

Raport z prac wykonanych w ramach konkursu „POLEĆ Z NAMI DO DUBAJU”

Nazwa Koła Naukowego: SKN Zastosowania Metod Sztucznej Inteligencji AI-METH

Uczelnia: Politechnika Śląska

Osoba zgłaszająca raport: Dawid Szolc

Skład osobowy zespołu: Dawid Szolc, Filip Stasiuk, Jacek Stanisławski

Opiekun zespołu: dr hab. Inż. Piotr Przystałka, prof. PŚ

Data wykonania raportu: 31.10.2022

Spis Treści:

1.	Część wstępna	3
1.1.	Instalacja i przygotowanie środowiska NAZCA 4.0 do zbierania oraz eksploracji danych	3
1.2.	Przedstawienie projektu badawczego symulującego proces przemysłowy	4
1.3.	Podłączenie badanego układu do środowiska NAZCA 4.0.....	22
1.4.	Zebranie danych pomiarowych.....	26
2.	Część związana z agregacją danych w NAZCA 4.0.....	28
2.1.	Agregacja danych z różnych czujników z własnego układu pomiarowego.....	28
2.2.	Synchronizacja danych	30
2.3.	Wyciągnięcie cyklu z sygnału	34
2.4.	Wyznaczenie trendu i sezonowości	36
2.5.	Określenie współczynników.....	39
2.6.	Wyliczenie wskaźników za pomocą narzędzia JUPYTER	41
3.	Część związana z uczeniem maszynowym – analiza eksploracyjna danych (EDA) wraz z analizą statystyczną.....	42
3.1.	Wyznaczenie zależności pomiędzy zmiennymi	42
3.2.	Selekcja i ekstrakcja cech sygnalowych na podstawie, których można przeprowadzić wnioskowanie	43
3.3.	Analiza statystyczna zmiennych.....	51
3.4	Dobór algorytmu do klasy problemu.....	56
3.5	Przygotowanie danych do procesu uczenia i weryfikacji algorytmu.....	58
3.6	Analiza uzyskanych rezultatów.....	58
4.	Wnioski.....	66
5.	Bibliografia.....	66

1. Część wstępna

1.1. Instalacja i przygotowanie środowiska NAZCA 4.0 do zbierania oraz eksploracji danych

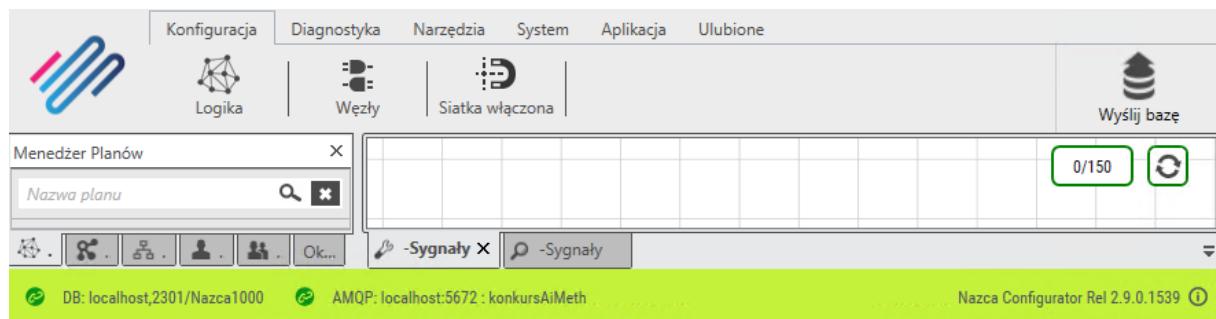
Stosując się do instrukcji zapewnionych przez ekspertów z firmy APA z powodzeniem zainstalowano środowisko NAZCA 4.0, w którego skład wchodzi m.in:

- rdzeń systemu (core), służący do pobierania danych ze sterownika PLC i wysyłania ich do bazy systemu Kafka.

Serwis		Licencja produktu	Statystyka logiki	Konsola	
Statusy		Okno logów	Sesje klientów	Stan węzłów	Uruchamianie
Usluga	Status	Informacje			Czas
CALDAV	NotStarted	Disabled			0:00:00
DATABASE	Succeeded				0:00:00.0068682
GLOBAL ARCHIVES	Succeeded				0:00:00
INSTANCE	Succeeded	Running as 'Single master'			0:00:00.0064052
KAFKA - Serwer Kafka - 9	Succeeded	Serwer Kafka is up			0:00:00.1342969
LICENCES	Succeeded				0:00:00.0217014
LOGIC	Succeeded				0:00:00.0364803
MESSAGING - BASIC	Succeeded				0:00:00.1057314
MESSAGING - BIG DATA	Succeeded				0:00:00.0996709
MONGO - Serwer MongoDB	Succeeded	Serwer MongoDB is up			0:00:02.4432359
MONGO - testy	Succeeded	testy is up			0:00:03.0020472
NLOG	Succeeded	Logging into: \			0:00:00.141323

