadzonego na wydzielonym zbiorze testowym (tabela 7.11).

Ten model został wybrany do przeprowadzenia eksperymentu w rzeczywistych warunkach terenowych. Miejsce testowania było tym samym, w którym dokonano akwizycji danych do repozytorium PADRE. Jednak warunki pogodowe były zgoła inne. O ile akwizycja odbywała się w pogodny, słoneczny wieczór, o tyle testy skuteczności przeprowadzono w pochmurny poranek, podczas którego występował zmienny, chwilami porywisty wiatr, a pomiędzy kolejnymi lotami zdarzały się opady deszczu trwające od kilku do kilkudziesięciu minut. Wykonano 9 lotów odpowiadających 9 różnym scenariuszom uszkodzeń. Podczas każdego lotu trwającego ok. 20 s przeprowadzono 10000 niezależnych klasyfikacji. Pomimo innych warunków atmosferycznych, model przygotowany na wcześniej zebranych danych wykazał dokładność zbliżoną do uzyskanej podczas testów przeprowadzonych na wydzielonym zbiorze. Wyniki klasyfikacji poszczególnych uszkodzeń przedstawia macierz błędów z rysunku 7.8.

Wyraźnie gorszą skuteczność od pozostałych scenariuszy wykazała klasa *c2*, czyli dwucentymetrowe uszkodzenie tylnego lewego śmigła. Ponad 400 razy była rozpoznana przez model jako klasa *c7*, czyli podwójne uszkodzenie

Tab. 7.11: Parametry modelu dziedziny czasu dla 9 klas uszkodzeń o dwóch rozmiarach użytego podczas rzeczywistych testów terenowych.

Parametr	Wartość
model BSP	3DR Solo
wykorzystane osie akcelerometru	XYZ
wykorzystane osie żyroskopu	–
liczba wykorzystanych osi	12
szerokość okna pomiarowego	16 próbek
typ sieci	jednokierunkowa
rozmiar wsadu	1024
liczba warstw ukrytych	1
liczba neuronów w warstwie ukrytej	130
numer epoki	4917
czas przetwarzania	1,42 ms
<i>train_loss</i>	0,0990
<i>train_acc</i>	0,9833
<i>valid_loss</i>	0,0751
<i>valid_acc</i>	0,9919
<i>test_loss</i>	0,0791
dokładność (<i>test_acc</i>)	0,9905
dokładność w teście terenowym	0,9847
Q_{AC}	1,26
Q_{AC} w teście terenowym	1,24
Q_{ACT}	1,70
Q_{ACT} w teście terenowym	1,67
rozmiar pliku <i>h5</i> z modelem	225 KiB



Rys. 7.8: Macierz błędów modelu dziedziny czasu dla 9 klas uszkodzeń o dwóch rozmiarach użytego podczas rzeczywistych testów terenowych

występujące na lewych śmigłach, a ponad 300 razy jako c8, czyli podwójne uszkodzenie występujące na tylnych śmigłach. Klasa c4 ponad 300 razy mylona była z c8, a c3 ok. 100 razy z c7. Pozostałe rodzaje uszkodzeń klasyfikowane były poprawnie przynajmniej w 99,5% przypadków, co dało ogólną dokładność modelu wynoszącą 98,47% dla testów terenowych przeprowadzanych w czasie rzeczywistym.

7.4 Ekstrakcja cech wyrażonych w dziedzinie częstotliwości

Jedną z lepiej przebadanych i najczęściej stosowanych operacji na sygnałach jest przekształcenie Fouriera. W przypadku okresowych ciągów o skończonej długości mamy do czynienia z DFT (dyskretna transformacja Fouriera, ang. *Discrete Fourier Transform*) [92]. Jej podstawowym zadaniem jest zamiana postaci sygnału z czasowej na częstotliwościową, co umożliwia analizę przekształcanego ciągu pod kątem występujących w nim składowych widmowych. Operację takiej analizy sygnału opisuje wzór (7.14):

$$X(k) = \begin{cases} \sum_{n=0}^{N-1} x(n)W_N^{kn} & \text{dla } 0 \leq k \leq N-1 \\ 0 & \text{dla pozostałych przypadków.} \end{cases} \quad (7.14)$$

W powyższym zapisie współczynnik W_N wyliczany jest zgodnie ze wzorem (7.15), który wygląda następująco:

$$W_N = e^{-j(2\pi/N)}. \quad (7.15)$$

Z kolei operacja przeciwna nazywana jest synteza i pozwala na odtworzenie próbek czasowych z danych częstotliwościowych. Odwrotne przekształcenie Fouriera przeprowadza się z użyciem wzoru (7.16):

$$x(n) = \begin{cases} \frac{1}{N} \sum_{k=0}^{N-1} X(k)W_N^{-kn} & \text{dla } 0 \leq n \leq N-1 \\ 0 & \text{dla pozostałych przypadków.} \end{cases} \quad (7.16)$$

Zasadniczą wadą korzystania z bezpośredniej metody obliczania DFT jest jej duża złożoność obliczeniowa. Dla N próbek sygnału wejściowego wymaga

ona N^2 mnożeń. Z tego powodu w praktyce korzysta się z innych technik transformacji sygnału do dziedziny częstotliwości. Jedną z nich jest szybkie przekształcenie Fouriera. Spotyka się algorytmy wykorzystujące przekształcenie z podziałem czasowym oraz częstotliwościowym. Najpopularniejszy algorytm FFT o podstawie 2 ma złożoność obliczeniową równą $O(N \log_2 N)$. Idea polega na dekompozycji N -punktowego DFT na dwa mniejsze, niezależne DFT. Pierwszy zawiera $N/2$ próbek parzystych, drugi – tyle samo próbek nieparzystych [25]. W przypadku stosowanego w badaniach podziału czasowego, sygnał dzielony jest na dwie równe części w pierwszym etapie przetwarzania. W kolejnym etapie każda z nich ponownie dzielona jest na dwa sygnały. Operacja jest powtarzana tyle razy, ile potrzeba do uzyskania N sygnałów zawierających po jednym punkcie [120]. Ponieważ N -punktowa transformata FFT o podstawie 2 wymaga $N = 2^k$ próbek wejściowych, w standardowo stosowanych algorytmach liczących nie jest możliwa analiza dowolnej długości ciągu wejściowego. Poza tym ograniczeniem, FFT jest w pełni równoważne bezpośrednioj metodzie DFT ze wszystkimi jej własnościami [80].

Z powodu korzystania z oprogramowania *MATLAB*, używany wzór do przekształcenia FFT (5.1) zawiera numerację próbek zaczynającą się od 1 zamiast od 0 [87]. Wartości amplitud sygnału w poszczególnych częstotliwościach są także odpowiednio skalowane. Jednak najważniejsze jest stosowanie spójnych algorytmów FFT do przygotowania danych treningowych oraz w docelowym mikrokontrolerze realizującym zadanie klasyfikacji uszkodzeń [63]. Zastosowane w *MATLAB* oraz *STM32CubeIDE* transformacje FFT dają te same wartości wyjściowe dla jednakowych próbek wejściowych. Pozwala to na korzystanie w systemie wbudowanym z modelu wytrenowanego na bazie zestawów danych przygotowanych na komputerze osobistym.

Pierwsze badania eksperymentalne, jakie zostały przeprowadzone w ramach prac nad doktoratem [99], wykorzystywały dane z drona *Falcon V5* [16]. W pracy [17] przedstawiono wyniki badań nad skutecznym wykrywaniem uszkodzeń śmigieł multirotora o konstrukcji X-8 (ze współosiowym układem jednostek napędowych). Jest to dron posiadający cztery ramiona, na końcach których umieszczono współosiowo po dwa silniki i dwa podwójne śmigła. Zapewnia to większą niezawodność i stabilność podczas lotu nawet w przypadku poważnej awarii któregoś z rotorów. Testom poddano 7 kombinacji wykorzystujących śmigła sprawne oraz z uszkodzeniem w postaci wyłamanej końcówki. Przeanalizowano przypadek ze sprawnymi śmigłami, cztery scenariusze o pojedynczych uszkodzeniach na kolejnych ramionach,

podwójne uszkodzenie na przeciwstawnych ramionach oraz podwójne uszkodzenie na sąsiadujących ramionach. W sumie zebrano ponad 300 sekund odczytów z czterech akcelerometrów. Wykorzystano po dwie poziome osie z czujników MMA8452. Akwizycja odbywała się z częstotliwością 400 Hz.

Autorzy wspomnianej pracy, wykorzystując metodę drzew decyzyjnych, uzyskali średnią dokładność na poziomie 86,6%. Największe trudności klasyfikator napotkał przy poprawnej identyfikacji ostatniej klasy – uszkodzeń występujących na dwóch sąsiednich ramionach drona. Dla tej klasy uszkodzeń czułość wyniosła zaledwie 0,48. Ta para uszkodzeń najczęściej mylona była z drugim rodzajem podwójnego uszkodzenia.

Na podstawie tych samych odczytów, wykorzystując dane zebrane przez zespół pracujący nad rozwojem platformy *Falcon V5*, postanowiono opracować nowy klasyfikator uszkodzeń w postaci sztucznej sieci neuronowej. Pełna procedura zawierała kilka kluczowych etapów. W pierwszym z nich dane z akcelerometrów wyznaczone były przez funkcję okna. Następnie nowe wartości sygnałów były przekształcane z zastosowaniem szybkiej transformacji Fouriera. Tak przygotowane dane poddawano procesowi trenowania modelu SSN, który był implementowany w mikrokontrolerze jako klasyfikator uszkodzeń śmigieł wielowirnikowego BSP.

Sygnały z czujników nie są idealnymi sinusoidami, a ciągi wejściowe zawierają także częstotliwości inne niż te, dla których przeprowadzana jest analiza widmowa (wielokrotności częstotliwości podstawowej f_s/N) [80]:

$$f_{analizowane}(k) = \frac{k \cdot f_s}{N}, \quad (7.17)$$

gdzie $k = 0, 1, 2, \dots, N - 1$.

Ciąg analizowanych próbek w rzeczywistości nie zawiera pełnych okresów składowych harmonicznych obecnych w sygnale. Energia poszczególnych komponentów widmowych sygnału wejściowego, które nie spełniają równania (7.17), rozpraszana jest na inne prążki. Te właściwości prowadzą do zniekształceń sygnału, określanych jako przeciek widma.

Aby ograniczyć wpływ przecieku widma na wiarygodność otrzymywanych składowych częstotliwościowych, powszechnie stosuje się technikę tłumienia wycinka sygnału na obu krańcach analizowanego przedziału. Taki

efekt uzyskuje się poprzez zastosowanie okna czasowego o kształcie innym niż prostokątny.

Podczas przygotowywania klasyfikatora SSN analizowano wpływ popularnych funkcji okna na zdolność do rozpoznawania uszkodzeń śmigieł drona. Wykorzystano następujące funkcje okna:

- **okno prostokątne**

$$w(n) = 1, \quad (7.18)$$

- **okno Hanna**

$$w(n) = 0,5 \cdot \left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right), \quad (7.19)$$

- **okno Hamminga**

$$w(n) = 0,54 - 0,46 \cdot \cos\left(\frac{2\pi n}{N-1}\right), \quad (7.20)$$

- **okno Blackmana**

$$w(n) = 0,42 - 0,5 \cdot \cos\left(\frac{2\pi n}{N-1}\right) + 0,08 \cdot \cos\left(\frac{4\pi n}{N-1}\right), \quad (7.21)$$

- **okno Blackmana-Harrisa**

$$w(n) = 0,35875 - 0,48829 \cdot \cos\left(\frac{2\pi n}{N-1}\right) + 0,14128 \cdot \cos\left(\frac{4\pi n}{N-1}\right) - 0,01168 \cdot \cos\left(\frac{6\pi n}{N-1}\right), \quad (7.22)$$

- **okno Nuttalla**

$$w(n) = 0,355768 - 0,487396 \cdot \cos\left(\frac{2\pi n}{N-1}\right) + 0,144232 \cdot \cos\left(\frac{4\pi n}{N-1}\right) - 0,012604 \cdot \cos\left(\frac{6\pi n}{N-1}\right), \quad (7.23)$$

- okno flat top

$$w(n) = 1 - 1,93 \cdot \cos\left(\frac{2\pi n}{N-1}\right) + 1,29 \cdot \cos\left(\frac{4\pi n}{N-1}\right) - 0,388 \cdot \cos\left(\frac{6\pi n}{N-1}\right) + 0,028 \cdot \cos\left(\frac{8\pi n}{N-1}\right), \quad (7.24)$$

gdzie n oraz N są takie, że $0 \leq n \leq N-1$. Badano także wpływ czasu trwania akwizycji pojedynczego okna pomiarowego (a co za tym idzie, liczby próbek z każdej osi czujnika poddawanych przetworzeniu FFT, a w konsekwencji – rozdzielczości częstotliwości), a także wybór liczby oraz położenia dolnego i górnego prążka widmowego (odpowiednio f_D i f_G), które ograniczały zakres analizowanego spektrum. Wyniki skuteczności klasyfikacji uzyskane na podstawie pomiarów z drona *Parrot Bebop 2*, uwzględniające wymienione parametry, zestawiono w tabeli 7.12.

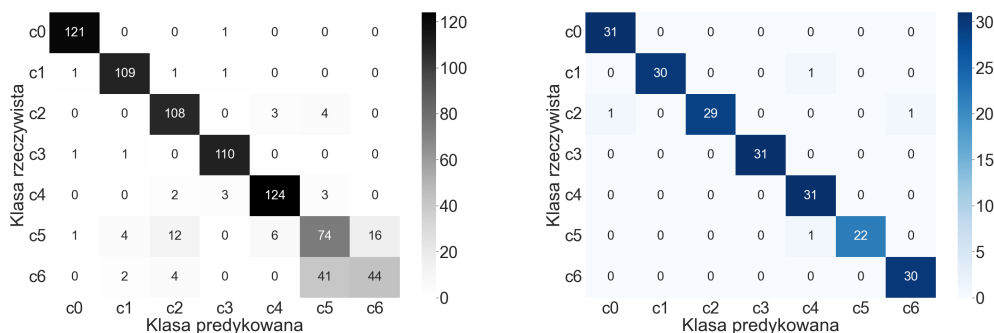
Tab. 7.12: Porównanie skuteczności klasyfikacji w zależności od różnych parametrów przetwarzania sygnałów (opracowanie na podstawie [99]).

Czas akwiz.	Szer. okna	Rozdz. częstotl.	Funkcja okna	Liczba prążków	f_D [Hz]	f_G [Hz]	Global. dokł.
160 ms	64	6,250 Hz	Hann	17	56	156	86,18%
320 ms	128	3,125 Hz	Hann	33	59	159	98,08%
640 ms	256	1,563 Hz	Hann	66	59	161	93,08%
1280 ms	512	0,781 Hz	Hann	130	61	162	92,16%
320 ms	128	3,125 Hz	prostokąt	33	59	159	98,08%
320 ms	128	3,125 Hz	Hann	33	59	159	98,08%
320 ms	128	3,125 Hz	Hamming	33	59	159	95,79%
320 ms	128	3,125 Hz	Blackman	33	59	159	96,55%
320 ms	128	3,125 Hz	Black.–Harris	33	59	159	87,36%
320 ms	128	3,125 Hz	Nuttall	33	59	159	93,10%
320 ms	128	3,125 Hz	flat top	33	59	159	79,23%
320 ms	128	3,125 Hz	Hann	64	0	197	93,27%
320 ms	128	3,125 Hz	Hann	25	59	134	98,08%
320 ms	128	3,125 Hz	Hann	26	75	153	92,33%
320 ms	128	3,125 Hz	Hann	27	75	156	92,95%
320 ms	128	3,125 Hz	Hann	27	88	169	92,28%
320 ms	128	3,125 Hz	Hann	26	81	159	93,93%
320 ms	128	3,125 Hz	Hann	25	122	197	77,24%

W pierwszej kolejności szukano optymalnego czasu akwizycji, sprawdzając zakresy od 160 do 1280 ms. Przy stałej częstotliwości akwizycji 400 Hz determinowało to szerokość okna pomiarowego oraz odległość (rozumianą jako odstęp w dziedzinie częstotliwości) pomiędzy sąsiednimi prążkami

widma. Badano częstotliwości z przedziału ok. 60–160 Hz z zastosowaniem okna Hanna. Następnie testowano wpływ konkretnej funkcji okna na wyniki klasyfikacji. Pomimo podobnej złożoności obliczeniowej szacowanej na $O(N)$ wszystkich badanych funkcji (za wyjątkiem okna prostokątnego, którego złożoność wynosi $O(1)$) oraz jednakowej szerokości okna i analizowanego wycinka widma częstotliwościowego, wyniki globalnej dokładności klasyfikatora znacząco się różniły. Na koniec sprawdzono, jak na zdolność wykrywania uszkodzeń śmigieł wpływa wybór szerokości i konkretnego umiejscowienia badanego zakresu częstotliwości.

Na podstawie tych eksperymentów wybrano optymalne parametry do stosowania w zadaniu wykrywania pojedynczych i podwójnych uszkodzeń końcówek śmigieł drona *Falcon V5*: czas akwizycji 320 ms, okno Hanna, zakres częstotliwości ok. 60–160 Hz. Takie parametry pozwoliły uzyskać dokładność równą 98,08% i czas przetwarzania wynoszący 8 ms z zastosowaniem klasyfikatora w postaci jednokierunkowej sztucznej sieci neuronowej z jedną warstwą ukrytą, zawierającą 160 neuronów. W porównaniu z 86,57% dla metody drzew decyzyjnych, poprawa dokładności jest znacząca. Wyniki klasyfikacji poszczególnych klas uszkodzeń z zastosowaniem pierwotnie użytej metody drzew decyzyjnych oraz nowego klasyfikatora SSN przedstawiono na rysunku 7.9.



(a) drzewa decyzyjne (86,57%)

(b) klasyfikator SSN (98,08%)

Rys. 7.9: Porównanie metody drzew decyzyjnych oraz proponowanego klasyfikatora w postaci modelu SSN

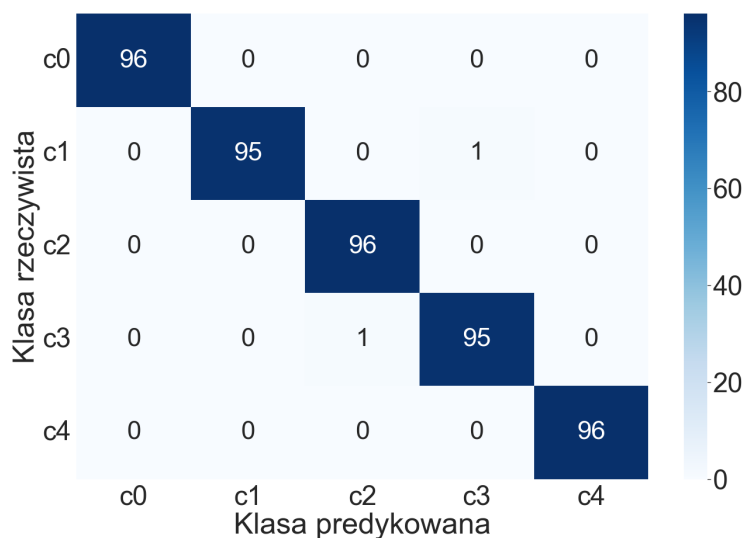
Kolejne badania dotyczyły już danych zebranych w ramach projektu PA-DRE. Dysponując pomiarami z akcelerometrów i żyroskopów umieszczonych na dronie *Parrot Bebop 2*, przygotowano klasyfikator zdolny do lokalizacji uszkodzenia pojedynczego śmigła w postaci wyszczerbionej krawędzi (rysunek 5.2). Zatem rozważanych jest 5 klas – jedna odpowiadająca sprawnym śmigłom oraz cztery klasy, w których uszkodzone śmigło znajduje się kolejno

na jednym z czterech ramion BSP. W tabeli 7.13 zestawiono parametry klasyfikatora najlepiej radzącego sobie z zadaniem lokalizacji tego typu usterki. Model korzystający z odczytów ze wszystkich dostępnych osi pomiarowych czujników w 99,58% przypadków prawidłowo rozpoznał miejsce wystąpienia anomalii. Na 480 dokonanych predykcji klasyfikator dwukrotnie wskazał błędne ramię drona (rysunek 7.10).

Tab. 7.13: Parametry modelu dziedziny częstotliwości dla 5 klas z pojedynczymi uszkodzeniami w postaci wyszczerbionej krawędzi śmigła.

Parametr	Wartość
model BSP	Parrot Bebop 2
wykorzystane osie akcelerometru	XYZ
wykorzystane osie żyroskopu	XYZ
liczba wykorzystanych osi	24
szerokość okna pomiarowego	128 próbek
funkcja okna	Hann
liczba cech każdej osi	33
typ sieci	jednokierunkowa
rozmiar wsadu	512
liczba warstw ukrytych	1
liczba neuronów w warstwie ukrytej	96
numer epoki	20000
<i>train_loss</i>	0,08
<i>train_acc</i>	0,99
<i>valid_loss</i>	0,08
<i>valid_acc</i>	1,00
<i>test_loss</i>	0,0474
dokładność (<i>test_acc</i>)	0,9958
Q_{AC}	1,16
rozmiar pliku <i>h5</i> z modelem	619 KiB

Podobne badanie przeprowadzono dla drugiego typu uszkodzenia – wygiętej końcówki śmigła. W tym przypadku dokładność wyniosła 99,17% (tabela 7.14). Klasyfikator czterokrotnie niewłaściwie określił lokalizację uszkodzenia. Pozostałe 476 predykcji przeprowadził poprawnie (rysunek 7.11). Oba modele wskazujące miejsce wystąpienia uszkodzenia śmigła, w 100% przypadków poprawnie wskazały klasę bez uszkodzeń (*c0*). Żadne uszkodzenie nie zostało także zignorowane. Nawet jeśli klasyfikator się mylił co do lokalizacji uszkodzenia, to sam fakt występowania nieprawidłowości był poprawnie wskazywany bez fałszywych alarmów.



Rys. 7.10: Macierz błędów modelu dziedziny częstotliwości dla 5 klas z pojedynczymi uszkodzeniami w postaci wyszczerbionej krawędzi śmigła

Tab. 7.14: Parametry modelu dziedziny częstotliwości dla 5 klas z pojedynczymi uszkodzeniami w postaci wygiętej końcówki śmigła.

Parametr	Wartość
model BSP	Parrot Bebop 2
wykorzystane osie akcelerometru	XYZ
wykorzystane osie żyroskopu	XYZ
liczba wykorzystanych osi	24
szerokość okna pomiarowego	128 próbek
funkcja okna	Hann
liczba cech każdej osi	65
typ sieci	jednokierunkowa
rozmiar wsadu	512
liczba warstw ukrytych	1
liczba neuronów w warstwie ukrytej	64
numer epoki	20000
<i>train_loss</i>	0,08
<i>train_acc</i>	1,00
<i>valid_loss</i>	0,08
<i>valid_acc</i>	0,99
<i>test_loss</i>	0,0918
dokładność (<i>test_acc</i>)	0,9917
Q_{AC}	1,14
rozmiar pliku <i>h5</i> z modelem	802 KiB



Rys. 7.11: Macierz błędów modelu dziedziny częstotliwości dla 5 klas z pojedynczymi uszkodzeniami w postaci wygiętej końcówki śmigła

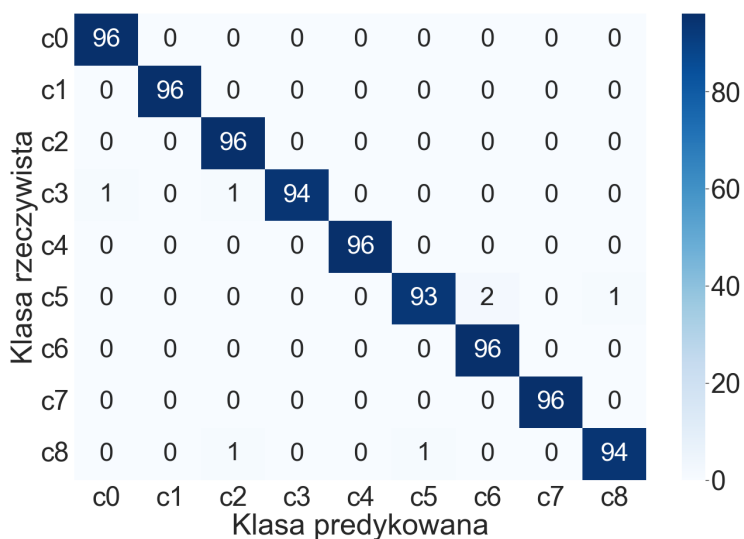
W nieco trudniejszym zadaniu jednoczesnej lokalizacji oraz rozróżnienia rodzaju uszkodzenia, wyodrębniono 9 scenariuszy. Pierwszy to zdrowy system bez uszkodzeń, kolejne cztery dotyczą pojedynczych uszkodzeń w postaci wyszczerbionej krawędzi śmigła, a ostatnie cztery to wygięte końcówki śmigła występujące kolejno na każdym z ramion *Bebop 2*. Ciekawą obserwacją jest fakt, iż model ten wykazał nieco wyższą dokładność od poprzednio omawianego klasyfikatora rozpatrującego 5 klas. Uzyskany wynik to 99,19%. Prawdopodobną przyczyną tego zjawiska jest stosunkowo łatwiejsza lokalizacja pierwszego typu uszkodzenia, co przekłada się na poprawę efektywności również w modelach analizujących oba rodzaje anomalii. Parametry tego klasyfikatora wyszczególniono w tabeli 7.15, a macierz błędów przedstawiono na rysunku 7.12.

Najbardziej złożony problem dotyczy 20 klas, w których występują uszkodzenia w liczbie od 0 do 4. Dotyczą one dwóch rodzajów uszkodzeń omawianych w kilku poprzednich testach. Dla 20 klas, korzystając z odczytów ze wszystkich dostępnych osi akcelerometrów i żyroskopów, uzyskana dokładność wyniosła 98,80%. Szczegółowe parametry modelu znajdują się w tabeli 7.16, a wyniki poszczególnych klasyfikacji na rysunku 7.13.

Dotychczas omówione klasyfikatory posługujące się danymi częstotliwościowymi, z uwagi na korzystanie z odczytów ze wszystkich dostępnych osi używanego czujnika (lub czujników), cechowały się relatywnie długim

Tab. 7.15: Parametry modelu dziedziny częstotliwości dla 9 klas z pojedynczymi uszkodzeniami dwóch rodzajów.

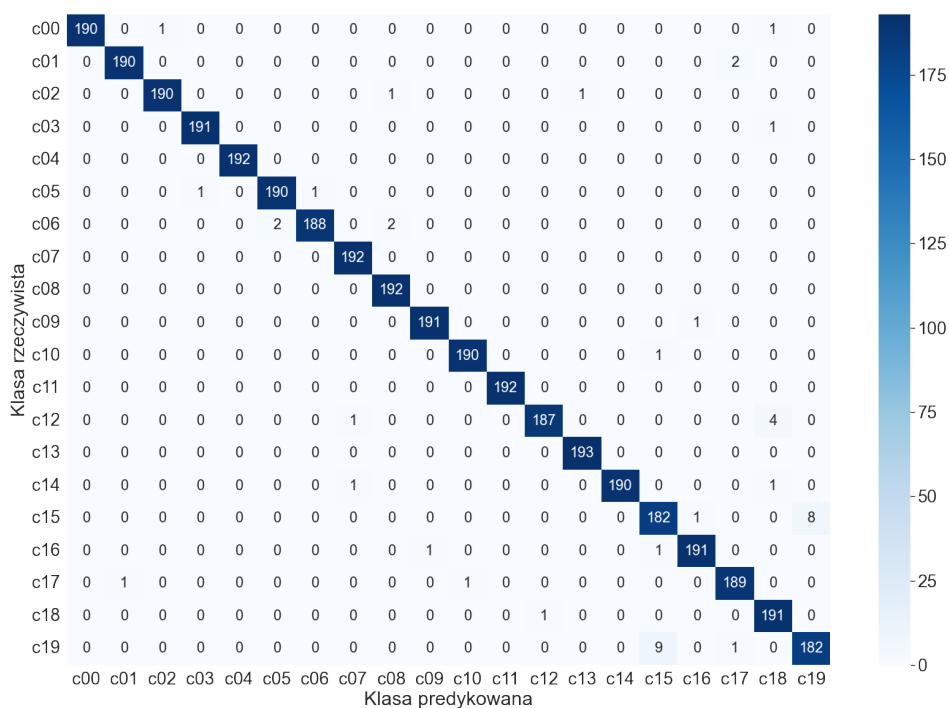
Parametr	Wartość
model BSP	Parrot Bebop 2
wykorzystane osie akcelerometru	XYZ
wykorzystane osie żyroskopu	XYZ
liczba wykorzystanych osi	24
szerokość okna pomiarowego	128 próbek
funkcja okna	Hann
liczba cech każdej osi	33
typ sieci	jednokierunkowa
rozmiar wsadu	512
liczba warstw ukrytych	1
liczba neuronów w warstwie ukrytej	96
numer epoki	20000
<i>train_loss</i>	0,09
<i>train_acc</i>	0,99
<i>valid_loss</i>	0,09
<i>valid_acc</i>	1,00
<i>test_loss</i>	0,0853
dokładność (<i>test_acc</i>)	0,9919
Q_{AC}	1,27
rozmiar pliku <i>h5</i> z modelem	622 KiB



Rys. 7.12: Macierz błędów modelu dziedziny częstotliwości dla 9 klas z pojedynczymi uszkodzeniami dwóch rodzajów

Tab. 7.16: Parametry modelu dziedziny częstotliwości dla 20 uszkodzeń.

Parametr	Wartość
model BSP	Parrot Bebop 2
wykorzystane osie akcelerometru	XYZ
wykorzystane osie żyroskopu	XYZ
liczba wykorzystanych osi	24
szerokość okna pomiarowego	64 próbki
funkcja okna	Hann
liczba cech każdej osi	33
typ sieci	jednokierunkowa
rozmiar wsadu	1000
liczba warstw ukrytych	1
liczba neuronów w warstwie ukrytej	70
numer epoki	42437
<i>train_loss</i>	0,2067
<i>train_acc</i>	0,9509
<i>valid_loss</i>	0,1248
<i>valid_acc</i>	0,9852
<i>test_loss</i>	0,1198
dokładność (<i>test_acc</i>)	0,9880
Q_{AC}	1,39
rozmiar pliku <i>h5</i> z modelem	465 KiB



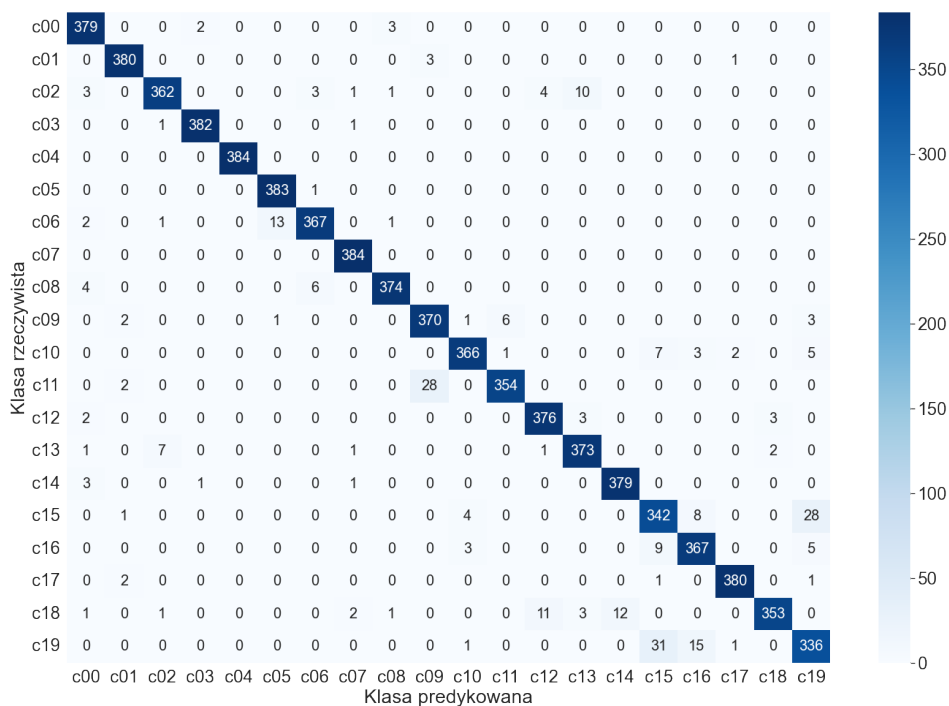
Rys. 7.13: Macierz błędów modelu dziedziny częstotliwości dla 20 klas uszkodzeń

czasem przetwarzania. Aby skrócić czas obliczeń do poziomu poniżej 2 ms, należało zmniejszyć ilość przetwarzanych danych. Po sprawdzeniu różnych możliwości, optymalnym wyborem okazało się wykorzystanie jedynie osi z akcelerometrów oraz osi x żyroskopów (tabela 7.17). Taki zabieg pozwolił na skrócenie czasu przetwarzania do 1,95 ms, umożliwiając przeprowadzenie klasyfikacji w czasie rzeczywistym. Dokładność modelu spadła do 96,25%. Macierz błędów dla tego klasyfikatora prezentuje rysunek 7.14.

Tab. 7.17: Parametry modelu dziedziny częstotliwości dla 20 uszkodzeń przeprowadzającego klasyfikację w czasie rzeczywistym.

Parametr	Wartość
model BSP	Parrot Bebop 2
wykorzystane osie akcelerometru	Z
wykorzystane osie żyroskopu	X
liczba wykorzystanych osi	8
szerokość okna pomiarowego	32 próbki
funkcja okna	Hann
liczba cech każdej osi	17
typ sieci	jednokierunkowa
rozmiar wsadu	500
liczba warstw ukrytych	1
liczba neuronów w warstwie ukrytej	120
numer epoki	9494
czas przetwarzania	1,95 ms
<i>train_loss</i>	0,2975
<i>train_acc</i>	0,9206
<i>valid_loss</i>	0,2077
<i>valid_acc</i>	0,9573
<i>test_loss</i>	0,1963
dokładność (<i>test_acc</i>)	0,9625
Q_{AC}	1,29
Q_{ACT}	1,47
rozmiar pliku <i>h5</i> z modelem	167 KiB

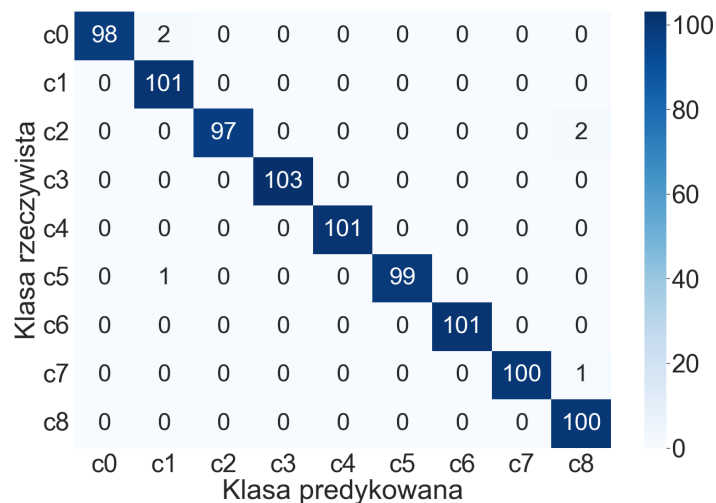
Kolejne testy dotyczą drona *3DR Solo* i obejmują 9 klas z pojedynczymi uszkodzeniami dwóch rozmiarów: braku 1 cm oraz 2 cm końcówki śmigła. Pierwszy model korzysta z odczytów ze wszystkich osi akcelerometrów oraz żyroskopów i wykorzystuje konwolucyjną sieć neuronową (tabela 7.18). Dokładność klasyfikacji z jego użyciem wyniosła 99,34%, co przy 906 próbach odpowiada 900 poprawnym predykcjom (rysunek 7.15).



Rys. 7.14: Macierz błędów modelu dziedziny częstotliwości dla 20 klas uszkodzeń przeprowadzającego klasyfikację w czasie rzeczywistym

Tab. 7.18: Parametry modelu dziedziny częstotliwości dla 9 klas uszkodzeń o dwóch rozmiarach.

Parametr	Wartość
model BSP	3DR Solo
wykorzystane osie akcelerometru	XYZ
wykorzystane osie żyroskopu	XYZ
liczba wykorzystanych osi	12
szerokość okna pomiarowego	128 próbek
funkcja okna	Hann
liczba cech każdej osi	65
typ sieci	konwolucyjna
rozmiar wsadu	2000
liczba filtrów	10
rozmiar jądra	8
numer epoki	988
<i>train_loss</i>	0,0140
<i>train_acc</i>	0,9998
<i>valid_loss</i>	0,0238
<i>valid_acc</i>	0,9956
<i>test_loss</i>	0,0339
dokładność (<i>test_acc</i>)	0,9934
Q_{AC}	1,27
rozmiar pliku <i>h5</i> z modelem	1112 KiB



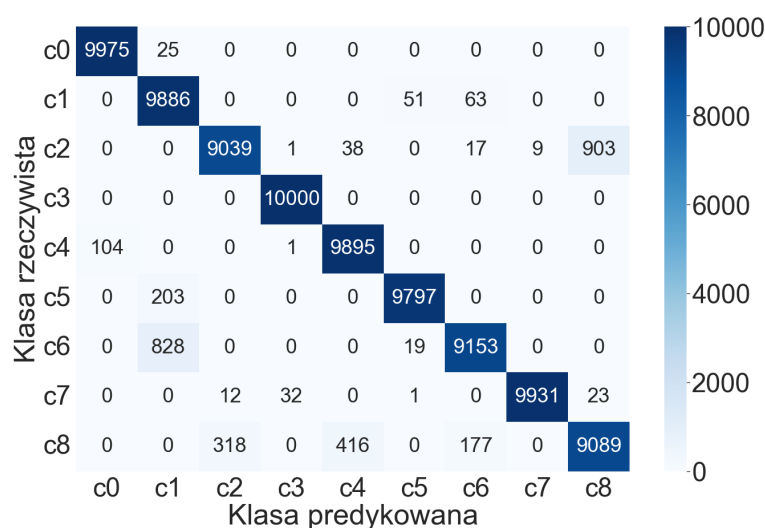
Rys. 7.15: Macierz błędów modelu dziedziny częstotliwości dla 9 klas uszkodzeń o dwóch rozmiarach

Powyższy model potrzebuje prawie 300 ms na wykonanie wszystkich obliczeń i uzyskanie wyniku klasyfikacji. Aby znacząco zminimalizować czas przetwarzania, postanowiono wybrać tylko jedną oś pomiarową z każdego czujnika umieszczonego na ramieniu drona. Po wielu próbach wybór padł na oś y akcelerometru, która dostarczała modelowi najwięcej informacji dających możliwość przewidzenia miejsca i rozmiaru występowania usterki. W testach offline klasyfikator taki uzyskał dokładność na poziomie 97,44% przy zaledwie 0,77 ms opóźnienia. Z tego powodu został wybrany do testów wykonywanych podczas lotu w rzeczywistych warunkach terenowych. Eksperyment przebiegł analogicznie do opisanego w podrozdziale 7.3. Podczas 9 lotów z różnymi konfiguracjami uszkodzeń przeprowadzono w sumie 90000 klasyfikacji. Poprawnie sklasyfikowano 86765 niezależnych predykcji, co dało globalną dokładność na poziomie 96,41%. Wynik ten osiągnięto pomimo stosunkowo prostego modelu korzystającego z widma zawierającego zaledwie 9 niezależnych pasm częstotliwości sygnałów pochodzących z jednej osi akcelerometru oraz 80 neuronów pojedynczej warstwy ukrytej jednokierunkowej sieci neuronowej (tabela 7.19). Sześć z badanych klas uzyskało dokładność powyżej 97%, a klasa c3 została poprawnie rozpoznana w 100% przypadków (rysunek 7.16).

Dysponując sygnałami dźwiękowymi z *3DR Solo*, przygotowano także klasyfikator korzystający z danych akustycznych. Widmo sygnału podzielono na 513 różnych częstotliwości i – korzystając z sieci konwolucyjnej – wytrenowano model, który w testach osiągnął dokładność 97,43%. W tabeli 7.20

Tab. 7.19: Parametry modelu dziedziny częstotliwości dla 9 klas uszkodzeń o dwóch rozmiarach użytego podczas rzeczywistych testów terenowych.

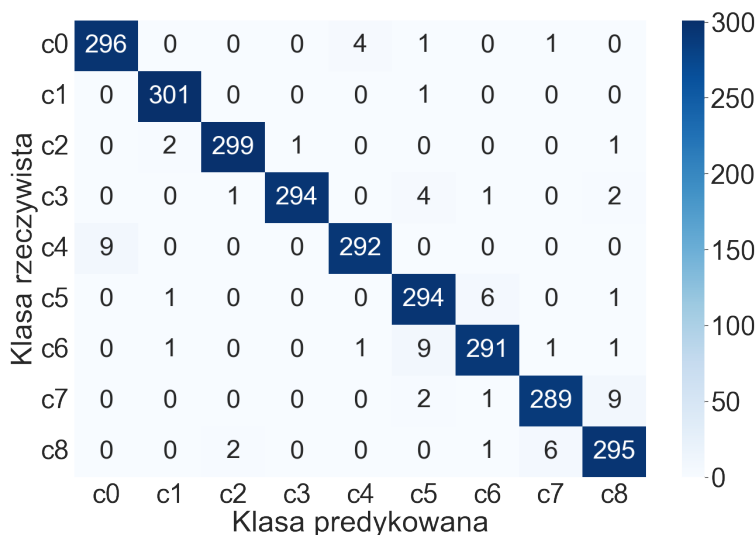
Parametr	Wartość
model BSP	3DR Solo
wykorzystane osie akcelerometru	Y
wykorzystane osie żyroskopu	–
liczba wykorzystanych osi	4
szerokość okna pomiarowego	16 próbek
funkcja okna	prostokąt
liczba cech każdej osi	9
typ sieci	jednokierunkowa
rozmiar wsadu	512
liczba warstw ukrytych	1
liczba neuronów w warstwie ukrytej	80
numer epoki	4871
czas przetwarzania	0,77 ms
<i>train_loss</i>	0,1333
<i>train_acc</i>	0,9608
<i>valid_loss</i>	0,0955
<i>valid_acc</i>	0,9722
<i>test_loss</i>	0,0900
dokładność (<i>test_acc</i>)	0,9744
dokładność w teście terenowym	0,9641
Q_{AC}	1,20
Q_{AC} w teście terenowym	1,16
Q_{ACT}	2,33
Q_{ACT} w teście terenowym	2,25
rozmiar pliku <i>h5</i> z modelem	49 KiB



Rys. 7.16: Macierz błędów modelu dziedziny częstotliwości dla 9 klas uszkodzeń o dwóch rozmiarach użytego podczas rzeczywistych testów terenowych

Tab. 7.20: Parametry modelu dziedziny częstotliwości wykorzystującego sygnały dźwiękowe dla 9 klas uszkodzeń o dwóch rozmiarach.

Parametr	Wartość
model BSP	3DR Solo
rodzaj czujnika	mikrofon
liczba wykorzystanych kanałów	4
szerokość okna pomiarowego	1024 próbki
funkcja okna	Hann
liczba cech każdego kanału	513
typ sieci	konwolucyjna
rozmiar wsadu	512
liczba filtrów	1
rozmiar jądra	20
numer epoki	847
<i>train_loss</i>	0,1179
<i>train_acc</i>	0,9900
<i>valid_loss</i>	0,1603
<i>valid_acc</i>	0,9743
<i>test_loss</i>	0,1605
dokładność (<i>test_acc</i>)	0,9743
Q_{AC}	1,20
rozmiar pliku <i>h5</i> z modelem	740 KiB



Rys. 7.17: Macierz błędów modelu dziedziny częstotliwości wykorzystującego sygnały dźwiękowe dla 9 klas uszkodzeń o dwóch rozmiarach

zestawiono najważniejsze parametry klasyfikatora, a na rysunku 7.17 zamieszczono wyniki klasyfikacji w postaci macierzy błędów. Akwizycję danych przeprowadzano w sposób umożliwiający efektywne próbkowanie sygnału z częstotliwością 48 kHz. Obejmuje to z zapasem całe pasmo akustyczne. Choć uzyskane wyniki są obiecujące, to ze względu na dużą objętość danych nie było możliwe przygotowanie modelu pracującego w czasie rzeczywistym na wybranym mikrokontrolerze. Przetwarzanie takiej ilości danych wymaga czasu przekraczającego wymogi nałożone na systemy czasu rzeczywistego.

7.5 Ekstrakcja cech czasowo-częstotliwościowych

Oprócz dobrze przebadanych i wielokrotnie przetestowanych pomiarów w dziedzinie czasu oraz w dziedzinie częstotliwości, postanowiono sprawdzić skuteczność detekcji i klasyfikacji uszkodzeń śmigieł z wykorzystaniem danych o mieszanej naturze. Jednym z przykładów reprezentacji czasowo-częstotliwościowej jest DWT, znana też jako przekształcenie lub transformacja zafalowaniowa [98, 26, 72]. W odróżnieniu od DFT, która w teorii pozwala na odtworzenie sygnału pierwotnego, DWT umożliwia także uchwycenie zmian widma sygnału następującego w czasie. DWT bazuje na oknach o zmiennej rozdzielczości, co stanowi główny czynnik odróżniający ją od STFT (krótkookresowa transformacja Fouriera, ang. *Short-Time Fourier Transform*), która rozkłada sygnał na składowe o tej samej rozdzielczości [134]. Na przykład w problemach dotyczących mowy lub muzyki, gdzie składowych częstotliwościowych dźwięku zwykle nie powinno się rozpatrywać w ujęciu globalnym, DWT wydaje się bardziej naturalnym wyborem do analizy sygnału [52]. Jednakże również w analizie anomalii pojazdów autonomicznych spotkać można falki używane w zadaniach dotyczących detekcji uszkodzeń [148, 153, 48]. Dane z drona *3DR Solo* przekształcono za pomocą oprogramowania *MATLAB*, korzystając z funkcji *dwt* oraz falek Daubechies 4. rzędu (*db4*) i dyskretnej falki Meyera (*dmey*). Analizie poddawana była zarówno część uproszczona (tzw. aproksymacja), jak i część zawierająca współczynniki szczegółowości (tzw. detal). We wszystkich testach korzystano z wsadu o rozmiarze 1024 oraz jednokierunkowej SSN o jednej warstwie ukrytej zawierającej 130 neuronów. Klasyfikatory trenowano przez 1000 epok. Badano różne szerokości okien pomiarowych, wynoszących 16, 32 oraz 64 odczyty z wszystkich osi używanych czujników inercyjnych. Najlepsze

wyniki otrzymano dla danych ze wszystkich akcelerometrów o szerokości okna pomiarowego wynoszącej 32 próbki oraz falek $db4$. Uzyskana globalna dokładność wyniosła 0,9904 przy parametrze *test_loss* równym 0,1117. Dla dwukrotnie mniejszej szerokości okna dokładność była tylko nieznacznie gorsza. Klasyfikatory korzystające z przekształcenia DWT danych żyroskopowych wykazywały niższą precyzję.

Kolejnym przykładem metody badania złożonych sygnałów o nieznanach częstotliwościach jest EMD [107]. Podobnie jak DWT, EMD dekomponuje sygnał na składniki odpowiadające różnym częstotliwościom. Także umożliwia uchwycenie zarówno ogólnych trendów, jak i bardziej szczegółowych oscylacji. Natomiast tę metodę cechuje bardziej adaptacyjny sposób analizy, dostosowujący się do charakteru zmian w czasie widma badanego sygnału [149]. Również w tym przypadku posłużono się oprogramowaniem *MATLAB*, korzystając z funkcji *emd*. Rozkłada ona sygnał na IMF, czyli tzw. mody oraz część *residual*, której nie można już podzielić na mniejsze składniki. Liczba modów zależy od charakteru analizowanego sygnału. Aby ujednoczyć proces przetwarzania sygnałów, dla każdego z niezależnych odczytów z czujników drona *3DR Solo* badano 5 pierwszych części z rozkładu EMD, niezależnie od faktu, czy było to 5 IMF, czy 4 IMF oraz 1 część *residual*. Dla tego rodzaju przetwarzania danych najwyższą dokładność, na poziomie 0,9861, uzyskano dla odczytów ze wszystkich akcelerometrów, z użyciem okien o szerokości 16 próbek dla każdej osi. Wskaźnik *test_loss* wyniósł 0,1208. Nieco gorszy wynik osiągnięto przy 32-elementowej szerokości okna. Pomiary z żyroskopów uzyskiwały dokładność niższą o ok. 0,1 w porównaniu z analogicznymi danymi z akcelerometrów. Proces trenowania SSN cechowały te same parametry, co w przypadku DWT.

Ostatnim przekształceniem czasowo-częstotliwościowym, które analizowano w ramach badań opisywanych w rozprawie, była HHT [57]. Jest to rozwinięcie EMD o transformację Hilberta [35], którą stosuje się w odniesieniu do każdego z wyodrębnionych modów IMF. Operacja ta poszerza analizę EMD o wyznaczenie chwilowej częstotliwości i amplitudy sygnału, a zarazem charakteru ich zmian w czasie. Te same dane wejściowe i parametry, użyte podczas testów z użyciem DWT i EMD, zostały zastosowane w przypadku HHT. Uzyskane wyniki okazały się jednak dużo gorsze niż we wcześniejszych testach. Najwyższy wynik dokładności przypadł ponownie pomiarom akcelerometrycznym o 16-elementowym oknie pomiarowym. Wyniósł 0,9092 przy *test_loss* równym 0,3791. Im szersze okno pomiarowe,

tym gorszy klasyfikator. Dane z żyroskopów pozwoliły na uzyskanie *test_acc* na poziomie 0,6125 dla *test_loss* wynoszącego aż 1,2199. Jedną z prawdopodobnych przyczyn uzyskiwania wyników gorszych niż dla analogicznych danych EMD jest większa zawartość informacyjna, która wymaga bardziej złożonych struktur SSN. Jednakże na potrzeby porównania metod omówionych w tym podrozdziale zdecydowano się na ujednoczenie parametrów i hiperparametrów procesu trenowania klasyfikatora.

W tabeli 7.21 zestawiono wyniki trzech omawianych metod wykorzystujących ekstrakcję cech czasowo-częstotliwościowych. Z uwagi na złożoność obliczeniową, techniki te zastosowano jedynie do obliczeń offline na komputerze stacjonarnym. Ponieważ EMD i HHT obecnie nie wykazują zdolności do klasyfikowania uszkodzeń śmigieł BSP w czasie rzeczywistym, zaniechano przeprowadzania testów z użyciem mikrokontrolera dla tej grupy metod. Nie zmienia to faktu, że bardzo obiecujące są wyniki testów wykorzystujących przekształcenie DWT. Jego zastosowanie wykazuje potencjalnie wyższą skuteczność w klasyfikacji uszkodzeń od alternatywnych metod dziedziny czasu oraz dziedziny częstotliwości. Stosując proste filtry, prawdopodobnie możliwe byłoby użycie DWT w zadaniach FDI także dla modelu pracującego w mikrokontrolerze, z zachowaniem reżimu czasu rzeczywistego.

Tab. 7.21: Wyniki klasyfikacji 9 klas uszkodzeń drona *3DR Solo* z wykorzystaniem cech czasowo-częstotliwościowych.

Metoda	Szer. okna	Typ czujnika	<i>train loss</i>	<i>train acc</i>	<i>valid loss</i>	<i>valid acc</i>	<i>test loss</i>	<i>test acc</i>
DWT	16	akcel.	0,1220	0,9797	0,0921	0,9899	0,0900	0,9901
DWT	32	akcel.	0,1352	0,9815	0,1166	0,9893	0,1117	0,9904
DWT	64	akcel.	0,1738	0,9790	0,1560	0,9862	0,1893	0,9735
DWT	16	żyros.	0,3397	0,9435	0,2547	0,9813	0,2506	0,9840
DWT	32	żyros.	0,4516	0,9304	0,3542	0,9760	0,3594	0,9757
DWT	64	żyros.	0,5970	0,9226	0,5738	0,9465	0,5551	0,9570
EMD	16	akcel.	0,1408	0,9771	0,1145	0,9877	0,1208	0,9861
EMD	32	akcel.	0,1559	0,9800	0,1524	0,9804	0,1437	0,9848
EMD	64	akcel.	0,1950	0,9813	0,2288	0,9642	0,2305	0,9680
EMD	16	żyros.	0,3774	0,9360	0,2856	0,9792	0,2887	0,9774
EMD	32	żyros.	0,4570	0,9294	0,3655	0,9757	0,3639	0,9749
EMD	64	żyros.	0,6123	0,9171	0,5643	0,9619	0,5887	0,9410
HHT	16	akcel.	0,3555	0,9163	0,3812	0,9080	0,3791	0,9092
HHT	32	akcel.	0,4334	0,8903	0,4428	0,8867	0,4669	0,8797
HHT	64	akcel.	1,0972	0,6804	1,2143	0,6417	1,2440	0,6376
HHT	16	żyros.	1,6484	0,4259	1,5886	0,5045	1,5866	0,4959
HHT	32	żyros.	1,2866	0,5681	1,2283	0,6053	1,2199	0,6125
HHT	64	żyros.	1,4729	0,4815	1,4520	0,4884	1,4434	0,4837

7.6 Detekcja uszkodzeń spoza zbioru treningowego modelu

Jednym z założeń przeprowadzanych testów było określenie skuteczności wykrywania wieloklasowych scenariuszy uszkodzeń. Obejmują one uszkodzenia różnych typów, różnych rozmiarów, a także uszkodzenia wielokrotne. Wszystkie opisane dotychczas eksperymenty przeprowadzane były w powszechnie stosowanej konwencji, polegającej na wykrywaniu takich anomalii, do których rozpoznawania model klasyfikatora był przygotowywany na etapie trenowania. W tego typu podejściu istnieje jednak pewne niebezpieczeństwo, polegające na zbyt dokładnym dopasowaniu modelu do danych treningowych, czyli w tym przypadku do uszkodzeń konkretnego rodzaju i konkretnego rozmiaru. W związku z tym postanowiono przeprowadzić eksperymenty, w których modele wyuczone do rozpoznawania jednego rodzaju uszkodzenia będą dokonywały wnioskowania na temat występującego uszkodzenia innego rodzaju lub o innym rozmiarze.

Przeprowadzono serię testów na dronie *Parrot Bebop 2*, w których do trenowania i walidacji modelu wykorzystano pomiary zebrane z lotów z pojedynczym uszkodzeniem w postaci wyszczerbionej krawędzi łożaty śmigła (*typ 1*). Danymi do testowania były z kolei pomiary z lotów, podczas których końcówka łożaty śmigła była wygięta (*typ 2*). Zatem klasy opisane w podrozdziale 5.1 jako *1000*, *0100*, *0010* i *0001* zostały połączone w jedną klasę uszkodzeń, natomiast drugą rozpatrywaną klasę stanowił system sprawny *0000*. Dane zostały tak przygotowane, by obie nowe klasy były równo reprezentowane przez dostępne dane pomiarowe, a żaden fragment pliku z zestawu treningowego klasy *0000* nie znalazł się w zestawie testowym. Z kolei klasę uszkodzeń na potrzeby testów tworzył zestaw danych złożony z klas wcześniej oznaczonych jako *2000*, *0200*, *0020* i *0002*. W podobny sposób przygotowano modele do detekcji wyszczerbionej krawędzi, wytrenowane na danych dotyczących wygiętej końcówki.

W odniesieniu do drona *3DR Solo*, z dwóch klas obejmujących jedno-centymetrowe uszkodzenie końcówki śmigła, przedstawionych w podrozdziale 5.2 jako *0010* oraz *0001*, stworzono nową klasę o nazwie *1 cm*. Z kolei z klas *2000* oraz *0200*, dotyczących dwukrotnie większego uszkodzenia, powstała nowa klasa *2 cm*. Zarówno dane treningowe, walidacyjne oraz testowe,

jak i sam sposób przeprowadzenia eksperymentów, przygotowano w sposób analogiczny do przypadku dotyczącego drona *Parrot Bebop 2*.

Na bazie tak przygotowanych zestawów treningowych i testowych przeprowadzono analizę skuteczności detekcji nieznanymi klasyfikatorowi uszkodzeń, wykorzystując dane reprezentowane w dziedzinie czasu oraz cechy FFT. Wyniki uzyskane przez poszczególne modele zestawiono w tabeli 7.22.

Tab. 7.22: Wyniki detekcji uszkodzeń, na których model nie był trenowany.

BSP	Rodzaj cech	Szerokość okna	Liczba osi akc/żyr/mik	Uszkodzenie trenowane	Uszkodzenie testowane	Ogólna dokładność
Bebop	czas	8	3/-/-	typ 1	typ 2	0,8704
Bebop	czas	8	3/-/-	typ 2	typ 1	0,8001
Bebop	czas	8	-/3/-	typ 1	typ 2	0,9277
Bebop	czas	8	-/3/-	typ 2	typ 1	0,8487
Bebop	FFT	128	3/3/-	typ 1	typ 2	0,8182
Bebop	FFT	128	3/3/-	typ 2	typ 1	0,7604
Solo	czas	16	3/-/-	1 cm	2 cm	0,8838
Solo	czas	16	3/-/-	2 cm	1 cm	0,5081
Solo	czas	16	-/3/-	1 cm	2 cm	0,9975
Solo	czas	16	-/3/-	2 cm	1 cm	0,5782
Solo	FFT	128	3/3/-	1 cm	2 cm	1,0000
Solo	FFT	128	3/3/-	2 cm	1 cm	0,6300
Solo	FFT	1024	-/-/1	1 cm	2 cm	0,9570
Solo	FFT	1024	-/-/1	2 cm	1 cm	0,6887

Zauważyć można, że modele wytrenowane na uszkodzeniu obiektywnie mniejszym (1 cm) lub mającym prawdopodobnie mniejszy wpływ na aerodynamikę pracy śmigła (typ 1), uzyskują wyraźnie większą dokładność detekcji poważniejszych uszkodzeń (2 cm, typ 2) od modeli wytrenowanych na uszkodzeniach większych, a rozpoznających mniej znaczące anomalie. W tym drugim przypadku najwyższą skuteczność detekcji uszkodzeń wykazał model, który przetwarzał sygnały z żyroskopu reprezentowane w dziedzinie czasu. Podczas detekcji wyszczerbionej krawędzi śmigła na dronie *Parrot Bebop 2* model uzyskał dokładność wynoszącą 84,87%.

Rozpatrując detekcję wygiętej końcówki oraz uszkodzenia polegającego na utracie 2 cm łopaty śmigła, rezultaty kształtują się dużo lepiej. Klasyfikatory bazujące na sygnałach w dziedzinie czasu, których złożoność obliczeniowa pozwala na pracę w czasie rzeczywistym, lepiej radziły sobie

z pomiarami żyroskopowymi niż akcelerometrycznymi. Najlepszy klasyfikator wykazał precyzję na poziomie 99,75%. W przypadku danych częstotliwościowych korzystano z obu czujników inercyjnych. Dokładność jednego z modeli sięgnęła tutaj 100%. Korzystając z sygnałów akustycznych uzyskano 95,70% poprawnych wnioskowań.

Przedstawione w tabeli 7.22 wyniki dotyczą procesu detekcji, czyli przeprowadzania klasyfikacji binarnej. Dla danych z drona *Parrot Bebop 2* przetestowano także wieloklasową lokalizację uszkodzeń. Uzyskane rezultaty nie pozwalają jednak uznać przeprowadzonej klasyfikacji za skuteczną, w związku z czym nie zostały one uwzględnione w zestawieniu. Najwyższa dokładność lokalizacji w pięcioklasowym problemie wyniosła 51,46%. O ile klasy *c0* (brak uszkodzeń) oraz *c1* (uszkodzenie śmigła A) były rozpoznawane z wysoką czułością (odpowiednio 99,28% i 99,80%), o tyle w przypadku pozostałych model wykazał znacząco niższą skuteczność (poniżej 55%). Prawdopodobną przyczyną takiego stanu rzeczy jest zbyt mała liczba różnych rodzajów uszkodzeń rozpatrywanych na etapie trenowania modelu. Przypuszcza się, że użycie kilku różnych typów anomalii do wyuczenia SSN pozwoliłoby na dużo większe uogólnienie rozpoznawanych wzorców uszkodzeń. Ponieważ w dotychczas zebranych danych analizowano jedynie dwa typy defektów, trudno mówić o uniwersalności zastosowanego klasyfikatora. Z kolei w przypadku *3DR Solo*, sposób rozstawienia uszkodzonych śmigieł na poszczególnych ramionach drona uniemożliwił przeprowadzenie podobnych eksperymentów. Uszkodzenia konkretnych rozmiarów obejmowały śmigła wirujące zawsze w jednym kierunku. Zatem nie były wykonywane loty, podczas których na danym ramieniu drona raz było zamontowane śmigło uszkodzone w mniejszym, a raz w większym stopniu.

7.7 Podsumowanie testów skuteczności klasyfikatorów

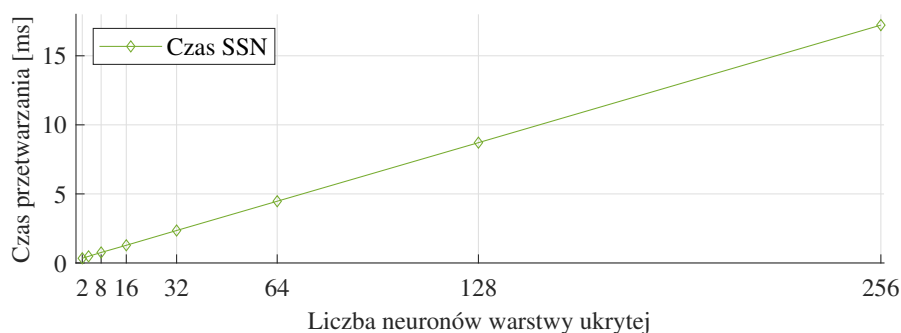
Na różnych etapach prac nad detekcją uszkodzeń śmigieł multirotorów wykonano setki, a być może nawet tysiące testów różnych klasyfikatorów. Poszukując najlepszych parametrów, zapewniając możliwość uzyskiwania powtarzalnych wyników, a także optymalizując sam proces wykonywania testów, sposób przeprowadzania eksperymentów ewoluował w czasie. Kolejne etapy badań wskazywały najbardziej obiecujące kierunki, jednocześnie

sygnalizowały, z których pomysłów należy rezygnować. W związku z tym nie każdy z przytoczonych wyników testów opatrzony jest wszystkimi parametrami albo podane parametry cechują się niższą precyzją. Tam, gdzie możliwe było powtórzenie testów z zachowaniem nowo wypracowanych standardów, robiono to. Jeżeli natomiast ponowienie eksperymentów było niemożliwe albo wiązało się z nieproporcjonalnym do potencjalnych korzyści nakładem pracy, rezygnowano z powtarzania testów. Celowo zaniechano wykonywania pomiarów czasu przetwarzania przez klasyfikatory, które z założenia nie były przystosowane do pracy w czasie rzeczywistym. W początkowym stadium badań głównym celem było wypracowanie metody zdolnej do jak najdokładniejszego wykrywania usterek z jak najmniejszą liczbą fałszywych alarmów. Dopiero opracowanie modelu zdolnego do działania w mikrokontrolerze pokazało możliwości szybkiego przetwarzania sygnałów w systemach wbudowanych. Po pierwszych obiecujących wynikach takiego sposobu działania, ciężar prac przeniesiono w kierunku klasyfikacji przeprowadzanej w sposób ciągły podczas lotu BSP. Natomiast modele SSN, których parametry wykluczają możliwości takiego sposobu działania, nie posiadają wyliczonego wskaźnika Q_{ACT} . Zwykle jednak w takim przypadku opracowywany był mniej złożony model realizujący identyczne zadanie klasyfikacji, ale przystosowany do pracy w czasie rzeczywistym.

Choć rozprawa dotyczy precyzyjnej diagnostyki uszkodzeń polegającej na określaniu miejsca wystąpienia anomalii oraz wskazaniu rozmiaru uszkodzenia, przeprowadzono także testy obejmujące jedynie pierwszy etap diagnostyki, a mianowicie samą detekcję uszkodzeń. Jednakże idea tych eksperymentów zawiera w sobie niespotykane w innych badaniach podejście. Polega ono na wykrywaniu uszkodzeń, do których rozpoznawania klasyfikator nie był trenowany. Zarówno inny typ badanej nieprawidłowości, jak i inny rozmiar uszkodzenia, dają możliwość wykorzystania jednej z największych zalet technik uczenia maszynowego – zdolności do uogólniania. Chociaż niektóre przedstawione w tabeli 7.22 wyniki nie są bardzo wysokie (zwłaszcza w odniesieniu do prostej klasyfikacji binarnej), to dają nadzieję na dalszy rozwój metod diagnostyki uszkodzeń opartych na SSN.

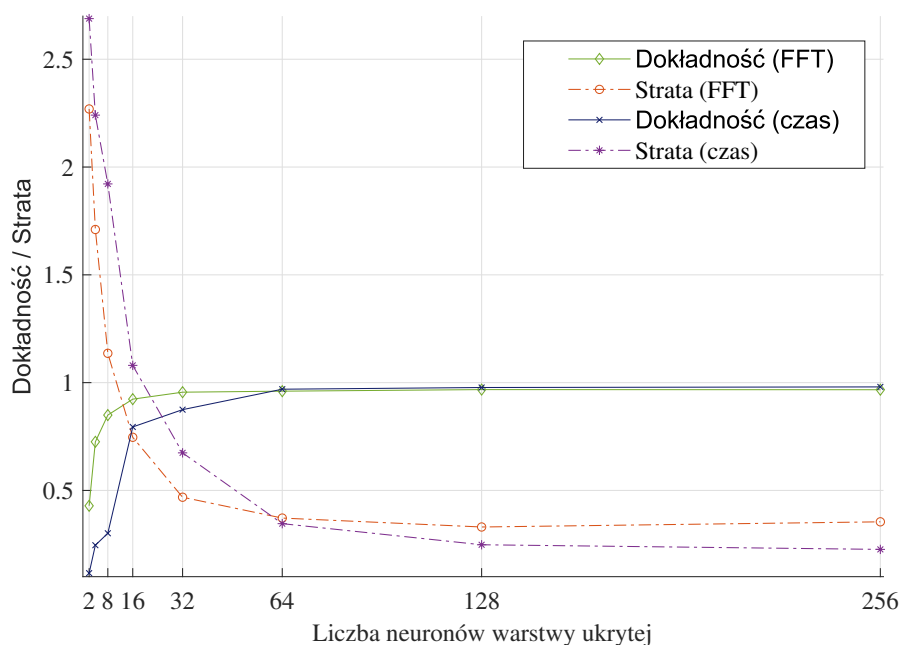
Analiza uzyskanych rezultatów

W rozdziale 7 przedstawiono wyniki przeprowadzonych testów skuteczności działania klasyfikatorów uszkodzeń śmigieł wielowirnikowych BSP. Badane modele charakteryzują się różnymi parametrami i złożonością. Zwykle eksperymentalnie dobierane były wartości różnych parametrów, by znaleźć możliwie najlepszą zależność między skutecznością klasyfikacji a stopniem skomplikowania modelu i czasem potrzebnym na uzyskanie wyników. Jedną ze zmiennych, którą należało dobrać, była architektura zastosowanej sieci neuronowej. Bardzo szybko okazało się, że sieci o liczbie warstw ukrytych większej od 1 radziły sobie zdecydowanie gorzej. Wyniki dokładności modeli wielowarstwowych odbiegały znacznie od precyzji uzyskiwanych przez sieci zbudowane z 1 warstwy ukrytej. Natomiast liczbę neuronów znajdujących się w tej warstwie należało dopasować empirycznie. Logika podpowiada, iż im większa będzie ta wartość, tym czas przetwarzania również będzie większy. Okazuje się, że opóźnienia wprowadzane przez sam człon SSN działającego w mikrokontrolerze programu są w przybliżeniu wprost proporcjonalne do liczby neuronów. Widać to wyraźnie na rysunku 8.1, przedstawiającym wykres tych zależności, mający w przybliżeniu charakter liniowy. Dotyczy on przykładowego klasyfikatora przetwarzającego pomiary z dziedziny częstotliwości, trenowanego przez 1000 epok. Szerokość okna pomiarowego to 128 próbek z każdej z 12 osi akcelerometrów i 12 osi żyroskopów.



Rys. 8.1: Wpływ liczby neuronów warstwy ukrytej na czas przetwarzania przez SSN (opracowanie na podstawie [102])

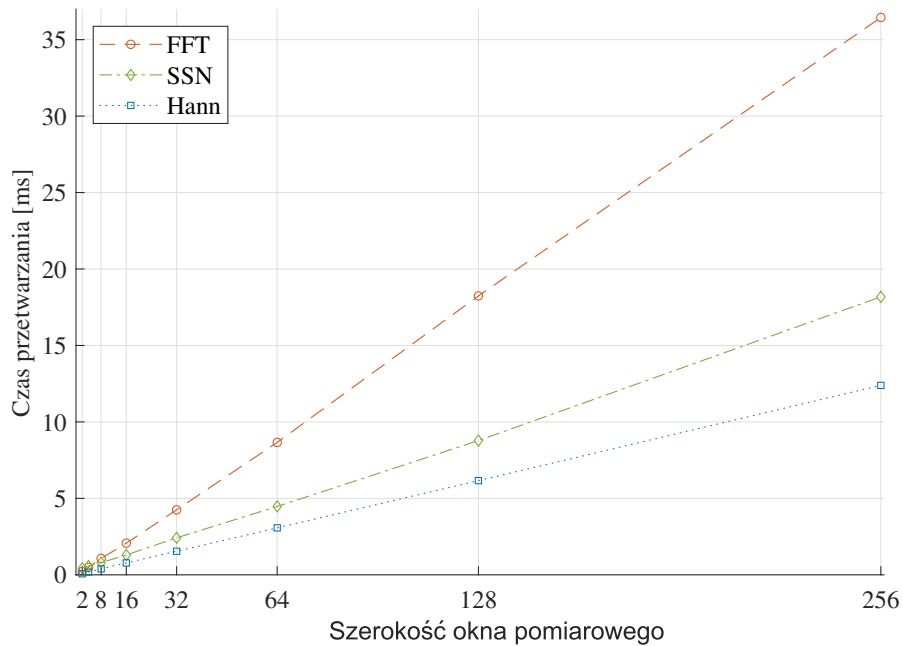
Wpływ liczby neuronów warstwy ukrytej na metryki dokładności oraz straty nie jest już tak oczywisty. Rysunek 8.2 prezentuje tę zależność dla dwóch rodzajów sygnałów wejściowych: w dziedzinie czasu oraz w dziedzinie częstotliwości (*FFT*). Model dziedziny czasu obejmuje odczyty z 12 osi żyroskopów o 8-elementowej długości okna pomiarowego. Model *FFT* przyjmuje parametry identyczne jak klasyfikator scharakteryzowany na rysunku 8.1. Modele dziedziny czasu i częstotliwości opisane w dalszej części rozdziału były przygotowane w analogiczny sposób. O ile początkowo powiększanie złożoności warstwy ukrytej pozytywnie wpływa na analizowane wskaźniki, o tyle przy 256 neuronach wyniki modelu *FFT* są już nieco gorsze niż dla 128 neuronów. Dalsze zwiększanie tego parametru nie przynosi poprawy jakości działania, a jedynie zwiększa złożoność obliczeniową.



Rys. 8.2: Wpływ liczby neuronów warstwy ukrytej na wartość dokładności i straty (opracowanie na podstawie [102])

Podobnie jak wzrost liczby neuronów, także szerokość okna pomiarowego oddziałuje na czas obliczeń potrzebnych do uzyskania wyniku klasyfikacji. Wpływ liczby próbek pomiarowych przetwarzanych przez model na czas wykonywania obliczeń na poszczególnych etapach przetwarzania przedstawia rysunek 8.3. Znaczenie tego parametru objawia się przede wszystkim podczas przeprowadzania przekształcenia Fouriera. Nieco mniej wpływa natomiast na czas przetwarzania danych przez sieć neuronową oraz na opóźnienia wprowadzane przez operację wymnażania pomiarów przez funkcję

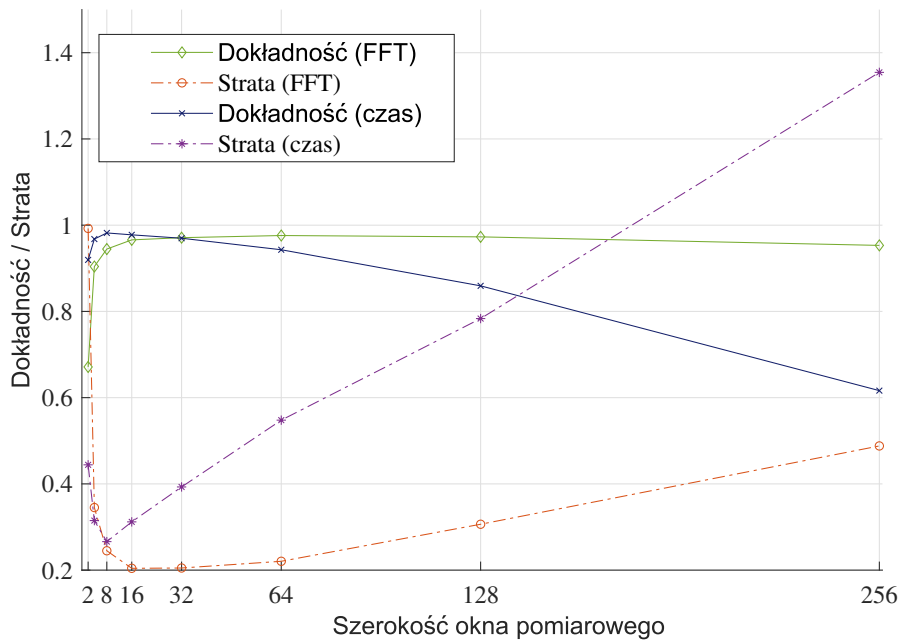
okna. Jednak w przypadku konieczności zastosowania wszystkich tych operacji, dobór odpowiedniej szerokości okna pomiarowego jest kluczowy dla szybkości działania całego systemu.



Rys. 8.3: Wpływ szerokości okna pomiarowego na czas obliczeń w poszczególnych etapach przetwarzania danych: zastosowanie funkcji okna Hann, FFT, klasyfikacja SSN (opracowanie na podstawie [102])

Z kolei stopień oddziaływania liczby próbek zawartych w pojedynczym oknie pomiarowym na precyzję klasyfikacji pokazano na rysunku 8.4. W odniesieniu do modelu częstotliwościowego dokładność klasyfikacji rośnie wraz z poszerzaniem okna. Maksymalne wartości dokładności przypadają na okna o szerokości 64 oraz 128 próbek. Jednak wykres straty zaczyna nieznacznie rosnąć już dla okien szerszych od 32 próbek. Negatywne oddziaływanie przesadnego zwiększania tego parametru na jakość klasyfikatora widać jeszcze wyraźniej dla modelu dziedziny czasu. W tym przypadku optymalna wartość mieściła się w okolicy zaledwie 8 próbek. Jest to dość nieintuicyjne, ale dla zastosowanych parametrów i hiperparametrów SSN, szersze okna pomiarowe mocno pogarszały uzyskiwane przez klasyfikator wyniki.

Dostrzec można, że poza bezsprzecznym wpływem poziomu złożoności modelu i liczby przetwarzanych próbek na czas działania klasyfikatora, inne relacje dotyczące parametrów modelu i uzyskiwanych wyników nie są już tak oczywiste. W celu otrzymania precyzyjniejszej klasyfikacji nie wystarczy zastosowanie bardziej skomplikowanej architektury sieci neuronowej ani po-

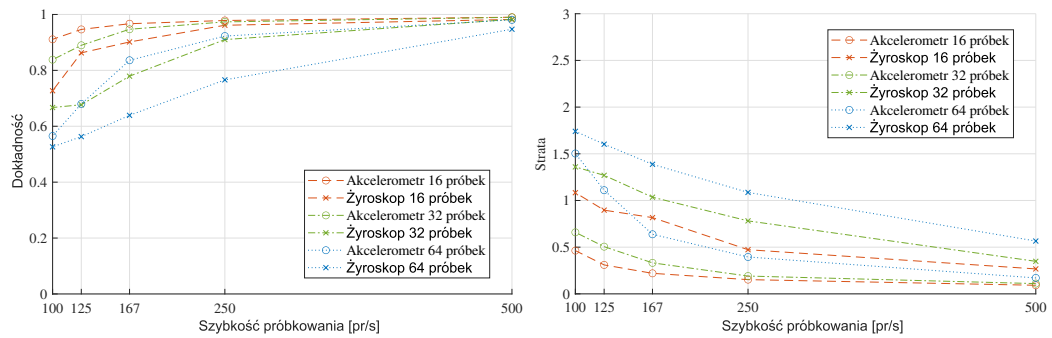


Rys. 8.4: Wpływ szerokości okna pomiarowego na wartość dokładności i straty (opracowanie na podstawie [102])

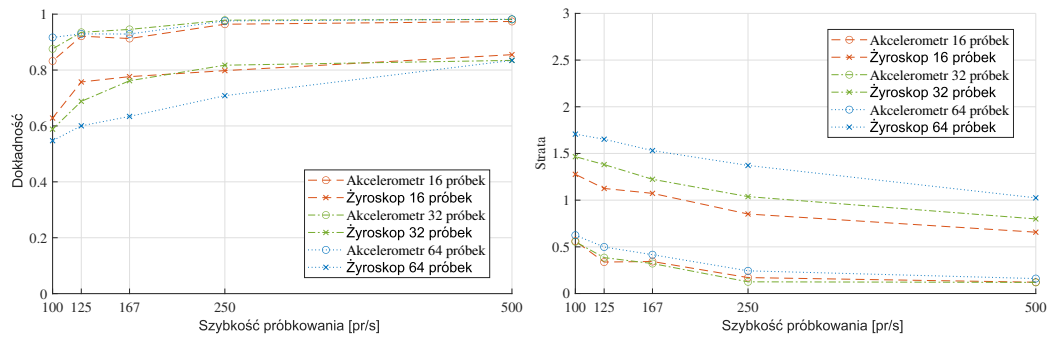
szerzenie okna pomiarowego. Gdy dodatkowo dochodzi potrzeba sprostania wymogom czasowym, problem doboru odpowiedniej struktury i parametrów modelu przestaje być trywialny.

Podczas przeprowadzania eksperymentów posługiwano się danymi z dronów *Parrot Bebop 2* oraz *3DR Solo*. Dane te były pobierane z czujników z szybkością 500 akwizycji na sekundę. Sprawdzone również, jak częstotliwość próbkowania wpływa na skuteczność klasyfikacji. Oprócz nominalnej częstotliwości $f_s = 500$ Hz, badaniem objęto sygnały próbkowane 2, 3, 4 i 5 razy wolniej, a mianowicie o $f_s = 250$ Hz, $f_s = 167$ Hz, $f_s = 125$ Hz oraz $f_s = 100$ Hz. Aby jak najwyraźniej pokazać wpływ tego parametru, jednocześnie minimalizując oddziaływanie innych wielkości, podczas eksperymentów korzystano zawsze z tych samych pomiarów. W testach użyto po prostu konkretnych odczytów wybranych w taki sposób, by symulowały próbkowanie z niższą częstotliwością. Na przykład dla uzyskania $f_s = 125$ Hz, z danego pliku zawierającego dane sensoryczne wyselekcjonowano co czwarty odczyt. Przeprowadzono proces trenowania i testowania modeli osobno dla danych z akcelerometru oraz dla danych z żyroskopu. Eksperymenty powtórzono dla okien pomiarowych o długości 16, 32 i 64 próbek z każdej z osi odpowiedniego czujnika. W ten sposób uzyskano po 6 wykresów prezentujących maksymalną dokładność oraz odpowiadającą im stratę dla różnych

typów danych wejściowych: sygnałów dziedziny czasu (rysunek 8.5), cech dziedziny częstotliwości (rysunek 8.6), cech DWT (rysunek 8.7), cech EMD (rysunek 8.8) oraz cech HHT (rysunek 8.9).



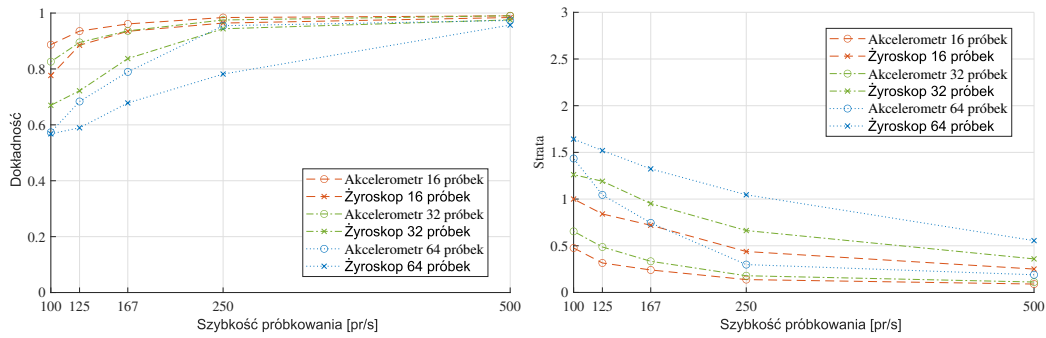
Rys. 8.5: Wpływ częstotliwości próbkowania na wyniki klasyfikacji dla różnych szerokości okna pomiarowego dla sygnałów dziedziny czasu



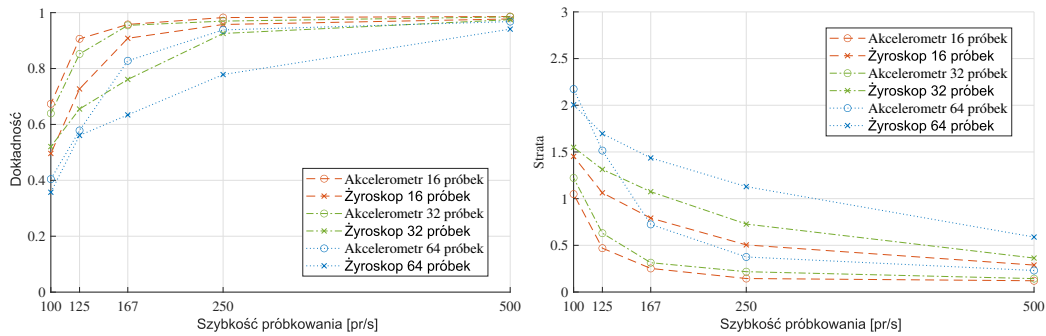
Rys. 8.6: Wpływ częstotliwości próbkowania na wyniki klasyfikacji dla różnych szerokości okna pomiarowego dla sygnałów dziedziny częstotliwości (FFT)

Przedstawione wykresy jednoznacznie wskazują, że zmniejszenie częstotliwości próbkowania negatywnie wpływa na wyniki przeprowadzanej klasyfikacji. Ogólny trend objawia się zarówno w przypadku badanej dokładności, jak i funkcji straty. Jedynie dane czasowo-częstotliwościowe przetworzone z zastosowaniem HHT nie są jednoznaczne pod tym względem. Natomiast osiągnięte przez te modele wyniki są przede wszystkim zbyt niskie, by używać klasyfikatora przygotowanego w ten sposób. Prawdopodobnie proces trenowania tego typu modeli wymagałby zastosowania bardziej złożonej architektury SSN, co wykraczało poza zakres prac opisywanych w rozprawie i stanowi aspekt istotny poznawczo w odrębnych, przyszłych badaniach.

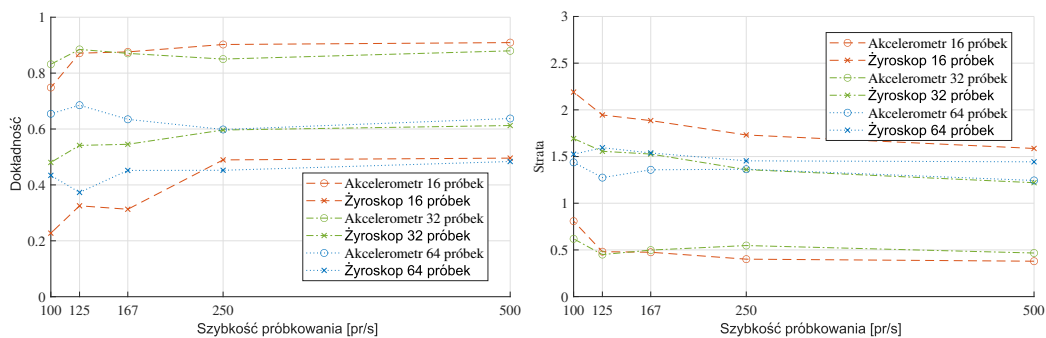
W tabeli 8.1 dokonano podsumowania klasyfikatorów uszkodzeń, których parametry i wyniki przeprowadzonych testów przedstawiono w postaci tabelarycznej oraz w formie macierzy błędów w całym rozdziale 7. Dla modeli korzystających z cech czasowo-częstotliwościowych nie zamieszczono macierzy błędów, gdyż uznano, że zmniejszyłoby to czytelność pracy. Modele te były



Rys. 8.7: Wpływ częstotliwości próbkowania na wyniki klasyfikacji dla różnych szerokości okna pomiarowego dla sygnałów czasowo-częstotliwościowych DWT



Rys. 8.8: Wpływ częstotliwości próbkowania na wyniki klasyfikacji dla różnych szerokości okna pomiarowego dla sygnałów czasowo-częstotliwościowych EMD



Rys. 8.9: Wpływ częstotliwości próbkowania na wyniki klasyfikacji dla różnych szerokości okna pomiarowego dla sygnałów czasowo-częstotliwościowych HHT

testowane jedynie offline, gdyż nie stanowiły przedmiotu badań z użyciem mikrokontrolera. W podsumowaniu zawarto jednak te klasyfikatory, które uzyskały najwyższą dokładność podczas testów, by można je było porównać z modelami opartymi na cechach reprezentowanych w dziedzinie czasu oraz w dziedzinie częstotliwości.

Tab. 8.1: Podsumowanie wyników testów przebadanych klasyfikatorów uszkodzeń.

Macierz błędów	BSP	Rodzaj cech	Liczba osi akc/żyr/mik	Szer. okna	Liczba klas	Dokł. [%]	Q_{AC}	Q_{ACT}
Rys. 7.2	Bebop	czas	-/3/-	8	5	99,52	1,16	1,77
Rys. 7.3	Bebop	czas	-/3/-	2	5	98,88	1,13	3,00
Rys. 7.4	Bebop	czas	-/3/-	8	5	99,15	1,14	1,75
Rys. 7.5	Bebop	czas	-/3/-	8	9	98,99	1,26	1,91
Rys. 7.6	Bebop	czas	-/3/-	8	20	98,17	1,36	2,05
Rys. 7.7	Solo	czas	3/-/-	32	9	99,50	1,28	-
Rys. 7.8	Solo	czas	3/-/-	16	9	98,47	1,26	1,67
Rys. 7.9b	Falcon	FFT	2/-/-	128	7	98,08	1,18	-
Rys. 7.10	Bebop	FFT	3/3/-	128	5	99,58	1,16	-
Rys. 7.11	Bebop	FFT	3/3/-	128	5	99,17	1,14	-
Rys. 7.12	Bebop	FFT	3/3/-	128	9	99,19	1,27	-
Rys. 7.13	Bebop	FFT	3/3/-	64	20	98,80	1,39	-
Rys. 7.14	Bebop	FFT	1/1/-	32	20	96,25	1,29	1,47
Rys. 7.15	Solo	FFT	3/3/-	128	9	99,34	1,27	-
Rys. 7.16	Solo	FFT	1/-/-	16	9	96,41	1,20	2,25
Rys. 7.17	Solo	FFT	-/-/1	1024	9	97,43	1,20	-
-	Solo	DWT	3/-/-	32	9	99,04	1,26	-
-	Solo	EMD	3/-/-	16	9	98,61	1,25	-
-	Solo	HHT	3/-/-	16	9	90,92	0,98	-

Konfrontując przedstawione wyniki z publikacjami przeanalizowanymi w rozdziale 2, dostrzec można wysoki potencjał zaprezentowanych w rozprawie metod. Szczególnie uwzględniając liczbę badanych klas uszkodzeń, wyłania się obraz bardzo obiecujących możliwości, jakie daje proponowany klasyfikator w zestawieniu z mikrokontrolerem, w którym realizowane jest zadanie diagnostyki. Widać także dużą elastyczność w zakresie stosowanych sygnałów wejściowych w odniesieniu do typu sensora, używanych osi pomiarowych oraz liczby przetwarzanych próbek. Opisywane metody detekcji pozwalają na szerokie modyfikacje w tym zakresie, wciąż umożliwiając uzyskanie modelu o wysokiej skuteczności działania. Jest to duża zaleta SSN, uprawniająca do stwierdzenia, iż nie sposób znaleźć granice stosowalności rozwiązań uczenia maszynowego. Nawet dysponując dużymi ograniczeniami w zakresie używanego sprzętu i dostępnych sygnałów pomia-

rowych, można budować modele niejednokrotnie przewyższające uzyskiwaną precyzją klasyfikacji i krótkim czasem przetwarzania dużo bardziej złożone i wyspecjalizowane systemy.

W odniesieniu do publikacji [58, 109, 17, 19, 135, 10, 11, 123], w których można jasno wskazać liczbę klas uszkodzeń oraz choćby w przybliżeniu uzyskaną dokładność (lub ewentualnie inną użytą miarę prezentującą skuteczność klasyfikacji), wyliczono wartość Q_{AC} dla parametru $\alpha = 3$. W tabeli 8.2 zestawiono wyniki uzyskane w testach realizowanych w ramach badań opisanych w rozdziale 2 z najlepszymi wynikami eksperymentów korzystających z metod zaproponowanych w rozprawie. Dopisek „RT” oznacza pracę systemu FDI w czasie rzeczywistym.

Tab. 8.2: Porównanie jakości klasyfikacji różnych badań z uwzględnieniem wskaźnika Q_{AC} .

Badanie	Dokładność	Liczba klas	Q_{AC}	Uwagi
[58]	0,9763	3	0,96	
[109]	0,8500	15	0,80	
[17]	0,8657	7	0,81	
[19]	0,9940	9	1,28	
[135]	0,7632	16	0,62	
[10]	0,9821	3	0,98	
[11]	0,9757	2	0,82	
[11]	0,6970	3	0,35	
[123]	0,9977	2	0,88	
[123]	0,9686	3	0,94	
<i>Parot Bebop 2</i>	0,9880	20	1,39	dz. częstotliwości
<i>Parot Bebop 2</i>	0,9817	20	1,36	dz. czasu, RT
<i>3DR Solo</i>	0,9950	9	1,28	dz. czasu
<i>3DR Solo</i>	0,9847	9	1,24	dz. czasu, RT

Choć niektórzy badacze wskazują na wysoki stopień dokładności predykcji uszkodzeń osiąganą za pomocą stosowanych przez nich metod, zazwyczaj dotyczą one stosunkowo prostych problemów, takich jak klasyfikacja binarna lub analiza trzech scenariuszy usterek. Po uwzględnieniu proponowanego wskaźnika jakości klasyfikacji Q_{AC} okazuje się jednak, że poza modelami prezentowanymi w rozprawie, wartość wskaźnika powyżej 1,0 osiągnęła tylko metoda MFCC zaprezentowana w artykule [19]. Wyznaczenie wskaźnika Q_{ACT} było niemożliwe, ponieważ autorzy najczęściej nie podają faktycznego czasu potrzebnego na uzyskanie wyniku pojedynczej klasyfikacji.

W kontekście badań nad detekcją uszkodzeń BSP opartą na zróżnicowanej informacji sensorycznej, nie dostrzega się wprost przydatności sygnałów barometrycznych. Postanowiono ich jednak nie usuwać z repozytorium PADRE. Podlegały one akwizycji przygotowanym do tego celu systemem, ponieważ zastosowany moduł z czujnikami bezwładnościowymi został wyposażony także w barometry. Choć można spotkać prace, w których pomiary barometryczne są używane np. w filtrach Kalmana jako jeden z kilku rodzajów mierzonych wartości [150], trudno znaleźć badania, w których pomiary z czujników ciśnienia wprost wykorzystywane są w zadaniach detekcji uszkodzeń śmigieł BSP. Według obecnego stanu wiedzy, ze względu na wolnozmiennosc odczytów oraz duże zakłócenia środowiskowe wywołujące fluktuacje ciśnienia nawet w obrębie niewielkich odległości w przestrzeni, pomiary barometryczne są niejako naturalnie pomijane. Być może, dzięki zastosowaniu innej metody, w przyszłości możliwe będzie wykorzystanie także tych pomiarów. Jako potencjalne przystosowanie tego rodzaju czujników można rozważyć zwiększenie ich czułości i szybkości akwizycji. Korzystne może okazać się także uwzględnienie dodatkowego barometru pełniącego funkcję referencyjną. Ponieważ ciśnienie atmosferyczne maleje wraz ze wzrostem wysokości, należy wziąć pod uwagę jego zmiany, do czego użyty może zostać umieszczony na kadłubie drona pomocniczy czujnik, który wyznaczałby punkt odniesienia tej wielkości fizycznej. Właściwe czujniki zlokalizowane w pobliżu śmigieł byłyby w takim układzie wykorzystane do śledzenia wahań ciśnienia spowodowanych pracą rotorów.

Niniejsza rozprawa porusza problematykę wykrywania uszkodzeń bezzałogowych statków powietrznych. W ramach prac badawczych opracowano metody detekcji anomalii i uszkodzeń śmigieł wielowirnikowych dronów. Choć temat rozprawy odnosi się do samej detekcji, w rzeczywistości opracowane techniki obejmują także lokalizację oraz identyfikację usterek, obejmując w pełni proces diagnostyki uszkodzeń. Przedstawione metody można podzielić ze względu na kilka kategorii:

- a) rodzaj używanych sygnałów pomiarowych
 - odczyty z akcelerometrów,
 - odczyty z żyroskopów,
 - odczyty z mikrofonów,
 - odczyty z różnych typów czujników,
- b) typy uszkodzeń śmigieł
 - różne rodzaje uszkodzeń,
 - różne rozmiary uszkodzeń,
 - różna lokalizacja uszkodzeń,
 - różna liczba jednocześnie występujących uszkodzeń,
- c) tryb pracy systemu detekcji
 - praca w trybie czasu rzeczywistego,
 - praca w trybie niespełniającym trybu czasu rzeczywistego,
- d) miejsce przeprowadzania eksperymentów
 - laboratorium (arena do lotów),
 - otwarta przestrzeń miejska.

W trakcie prac nad systemami diagnostyki przeprowadzono szereg testów skuteczności metod klasyfikacji uszkodzeń w każdej z ww. kategorii. Jako punkt wyjścia przyjęto możliwość zastosowania danego rozwiązania na pokładzie BSP bez konieczności korzystania z zewnętrznych jednostek obliczeniowych, bazując jedynie na natywnym mikrokontrolerze danego drona. Przedstawiono wyniki prezentujące dokładność działania poszczególnych modeli z uwzględnieniem czasu przetwarzania podczas pracy w trybie czasu rzeczywistego. Wartości dodane, których pomysł wdrożenia zrodził się już

w trakcie badań, mogą stanowić ogólnodostępne repozytorium danych pomiarowych PADRE oraz dwa nowe wskaźniki oceny jakości klasyfikatorów: Q_{AC} i Q_{ACT} .

W toku przeprowadzanych badań i eksperymentów udało się zrealizować najważniejsze cele w kontekście wykazania tezy głównej i tez szczegółowych. W szczególności skutecznie przeprowadzono automatyczną diagnostykę uszkodzeń z wykorzystaniem systemów znajdujących się na pokładzie BSP. Dużą dokładność uzyskano podczas działania klasyfikatora zaimplementowanego w mikrokontrolerze zamontowanym na kadłubie drona. Sygnałami wyjściowymi były odczyty z czujników umieszczonych na ramionach wielowirnikowca. W procesie diagnostyki żadne urządzenia zewnętrzne nie były wykorzystywane. Skutecznie przeprowadzano detekcję anomalii już na wczesnym stadium. Systemy detekcji z wysoką precyzją wykrywały występujące uszkodzenia od pierwszych cykli działania programu. Określenie rodzaju usterki, jej rozmiaru oraz miejsca wystąpienia zrealizowano dzięki zastosowaniu metod działających w ramach FDI oraz FDD. Uzyskano wysoką skuteczność we wskazywaniu lokalizacji śmigieł z wyszczerbioną krawędzią oraz wygiętą lub utraconą końcówką śmigła. Ten ostatni typ uszkodzenia rozpatrywano przy uwzględnieniu defektu występującego w dwóch rozmiarach.

Sygnały wyjściowe z klasyfikatora uszkodzeń stanowią cenną informację dla systemów sterowania odpornego na uszkodzenia. Choć prowadzone w ramach rozprawy badania nie obejmowały tej problematyki, opracowane modele dostarczać mogą informację wystarczającą do zastosowania odpowiedniego protokołu działania systemu FTC. Może on pozwolić na kontynuowanie misji w warunkach występujących uszkodzeń lub – w razie potrzeby – przeprowadzić procedurę awaryjnego lądowania drona.

Wyniki przeprowadzonych eksperymentów pokazują, że z użyciem niskokosztowych czujników można skutecznie wykrywać niebezpieczne dla działania drona uszkodzenia. Jedyne nieznacznie zwiększając masę całego układu, poprzez umieszczenie odpowiednich sensorów w pobliżu śmigieł oraz wydzielając niewielką część mocy obliczeniowej już znajdującego się na pokładzie systemu wbudowanego, możliwe jest wdrożenie metod diagnostyki uszkodzeń także w komercyjnie dostępnych BSP.

9.1 Istotny wkład w tematykę diagnozowania uszkodzeń BSP

Prace naukowe innych badaczy zajmujących się problematyką detekcji uszkodzeń BSP obarczone są różnymi ograniczeniami. Spory odsetek artykułów [23, 155, 47, 64, 34, 93, 41, 91, 94, 90, 114, 151] opisuje eksperymenty przeprowadzane na symulowanych danych. Nawet jeśli część pomiarów jest rzeczywistych, zwykle same uszkodzenia są już sztucznie generowane przez używane oprogramowanie [111, 141, 145, 156, 82, 5]. Niektórzy naukowcy podejmują próby wykonania badań podczas lotów z rzeczywistymi uszkodzeniami, ale najczęściej są one przeprowadzane dla niewielkiej liczby defektów [58, 10, 11, 123]. Zazwyczaj dotyczą one jednego rodzaju anomalii, czasami w różnych rozmiarach [43, 44, 11, 123]. Trudno jednak znaleźć prace odnoszące się do różnych scenariuszy możliwych do wystąpienia kombinacji uszkodzeń. Część badań obejmuje loty jedynie na uwięzi, a nawet z pojazdem unieruchomionym na podłożu, co uniemożliwia uwzględnienie dynamiki występującej podczas wykonywania manewrów. Zdarza się, że połączenie pomiędzy latającym dronem a aparaturą pomiarowo-obliczeniową jest realizowane przewodowo, mocno ograniczając swobodę manewrowania BSP.

Zaprezentowane w rozprawie metody w dużym stopniu pozbawione są ww. wad. Przedstawione eksperymenty były wykonywane podczas swobodnych lotów różnymi konstrukcjami czterowirnikowych BSP. Loty przeprowadzano tak w laboratorium, jak i na otwartej przestrzeni, zawsze wykonując manewry wznoszenia, opadania, gwałtownych zmian kierunku i orientacji, loty po spirali oraz innych trajektoriach spotykanych podczas typowej eksploatacji drona. Uszkodzone śmigła pochodziły z lotów, w których dochodziło do rzeczywistych kolizji, albo były celowo uszkodzone w sposób jak najbardziej przypominający prawdziwe defekty.

Nowatorskie podejście do detekcji uszkodzeń prezentują eksperymenty przedstawione w podrozdziale 7.6. Dotyczą one wykrywania uszkodzeń nowych, z którymi używany model wcześniej nie miał do czynienia. Trudno jest wytrenować klasyfikator uszkodzeń, przygotowując go w sposób bezpośredni do rozpoznawania wszelkich możliwych do zaistnienia awarii i defektów, ponieważ liczba potencjalnych wad jest w zasadzie nieograniczona. Dlatego stosowany model powinien być w stanie skutecznie rozpoznać także nieprawidłowość, na okoliczność wystąpienia której nie był jawnie projektowany.

Największą wartością dodaną, wniesioną do nowoczesnych metod FDD w ramach badań przeprowadzonych nad opracowaniem niniejszej rozprawy, jest niewątpliwie wykorzystanie technik umożliwiających przeprowadzanie skutecznej detekcji i klasyfikacji uszkodzeń w czasie rzeczywistym, bez konieczności korzystania z obliczeń wykonywanych poza jednostką latającą. Brak potrzeby przesyłania danych do wydajnych obliczeniowo systemów komputerowych (stacji naziemnych, chmury itp.) pozwala na natychmiastowe uzyskanie wyniku przeprowadzonej diagnostyki, która realizowana jest w trybie ciągłym podczas lotu. Niektóre badania wykazują dużą precyzję identyfikacji, jednocześnie wskazując na możliwość zastosowania użytych technik na pokładzie drona. Przykładowo, wnioski wskazane w rozprawie [15] odnoszą się do potencjału wykorzystania niektórych z zaproponowanych metod w przetwarzaniu online, jednakże nie przedstawiają rzeczywistych testów wykonanych w takich warunkach. Tego typu weryfikacja została przeprowadzona dla metod prezentowanych w niniejszej dysertacji. Na dzień ukazania się artykułu [103], w którym opisano wyniki eksperymentów przeprowadzanych online podczas lotu, nie były dostępne publikacje dotyczące podobnych rozwiązań. Niejednokrotnie inni autorzy posługują się terminem „czas rzeczywisty”, ale zwykle nie jest on ściśle zdefiniowany i mocno odbiega od założeń poczynionych w podrozdziale 1.4.

Dodatkowym wkładem w rozwój diagnostyki BSP jest repozytorium PADRE. Zawiera ono surowe pomiary z czujników, które zostały zebrane podczas lotów dronami wyposażonymi w różne kombinacje niesprawnych śmigieł. Dane zamieszczone w PADRE mogą być dowolnie przetwarzane na potrzeby badań nad identyfikacją uszkodzeń. Dla ułatwienia prac udostępniono także dane przetworzone, gotowe do wykorzystania na etapie trenowania i walidacji przygotowywanych klasyfikatorów. Powinno to znacząco przyspieszyć badania przede wszystkim w jej początkowych fazach.

W literaturze brakuje gotowych miar jakości pozwalających porównać skuteczność działania modeli przeprowadzających diagnostykę w ramach systemów charakteryzujących się różną złożonością. Wyzwanie stanowi też znalezienie sposobu na łatwe porównanie klasyfikatorów, które cechuje inny czas przetwarzania, a zatem różne wartości opóźnienia w procesie dostarczenia wyniku klasyfikacji. Zaproponowane w podrozdziale 7.1 nowe wskaźniki wypełniają tę lukę. Uwzględniając uzyskiwaną dokładność, pozwalają wprost porównać modele dokonujące podziału stanu badanego układu na różne liczby klas (Q_{AC}), a także pracujące z różnym opóźnieniem (Q_{ACT}). Ten

drugi parametr jest szczególnie istotny w przypadku systemów, w których czas przetwarzania jest krytyczny. Do tych systemów z pewnością należą bezzałogowe statki powietrzne.

9.2 Wnioski końcowe

W toku realizacji prac nad detekcją uszkodzeń BSP pojawił się szereg spostrzeżeń i wniosków. Jedną z najbardziej zaskakujących obserwacji był fakt, iż pomiary w dziedzinie czasu, które nie podlegały żadnemu procesowi ekstrakcji cech, w kontekście detekcji i klasyfikacji uszkodzeń okazały się nie mniej użyteczne od sygnałów, dla których wyznaczano sygnatury częstotliwościowe. Pomimo bardzo prostej formy danych w dziedzinie czasu, klasyfikatory korzystające z tego typu sygnałów uzyskiwały wysoką skuteczność w zakresie identyfikacji rodzaju i miejsca występowania uszkodzenia. Dodatkowo takie podejście w znaczący sposób zmniejszało złożoność obliczeniową modelu, a co za tym idzie, pozwalało wyraźnie skrócić czas potrzebny na przeprowadzenie procesu wnioskowania. Zastosowanie sygnałów dziedziny czasu, w tym o bardzo wąskich oknach pomiarowych, zawierających zaledwie po dwa kolejne odczyty z osi czujnika, umożliwiło zaprojektowanie systemu FDI pracującego w czasie rzeczywistym, bez wyraźnego spadku dokładności klasyfikacji.

Przeprowadzone badania wykazały, że określanie miejsca, rodzaju i rozmiaru uszkodzeń śmigieł bezzałogowych pojazdów latających można przeprowadzać z wykorzystaniem sygnałów z niskokosztowych czujników inercyjnych bądź mikrofonów. Zastosowanie wydajnego algorytmu przetwarzania sygnałów pomiarowych we współczesnych, standardowych mikrokontrolerach, umożliwia realizację procedury diagnostyki uszkodzeń BSP. Korzystając z tak przygotowanego systemu wbudowanego, możliwe jest precyzyjne diagnozowanie w czasie rzeczywistym defektów śmigieł, realizowane podczas lotu wielowirnikowego drona.

Diagnostyka uszkodzeń BSP jest procesem złożonym. Opisane w rozprawie metody i techniki nie wyczerpują potencjalnego spektrum możliwości, jakie daje dzisiejszy stan wiedzy. Podczas wyszczególnionych w dysertacji badaniach korzystano przede wszystkim z dwóch reprezentacji przetwarzanych sygnałów: czasowej i częstotliwościowej. Jednakże bardzo obiecujące wyniki ukazały również dane czasowo-częstotliwościowe. Z tego względu

na dalszych etapach rozwoju prac dotyczących diagnostyki uszkodzeń planowane jest dokładniejsze zbadanie m.in. sygnałów DWT, a także STFT i innych, w szczególności pod kątem możliwości ich przetwarzania w systemach wbudowanych.

Rozprawa dotyczy badań prowadzonych na czterowirnikowych BSP. W związku z faktem, że pojawia się coraz więcej rozwiązań konstrukcyjnych dronów, w tym również nietypowych, planuje się podjęcie prac badawczych nad uszkodzeniami innych typów robotów latających.

Rozważa się również zaprojektowanie i przeprowadzenie eksperymentów, w których uszkodzenie śmigieł następuje w trakcie lotu BSP. Taki zbiór danych, bazujący na różnych kolizjach BSP z elementami otoczenia, prowadzących do różnych uszkodzeń śmigieł, pozwoliłby na jeszcze precyzyjniejsze określenie realnego opóźnienia pomiędzy momentem pojawienia się anomalii a jej wykryciem, zlokalizowaniem i właściwym zidentyfikowaniem.

Bibliografia

- [1]3DR. *Solo operation manual*. URL: https://www.wvu.edu/faculty/wallin/esci497_uas/readings/3dr_solo/Solo_FAA_6_22_operations_manual.pdf (term. wiz. 3 czer. 2024) (cyt. na s. 40).
- [2]AeroLAB. *PADRE — Propeller anomaly data repository for UAVs various rotor fault configurations*. URL: <https://uav.put.poznan.pl/archives/763> (term. wiz. 10 grud. 2024) (cyt. na s. 33).
- [3]AeroLAB. *Repozytorium PADRE*. URL: https://github.com/AeroLabPUT/UAV_measurement_data/ (term. wiz. 27 list. 2024) (cyt. na s. 18, 33).
- [4]AeroLAB. *Sensory data acquisition during drone flight in real-world conditions for the PADRE database*. URL: <https://uav.put.poznan.pl/archives/788> (term. wiz. 10 grud. 2024) (cyt. na s. 33).
- [5]A. Altinors, F. Yol i O. Yaman. „A sound based method for fault detection with statistical feature extraction in UAV motors“. *Applied Acoustics* 183 (2021), s. 1–14 (cyt. na s. 3, 123).
- [6]N. Amarasingam, A. S. Ashan Salgadoe, K. Powell, L. F. Gonzalez i S. Natarajan. „A review of UAV platforms, sensors, and applications for monitoring of sugarcane crops“. *Remote Sensing Applications: Society and Environment* 26 (2022), s. 1–22 (cyt. na s. 1).
- [7]S.-i. Amari. „Backpropagation and stochastic gradient descent method“. *Neurocomputing* 5.4-5 (1993), s. 185–196 (cyt. na s. 60).
- [8]M. Andrejašic. „MEMS accelerometers“. W: *University of Ljubljana. Faculty for mathematics and physics, Department of physics, Seminar*. T. 49. 2008, s. 1–17 (cyt. na s. 22).
- [9]A. Baldini, L. D’Alleva, R. Felicetti i in. „UAV-FD: A dataset for actuator fault detection in multirotor drones“. W: *2023 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2023, s. 998–1004 (cyt. na s. 18).
- [10]A. Baldini, R. Felicetti, F. Ferracuti i in. „Real-time propeller fault detection for multirotor drones based on vibration data analysis“. *Engineering Applications of Artificial Intelligence* 123 (2023), s. 1–13 (cyt. na s. 16, 118, 123).

- [11]B. Bartoszewski, K. Jarzyna i J. Baranowski. „Diagnosis of mechanical rotor faults in drones using functional Gaussian mixture classifier“. *Aerospace* 11.9 (2024) (cyt. na s. 17, 118, 123).
- [12]L. Bin, W. Liang i Y. Guosheng. *A graph total variation regularized softmax for text generation*. 2021. arXiv: 2101.00153 (cyt. na s. 59).
- [13]H. S. Black i J. Edson. „Pulse code modulation“. *Transactions of the American Institute of Electrical Engineers* 66.1 (1947), s. 895–899 (cyt. na s. 30).
- [14]A. Blumberg. „Logistic growth rate functions“. *Journal of Theoretical Biology* 21.1 (1968), s. 42–44 (cyt. na s. 58).
- [15]A. Bondyra. „Automatyczna diagnostyka uszkodzeń wirników nośnych robota latającego“. Prac. dokt. Poznań: Politechnika Poznańska, 2023 (cyt. na s. 4, 33, 124).
- [16]A. Bondyra, S. Gardecki, P. Gąsior i A. Kasiński. „Falcon: A compact multirotor flying platform with high load capability“. W: *Progress in Automation, Robotics and Measuring Techniques*. Red. R. Szewczyk, C. Zieliński i M. Kaliczyńska. Cham: Springer International Publishing, 2015, s. 35–44 (cyt. na s. 88).
- [17]A. Bondyra, P. Gąsior, S. Gardecki i A. J. Kasinski. „Development of the sensory network for the vibration-based fault detection and isolation in the multirotor UAV propulsion system“. W: *International Conference on Informatics in Control, Automation and Robotics (ICINCO)*. 2018, s. 112–119 (cyt. na s. 15, 88, 118).
- [18]A. Bondyra, P. Gąsior, S. Gardecki i A. Kasiński. „Fault diagnosis and condition monitoring of UAV rotor using signal processing“. W: *2017 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*. IEEE. 2017, s. 233–238 (cyt. na s. 14).
- [19]A. Bondyra, M. Kołodziejczak, R. Kulikowski i W. Giernacki. „An acoustic fault detection and isolation system for multirotor UAV“. *Energies* 15.11 (2022) (cyt. na s. 15, 18, 118).
- [20]D. Chicco i G. Jurman. „A statistical comparison between Matthews correlation coefficient (MCC), prevalence threshold, and Fowlkes–Mallows index“. *Journal of Biomedical Informatics* 144 (2023), s. 1–7 (cyt. na s. 69).
- [21]D. Chicco, N. Tötsch i G. Jurman. „The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation“. *BioData Mining* 14 (2021), s. 1–22 (cyt. na s. 69).
- [22]F. Chollet. *Deep learning with Python*. New York: Manning Publications Co., 2017 (cyt. na s. 38, 56, 57).

- [23]E. D'Amato, M. Mattei, I. Notaro i V. Scordamaglia. „UAV sensor FDI in duplex attitude estimation architectures using a set-based approach“. *IEEE Transactions on Instrumentation and Measurement* 67.10 (2018), s. 2465–2475 (cyt. na s. 123).
- [24]E. D'Amato, V. A. Nardi, I. Notaro i V. Scordamaglia. „A particle filtering approach for fault detection and isolation of UAV IMU sensors: Design, implementation and sensitivity analysis“. *Sensors* 21.9 (2021), s. 1–24 (cyt. na s. 3).
- [25]A. Dąbrowski. *Przetwarzanie sygnałów przy użyciu procesorów sygnałowych*. Poznań: Wydawnictwo Politechniki Poznańskiej, 1997 (cyt. na s. 88).
- [26]A. Dąbrowski, P. Pawłowski, R. Weychan i in. „Real-time watermarking of one side of telephone conversation for speaker segmentation“. *Electrical Review* 88.6 (2012), s. 36–41 (cyt. na s. 103).
- [27]Y. Debele, H.-Y. Shi, A. Wondosen, T.-W. Ku i B.-S. Kang. „Deep learning-based robust actuator fault detection and isolation scheme for highly redundant multicopter UAVs“. *Drones* 7.7 (2023), s. 1–26 (cyt. na s. 15).
- [28]A. Demirkaya, J. Chen i S. Oymak. „Exploring the role of loss functions in multiclass classification“. W: *2020 54th Annual Conference on Information Sciences and Systems (CISS)*. IEEE. 2020, s. 1–5 (cyt. na s. 69).
- [29]R. J. Dilligan. „Binary numbers, ASCII code, and the Turing machine“. *Computing in the Web Age: A Web-Interactive Introduction* (2002), s. 45–88 (cyt. na s. 31).
- [30]S. Duan, X. Lu, H. Wu i in. „Unmanned aerial vehicle wireless power transfer system for long-distance transmission line patrol“. W: *2022 IEEE International Power Electronics and Application Conference and Exposition (PEAC)*. 2022, s. 1324–1329 (cyt. na s. 1).
- [31]J. Dudczyk, R. Czyba i K. Skrzypczyk. „Multi-Sensory Data Fusion in Terms of UAV Detection in 3D Space“. *Sensors* 22.12 (2022), s. 1–23 (cyt. na s. 1).
- [32]M. Dukhan i A. Ablavatski. „Two-pass softmax algorithm“. *2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)* (2020), s. 386–395 (cyt. na s. 59).
- [33]H. Eskandaripour i E. Boldsaikhan. „Last-mile drone delivery: Past, present, and future“. *Drones* 7.2: 77 (2023), s. 1–19 (cyt. na s. 1).
- [34]A. Faraji, Z. Nejati i M. Abedi. „Actuator faults estimation for a helicopter UAV in the presence of disturbances“. *Journal of Control, Automation and Electrical Systems* 31.5 (2020), s. 1153–1164 (cyt. na s. 123).
- [35]M. Feldman. „Hilbert transform in vibration analysis“. *Mechanical Systems and Signal Processing* 25.3 (2011), s. 735–802 (cyt. na s. 104).

- [36]K. R. M. Fernando i C. P. Tsokos. „Dynamically weighted balanced loss: Class imbalanced learning and confidence calibration of deep neural networks“. *IEEE Transactions on Neural Networks and Learning Systems* (2021), s. 1–12 (cyt. na s. 59).
- [37]G. K. Fourlas i G. C. Karras. „A survey on fault diagnosis and fault-tolerant control methods for unmanned aerial vehicles“. *Machines* 9.9 (2021), s. 1–34 (cyt. na s. 3).
- [38]G. K. Fourlas i G. C. Karras. „A survey on fault diagnosis methods for UAVs“. W: *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2021, s. 394–403 (cyt. na s. 3).
- [39]F. D. Franco, C. Rametta, M. Russo i M. Vaccaro. „An hardware implementation of the softmax function on FPGA“. W: *International Conference for Young Researchers in Informatics, Mathematics, and Engineering*. 2020, s. 21–25 (cyt. na s. 59).
- [40]C. Friedrich, A. Lechler i A. Verl. „Autonomous systems for maintenance tasks – requirements and design of a control architecture“. *Procedia Technology* 15 (2014), s. 596–605 (cyt. na s. 6).
- [41]X. Fu i X. Geng. „Fault estimation and robust fault-tolerant control for singular Markov switching systems with mixed time-delays and UAV applications“. *Journal of Control Engineering and Applied Informatics* 23.2 (2021), s. 53–66 (cyt. na s. 123).
- [42]B. Gao i L. Pavel. *On the properties of the softmax function with application in game theory and reinforcement learning*. 2018. arXiv: 1704.00805 (cyt. na s. 59).
- [43]B. Ghalamchi, Z. Jia i M. W. Mueller. „Real-time vibration-based propeller fault diagnosis for multicopters“. *IEEE/ASME Transactions on Mechatronics* 25.1 (2019), s. 395–405 (cyt. na s. 123).
- [44]M. S. Gomez, A.-K. Koschlik, E. Arts i F. Raddatz. „Non-destructive evaluation of the condition of a UAV's propellers by means of acoustics“. W: *NDE 4.0, Predictive Maintenance, and Communication and Energy Systems in a Globally Networked World*. T. 12049. 2022, s. 22–30 (cyt. na s. 3, 123).
- [45]B. Grindley, K. Phillips, K. J. Parnell i in. „Over a decade of UAV incidents: A human factors analysis of causal factors“. *Applied Ergonomics* 121 (2024), s. 1–9 (cyt. na s. 1).
- [46]S. Gururajan, K. Mitchell i W. Ebel. „Flights of a multirotor UAS with structural faults: Failures on composite propeller(s)“. *Data* 4.3 (2019) (cyt. na s. 18).

- [47]J. A. Guzmán-Rabasa, F. R. López-Estrada, B. M. González-Contreras i in. „Actuator fault detection and isolation on a quadrotor unmanned aerial vehicle modeled as a linear parameter-varying system“. *Measurement and Control* 52.9-10 (2019), s. 1228–1239 (cyt. na s. 123).
- [48]L. A. Al-Haddad, W. Giernacki, A. A. Shandookh, A. A. Jaber i R. Puchalski. „Vibration signal processing for multirotor UAVs fault diagnosis: Filtering or multiresolution analysis?“ *Eksploatacja i Niezawodność – Maintenance and Reliability* 26.1 (2024), s. 1–19 (cyt. na s. 3, 103).
- [49]S. H. Han, T. Rahim i S. Y. Shin. „Detection of faults in solar panels using deep learning“. W: *2021 International Conference on Electronics, Information, and Communication (ICEIC)*. 2021, s. 1–4 (cyt. na s. 1).
- [50]A. Hasan i T. A. Johansen. „Model-based actuator fault diagnosis in multirotor UAVs“. W: *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2018, s. 1017–1024 (cyt. na s. 6).
- [51]S. Haugse. „Investigation of UAV related incidents and accidents“. Prac. mag. Tromsø: UiT The Arctic University of Norway, 2022 (cyt. na s. 1, 3).
- [52]C. E. Heil i D. F. Walnut. „Continuous and discrete wavelet transforms“. *SIAM Review* 31.4 (1989), s. 628–666 (cyt. na s. 103).
- [53]Y. Ho i S. Wookey. „The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling“. *IEEE Access* 8 (2020), s. 4806–4813 (cyt. na s. 58).
- [54]V. T. Hoang, M. D. Phung, T. H. Dinh i Q. P. Ha. „System architecture for real-time surface inspection using multiple UAVs“. *IEEE Systems Journal* 14.2 (2020), s. 2925–2936 (cyt. na s. 1).
- [55]K. Hornik, M. Stinchcombe i H. White. „Multilayer feedforward networks are universal approximators“. *Neural Networks* 2.5 (1989), s. 359–366 (cyt. na s. 85).
- [56]J. Hu, H. Niu, J. Carrasco, B. Lennox i F. Arvin. „Fault-tolerant cooperative navigation of networked UAV swarms for forest fire monitoring“. *Aerospace Science and Technology* 123 (2022), s. 1–12 (cyt. na s. 1).
- [57]N. E. Huang i Z. Wu. „A review on Hilbert-Huang transform: Method and its applications to geophysical studies“. *Reviews of Geophysics* 46.2 (2008), s. 1–23 (cyt. na s. 104).
- [58]G. Iannace, G. Ciaburro i A. Trematerra. „Fault diagnosis for UAV blades using artificial neural network“. *Robotics* 8.3 (2019), s. 1–17 (cyt. na s. 13, 118, 123).
- [59]Infineon. *Mikrofony MEMS – nowa jakość w rejestracji dźwięku*. 2019. URL: <https://elektronikab2b.pl/technika/50827-mikrofony-mems-nowa-jakosc-w-rejestracji-dzwieku> (term. wiz. 26 grud. 2024) (cyt. na s. 24).

- [60]InvenSense. *MPU-9250 product specification revision 1.1*. 2016. URL: <https://invensense.tdk.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf> (term. wiz. 28 wrz. 2023) (cyt. na s. 42).
- [61]InvenSense. *MPU-6500 product specification revision 1.3*. 2020. URL: <http://invensense.wpenginpowered.com/wp-content/uploads/2020/06/PS-MPU-6500A-01-v1.3.pdf> (term. wiz. 30 paź. 2023) (cyt. na s. 35).
- [62]R. Isermann i P. Ballé. „Trends in the application of model based fault detection and diagnosis of technical processes“. W: *IFAC Proceedings Volumes*. T. 29. 1. 1996, s. 6325–6336 (cyt. na s. 6, 11).
- [63]Jarmo Lähdevaara. *Simple fast Fourier transform algorithms in C*. URL: <http://www.guitarscience.net/papers/fftalg.pdf> (term. wiz. 25 sierp. 2024) (cyt. na s. 88).
- [64]P. Kantue i J. O. Pedro. „Integrated fault detection and diagnosis of an unmanned aerial vehicle using time difference of arrival“. W: *2020 24th International Conference on System Theory, Control and Computing (ICSTCC)*. IEEE. 2020, s. 336–342 (cyt. na s. 123).
- [65]S. S. Katta i E. K. Viegas. „Towards a reliable and lightweight onboard fault detection in autonomous unmanned aerial vehicles“. W: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, s. 1284–1290 (cyt. na s. 18).
- [66]A. Khan i L. Brown. „Recreational drone-related injuries in children: A review of national electronic injury surveillance system (NEISS) data“. *Cureus* 13.6 (2021), s. 1–5 (cyt. na s. 1).
- [67]S. I. Khan, Z. Qadir, H. S. Munawar i in. „UAVs path planning architecture for effective medical emergency response in future networks“. *Physical Communication* 47 (2021), s. 1–13 (cyt. na s. 1).
- [68]B.-H. Kim i H.-S. Lee. „Acoustical-thermal noise in a capacitive MEMS microphone“. *IEEE Sensors Journal* 15.12 (2015), s. 6853–6860 (cyt. na s. 24).
- [69]M. Kołodziejczak, R. Puchalski, A. Bondyra, S. Śladic i W. Giernacki. „Toward lightweight acoustic fault detection and identification of UAV rotors“. W: *2023 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2023, s. 990–997 (cyt. na s. 3).
- [70]F. A. A. Konert i B. T. Balcerzak. „Legal and ethical aspects of rules for the operation of autonomous unmanned aircraft with artificial intelligence“. W: *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE. 2021, s. 602–609 (cyt. na s. 1).
- [71]J. Korbicz, J. M. Kościelny, K. Zdzisław i C. Wojciech. *Diagnostyka procesów. Modele. Metody sztucznej inteligencji. Zastosowania*. Warszawa: Wydawnictwa Naukowo-Techniczne, 2002 (cyt. na s. 5, 6).

- [72]P. Korohoda i A. Dąbrowski. „Wavelet-like decomposition stage with windowed filters defined for the discrete trigonometric transforms (DTTs)“. *Przegląd Elektrotechniczny* 88.6 (2012), s. 30–35 (cyt. na s. 103).
- [73]I. Kouretas i V. Paliouras. „Hardware implementation of a softmax-like function for deep learning“. *Technologies* 8.3 (2020), s. 1–20 (cyt. na s. 59).
- [74]P. Krzysztof. *Mikrokontroler – iNEMO: STM32F103 i MEMS-y*. 2012. URL: <https://mikrokontroler.pl/2012/07/05/inemo-stm32f103-i-mems-y/> (term. wiz. 26 grud. 2024) (cyt. na s. 22, 23).
- [75]T. Kulczak, B. Szczerbo i S. Brock. „Wykorzystanie akcelerometru i żyroskopu MEMS do pomiaru drgań w napędzie bezpośrednim o złożonej strukturze mechanicznej“. *Poznan University of Technology Academic Journals. Electrical Engineering* 87 (2016), s. 245–255 (cyt. na s. 22, 23).
- [76]M. J. Kusner i J. M. Hernández-Lobato. *GANS for sequences of discrete elements with the Gumbel-softmax distribution*. 2016. arXiv: 1611.04051 (cyt. na s. 59).
- [77]M. Lapin, M. Hein i B. Schiele. „Analysis and optimization of loss functions for multiclass, top-k, and multilabel classification“. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.7 (2018), s. 1533–1554 (cyt. na s. 69).
- [78]P. Li i H. Liu. „Binary decomposition for multi-class classification problems: Development and applications“. W: *2023 International Conference on Machine Learning and Cybernetics (ICMLC)*. IEEE. 2023, s. 452–457 (cyt. na s. 69).
- [79]Y. Li, M. Liu i D. Jiang. „Application of unmanned aerial vehicles in logistics: A literature review“. *Sustainability* 14.21 (2022) (cyt. na s. 1).
- [80]R. G. Lyons. *Wprowadzenie do cyfrowego przetwarzania sygnałów*. Warszawa: Wydawnictwa Komunikacji i Łączności, 1999 (cyt. na s. 88, 89).
- [81]K. Maenaka. „MEMS inertial sensors and their applications“. W: *2008 5th International Conference on Networked Sensing Systems*. 2008, s. 71–73 (cyt. na s. 22, 23).
- [82]H. Maqsood, M. Taimoor, Z. Ullah, N. Ali i M. Sohail. „Novel sensor fault detection and isolation for an unmanned aerial vehicle“. W: *2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST)*. IEEE. 2021, s. 486–493 (cyt. na s. 123).
- [83]T. Marciniak, A. Dąbrowski, R. Puchalski, D. Dratwiak i W. Marciniak. „Zastosowanie mikrokontrolera STM32F410 do prezentacji zagadnień cyfrowego przetwarzania sygnałów“. *Przegląd Elektrotechniczny* 95 (2019), s. 118–120 (cyt. na s. 3).
- [84]R. C. Maron, M. Weichenthal, J. S. Utikal i in. „Systematic outperformance of 112 dermatologists in multiclass skin cancer image classification by convolutional neural networks“. *European Journal of Cancer* 119 (2019), s. 57–65 (cyt. na s. 69).

- [85]M. T. Marty. „Eliminating artifacts in electrospray deconvolution with a soft-max function“. *Journal of the American Society for Mass Spectrometry* 30.10 (2019), s. 2174–2177 (cyt. na s. 59).
- [86]K. Masalimov, T. Muslimov, E. Kozlov i R. Munasypov. „CrazyPAD: A dataset for assessing the impact of structural defects on nano-quadcopter performance“. *Data* 9.6 (2024), s. 1–18 (cyt. na s. 18).
- [87]MathWorks. *FFT – Fast Fourier transform*. URL: <https://www.mathworks.com/help/matlab/ref/fft.html> (term. wiz. 25 sierp. 2024) (cyt. na s. 38, 88).
- [88]M. Mazzoleni, G. Di Rito i F. Previdi. „Fault diagnosis and condition monitoring approaches“. W: *Electro-Mechanical Actuators for the More Electric Aircraft*. Cham: Springer International Publishing, 2021, s. 87–117 (cyt. na s. 6).
- [89]S. A. H. Mohsan, N. Q. H. Othman, Y. Li, M. H. Alsharif i M. A. Khan. „Unmanned aerial vehicles (UAVs): Practical aspects, applications, open challenges, security issues, and future trends“. *Intelligent Service Robotics* 16.1 (2023), s. 109–137 (cyt. na s. 1).
- [90]S. Mokhtari, A. Abbaspour, K. K. Yen i A. Sargolzaei. „Neural network-based active fault-tolerant control design for unmanned helicopter with additive faults“. *Remote Sensing* 13.12 (2021), s. 1–14 (cyt. na s. 123).
- [91]Z. Nejati, A. Faraji i M. Abedi. „Robust three stage central difference Kalman filter for helicopter unmanned aerial vehicle actuators fault estimation“. *International Journal of Engineering* 34.5 (2021), s. 1290–1296 (cyt. na s. 123).
- [92]A. V. Oppenheim i R. W. Schaffer. *Cyfrowe przetwarzanie sygnałów*. Warszawa: Wydawnictwa Komunikacji i Łączności, 1979 (cyt. na s. 87).
- [93]A. Y. Ouadine, M. Mjahed, H. Ayad i A. El Kari. „UAV quadrotor fault detection and isolation using artificial neural network and Hammerstein-Wiener model“. *Studies in Informatics and Control* 29.3 (2020), s. 317–328 (cyt. na s. 123).
- [94]K. H. Park, E. Park i H. K. Kim. „Unsupervised fault detection on unmanned aerial vehicles: Encoding and thresholding approach“. *Sensors* 21.6 (2021), s. 1–17 (cyt. na s. 123).
- [95]Parrot. *Bebop-pro 3D modeling, all-in-one drone solution for 3D modeling*. URL: <https://www.parrot.com> (term. wiz. 3 czer. 2024) (cyt. na s. 35).
- [96]F. Pasandideh, J. P. J. da Costa, R. Kunst i in. „A review of flying ad hoc networks: Key characteristics, applications, and wireless technologies“. *Remote Sensing* 14.18 (2022), s. 1–25 (cyt. na s. 1).
- [97]R. Patton. „Fault detection and diagnosis in aerospace systems using analytical redundancy“. *Computing & Control Engineering Journal* 2.3 (1991), s. 127–136 (cyt. na s. 11).

- [98]H. Portalska i M. Portalski. „Akustyczne możliwości wspierania rehabilitacji chorych na SM“. W: *Ergonomia niepełnosprawnym. Jakość życia*. Łódź: Wydawnictwo Politechniki Łódzkiej, 2001. Rozd. XXV, s. 202–219 (cyt. na s. 103).
- [99]R. Puchalski, A. Bondyra, W. Giernacki i Y. Zhang. „Actuator fault detection and isolation system for multirotor unmanned aerial vehicles“. W: *2022 26th International Conference on Methods and Models in Automation and Robotics (MMAR)*. 2022, s. 364–369 (cyt. na s. 3, 88, 91).
- [100]R. Puchalski i W. Giernacki. „UAV fault detection methods, state-of-the-art“. *Drones* 6.11 (2022), s. 1–39 (cyt. na s. 3, 6, 7, 11, 12, 15).
- [101]R. Puchalski i W. Giernacki. „Movement pattern recognition in boxing using raw inertial measurements“. W: *Optimization, Learning Algorithms and Applications*. Red. A. I. Pereira, A. Mendes, F. P. Fernandes i in. Cham: Springer Nature Switzerland, 2024, s. 19–34 (cyt. na s. 58).
- [102]R. Puchalski, W. Giernacki i Q. Ha. „Real-time UAV fault detection and classification using measurement data from the PADRE database“. W: *2024 IEEE/SICE International Symposium on System Integration (SII)*. 2024, s. 663–668 (cyt. na s. 3, 56, 111–114).
- [103]R. Puchalski, Q. Ha, W. Giernacki, H. A. D. Nguyen i L. V. Nguyen. „PADRE – a repository for research on fault detection and isolation of unmanned aerial vehicle propellers“. *Journal of Intelligent & Robotic Systems* 110.74 (2024), s. 1–22 (cyt. na s. 29, 34, 36, 41, 47–52, 64, 70, 72, 124).
- [104]R. Puchalski, M. Kołodziejczak, A. Bondyra, J. Rao i W. Giernacki. „PADRE – propeller anomaly data repository for UAVs various rotor fault configurations“. W: *2023 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2023, s. 982–989 (cyt. na s. 25).
- [105]J. Qiao, Z. Liu i Y. Zhang. „Gain scheduling based PID control approaches for path tracking and fault tolerant control of a quad-rotor UAV“. *International Journal of Mechanical Engineering and Robotics Research* 7.4 (2018), s. 401–408 (cyt. na s. 3).
- [106]P. Ramesh i J. M. L. Jeyan. „Comparative analysis of the impact of operating parameters on military and civil applications of mini unmanned aerial vehicle (UAV)“. W: *AIP Conference Proceedings*. T. 2311. 030034. AIP Publishing LLC. 2020, s. 1–10 (cyt. na s. 1).
- [107]G. Rilling, P. Flandrin, P. Goncalves i in. „On empirical mode decomposition and its algorithms“. W: *IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*. T. 3. 3. Grado: IEEE. 2003, s. 8–11 (cyt. na s. 104).
- [108]D. Rotondo, M. Pazera i M. Witczak. „Integrated fault estimation and fault-tolerant control for constrained LPV systems subject to bounded disturbances“. *Journal of Process Control* 145 (2025), s. 1–11 (cyt. na s. 55).

- [109]V. Sadhu, S. Zonouz i D. Pompili. „On-board deep-learning-based unmanned aerial vehicle fault cause detection and identification“. W: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, s. 5255–5261 (cyt. na s. 3, 14, 118).
- [110]S. S. Saha, S. S. Sandha i M. Srivastava. „Machine learning for microcontroller-class hardware: A review“. *IEEE Sensors Journal* 22.22 (2022), s. 21362–21390 (cyt. na s. 56).
- [111]M. Saied, T. Mahairy, C. Francis i in. „Differential flatness-based approach for sensors and actuators fault diagnosis of a multirotor UAV“. *IFAC-PapersOnLine* 52.16 (2019), s. 831–836 (cyt. na s. 123).
- [112]M. Saied, B. Lussier, I. Fantoni, H. Shraim i C. Francis. „Fault diagnosis and fault-tolerant control of an octorotor UAV using motors speeds measurements“. *IFAC-PapersOnLine* 50.1 (2017), s. 5263–5268 (cyt. na s. 13).
- [113]M. Saied, H. Shraim i C. Francis. „A review on recent development of multirotor UAV fault-tolerant control systems“. *IEEE Aerospace and Electronic Systems Magazine* 39.9 (2024), s. 146–180 (cyt. na s. 11).
- [114]M. Saied, A. R. Tabikh, C. Francis, H. Hamadi i B. Lussier. „An informational approach for fault tolerant data fusion applied to a UAV’s attitude, altitude, and position estimation“. *IEEE Sensors Journal* 21.24 (2021), s. 27766–27778 (cyt. na s. 3, 123).
- [115]B. A. S. van Schijndel, S. Sun i C. C. de Visser. „Fast loss of effectiveness detection on a quadrotor using onboard sensors and a Kalman estimation approach“. W: *2023 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2023, s. 1–8 (cyt. na s. 3).
- [116]P. Sedgwick. „Pearson’s correlation coefficient“. *British Medical Journal* 345 (2012), s. 1–2 (cyt. na s. 49).
- [117]K. Severson, P. Chaiwatanodom i R. D. Braatz. „Perspectives on process monitoring of industrial systems“. *Annual Reviews in Control* 42 (2016), s. 190–200 (cyt. na s. 6).
- [118]R. Shadiev i S. Yi. „A systematic review of UAV applications to education“. *Interactive Learning Environments* (2022), s. 1–30 (cyt. na s. 1).
- [119]I. N. da Silva, D. Hernane Spatti, R. Andrade Flauzino, L. H. B. Liboni i S. F. dos Reis Alves. „Artificial neural network architectures and training processes“. W: *Artificial Neural Networks : A Practical Course*. Cham: Springer International Publishing, 2017, s. 21–28 (cyt. na s. 59).
- [120]S. W. Smith. *Cyfrowe przetwarzanie sygnałów: Praktyczny poradnik dla inżynierów i naukowców*. Warszawa: Wydawnictwo BTC, 2007 (cyt. na s. 88).
- [121]J. Söderkvist. „Micromachined gyroscopes“. *Sensors and Actuators A: Physical* 43.1 (1994), s. 65–71 (cyt. na s. 23).

- [122]S. Sreenath, H. Malik, N. Husnu i K. Kalaichelavan. „Assessment and use of unmanned aerial vehicle for civil structural health monitoring“. *Procedia Computer Science* 170 (2020), s. 656–663 (cyt. na s. 1).
- [123]L. Steinhoff, A.-K. Koschlik, E. Arts i in. „Development of an acoustic fault diagnosis system for UAV propeller blades“. *CEAS Aeronautical Journal* 15 (2024), s. 1–13 (cyt. na s. 17, 118, 123).
- [124]STMicroelectronics. *MP34DT01-M MEMS audio sensor omnidirectional digital microphone*. 2014. URL: <https://www.st.com/en/audio-ics/mp34dt01-m.html> (term. wiz. 28 wrz. 2023) (cyt. na s. 43).
- [125]STMicroelectronics. *UM2372 STM32Cube PDM2PCM software library for the STM32F4/F7/H7 Series*. 2018. URL: https://www.st.com/resource/en/user_manual/um2372-stm32cube-pdm2pcm-software-library-for-the-stm32f4f7h7-series-stmicroelectronics.pdf (term. wiz. 22 paź. 2023) (cyt. na s. 30, 44).
- [126]STMicroelectronics. *AN5027 application note, interfacing PDM digital microphones using STM32 MCUs and MPUs*. 2019. URL: www.st.com/en/microcontrollers-microprocessors/stm32h743-753/documentation.html (term. wiz. 2 kw. 2024) (cyt. na s. 26).
- [127]STMicroelectronics. *UM2526, user manual, getting started with X-CUBE-AI expansion package for artificial intelligence (AI) Rev 6*. 2020. URL: <https://www.st.com/en/embedded-software/x-cube-ai.html#documentation> (term. wiz. 23 grud. 2024) (cyt. na s. 61).
- [128]STMicroelectronics. *AN4989 application note, STM32 microcontroller debug toolbox*. 2021. URL: https://www.st.com/resource/en/application_note/an4989-stm32-microcontroller-debug-toolbox-stmicroelectronics.pdf (term. wiz. 2 kw. 2024) (cyt. na s. 27).
- [129]STMicroelectronics. *STM32H742xI/G STM32H743xI/G, datasheet - production data*. 2022. URL: <https://www.st.com/resource/en/datasheet/stm32h743xi.pdf> (term. wiz. 2 kw. 2024) (cyt. na s. 26).
- [130]STMicroelectronics. *RM0433 reference manual, STM32H742, STM32H743/753 and STM32H750 value line advanced ARM-based 32-bit MCUs*. 2023. URL: www.st.com/en/microcontrollers-microprocessors/stm32h743-753/documentation.html (term. wiz. 2 kw. 2024) (cyt. na s. 26).
- [131]STMicroelectronics. *AN5200 application note, getting started with STM32H7 MCUs SDMMC host controller*. 2024. URL: www.st.com/en/microcontrollers-microprocessors/stm32h743-753/documentation.html (term. wiz. 2 kw. 2024) (cyt. na s. 26).
- [132]X. Sun, Y. Hu, Y. Qin i Y. Zhang. „Risk assessment of unmanned aerial vehicle accidents based on data-driven Bayesian networks“. *Reliability Engineering & System Safety* 248 (2024), s. 1–15 (cyt. na s. 1).

- [133]A. Susini. „A technocritical review of drones crash risk probabilistic consequences and its societal acceptance“. *Lnis* 7 (2015), s. 27–38 (cyt. na s. 1).
- [134]Ż. Szymańska-Małysa. „Wykorzystanie transformacji falkowej do poprawiania stosunku sygnał/zakłócenie w procesie przetwarzania danych sejsmicznych“. Prac. dokt. Kraków: Akademia Górniczo-Hutnicza, 2013 (cyt. na s. 103).
- [135]J. J. Tong, W. Zhang, F. Liao, C. F. Li i Y. F. Zhang. *Machine learning for UAV propeller fault detection based on a hybrid data generation model*. 2023. arXiv: 2302.01556 (cyt. na s. 16, 118).
- [136]A. Utsav, A. Abhishek, P. Suraj i R. K. Badhai. „An IoT based UAV network for military applications“. W: *2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. 2021, s. 122–125 (cyt. na s. 1).
- [137]A. Victor, H. Geremias dos Santos, G. F. S. Silva i in. „Predictive modeling of gestational weight gain: A machine learning multiclass classification study“. *BMC Pregnancy and Childbirth* 24.733 (2024), s. 1–11 (cyt. na s. 69).
- [138]S. Y. Vural i C. Hacızade. „Sensor/actuator fault detection, isolation and accommodation applied to UAV model“. *Journal of Aeronautics and Space Technologies* 9.2 (2016), s. 1–12 (cyt. na s. 11).
- [139]B. Wang, D. Liu, Y. Peng i X. Peng. „Multivariate regression-based fault detection and recovery of UAV flight data“. *IEEE Transactions on Instrumentation and Measurement* 69.6 (2019), s. 3527–3537 (cyt. na s. 3).
- [140]B. Wang, X. Peng, M. Jiang i D. Liu. „Real-time fault detection for UAV based on model acceleration engine“. *IEEE Transactions on Instrumentation and Measurement* 69.12 (2020), s. 9505–9516 (cyt. na s. 3).
- [141]R. Wang, C. Zhao, Y. Bai, W. Du i J. Wang. „An actuator fault detection and reconstruction scheme for hex-rotor unmanned aerial vehicle“. *IEEE Access* 7 (2019), s. 93937–93951 (cyt. na s. 123).
- [142]Z. Wang, X. Ning i M. Blaschko. „Jaccard metric losses: Optimizing the Jaccard index with soft labels“. *Advances in Neural Information Processing Systems* 36 (2024), s. 1–27 (cyt. na s. 69).
- [143]D. Wied i R. Weißbach. „Consistency of the kernel density estimator: A survey“. *Statistical Papers* 53 (2012), s. 1–21 (cyt. na s. 46).
- [144]Z. Xiaoning. „Analysis of military application of UAV swarm technology“. W: *2020 3rd International Conference on Unmanned Systems (ICUS)*. 2020, s. 1200–1204 (cyt. na s. 1).
- [145]C. Xu, H. Jia i Z. Chen. „Simultaneous robust control and sensor fault detection for a ducted coaxial-rotor UAV“. *IEEE Access* 7 (2019), s. 167739–167753 (cyt. na s. 123).

- [146]Z. Yang, X. Yu, S. Dedman i in. „UAV remote sensing applications in marine monitoring: Knowledge visualization and review“. *Science of the Total Environment* 838 (2022), s. 1–23 (cyt. na s. 1).
- [147]M. Ye, J. Shen, X. Zhang, P. C. Yuen i S.-F. Chang. „Augmentation invariant and instance spreading feature for softmax embedding“. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.2 (2022), s. 924–939 (cyt. na s. 59).
- [148]S. Yoon, S. Kim, J. Bae, Y. Kim i E. Kim. „Experimental evaluation of fault diagnosis in a skew-configured UAV sensor system“. *Control Engineering Practice* 19.2 (2011), s. 158–173 (cyt. na s. 103).
- [149]A. Zeiler, R. Faltermeier, I. R. Keck i in. „Empirical mode decomposition-an introduction“. W: *The 2010 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2010, s. 1–8 (cyt. na s. 104).
- [150]H. Zhang, Q. Gao i F. Pan. „An online fault diagnosis method for actuators of quadrotor UAV with novel configuration based on IMM“. W: *2020 Chinese Automation Congress (CAC)*. 2020, s. 618–623 (cyt. na s. 119).
- [151]X. Zhang, H. Luo, K. Li i O. Kaynak. „Time-domain frequency estimation with application to fault diagnosis of the unmanned aerial vehicles' blade damage“. *IEEE Transactions on Industrial Electronics* 69.5 (2022), s. 5257–5266 (cyt. na s. 123).
- [152]Z. Zhang i M. R. Sabuncu. *Generalized cross entropy loss for training deep neural networks with noisy labels*. 2018. arXiv: 1805.07836 (cyt. na s. 58).
- [153]Z. Zhang, X. Zhang, T. Yan, S. Gao i Z. Yu. „Data-driven fault detection of AUV rudder system: A mixture model approach“. *Machines* 11.5 (2023), s. 1–25 (cyt. na s. 103).
- [154]S. Zhao, Q. Wang, X. Fang i in. „Application and development of autonomous robots in concrete construction: Challenges and opportunities“. *Drones* 6.12: 424 (2022), s. 1–26 (cyt. na s. 1).
- [155]Y. Zhong, Y. Zhang, W. Zhang, J. Zuo i H. Zhan. „Robust actuator fault detection and diagnosis for a quadrotor UAV with external disturbances“. *IEEE Access* 6 (2018), s. 48169–48180 (cyt. na s. 123).
- [156]L. Zuo, L. Yao i Y. Kang. „UIO based sensor fault diagnosis and compensation for quadrotor UAV“. W: *2020 Chinese Control And Decision Conference (CCDC)*. IEEE. 2020, s. 4052–4057 (cyt. na s. 123).