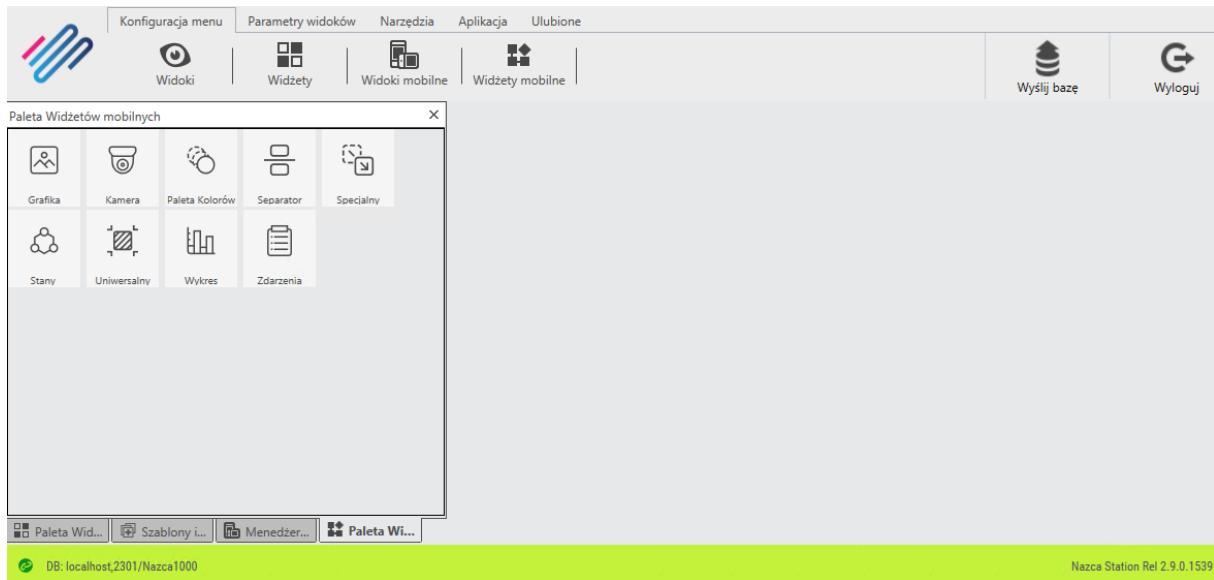
Rys. 1.1: Zrzut ekranu z działającego modułu Core

- konfigurator, służący do określenia, które sygnały oraz w jaki sposób przetwarzane rdzenie systemu (core) wysyłać ma do bazy



Rys. 1.2: Zrzut ekranu informujący o dobrym połączeniu konfiguradora z PLC oraz Core

- stacja operatorska, służąca do zarządzania automatyką budynkową



Rys. 1.3: Zrzut ekranu informujący o dobrym połączeniu stacji operatorskiej

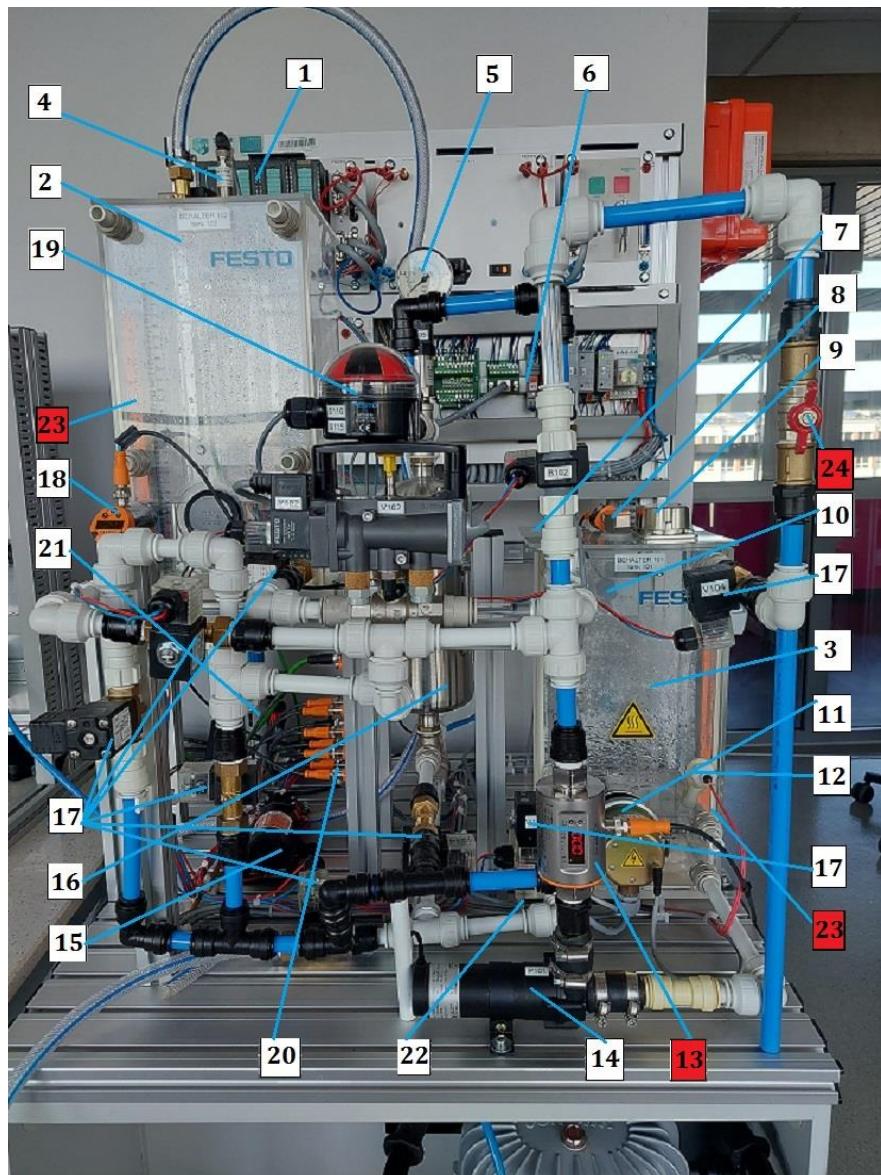
Dzięki poprawnej instalacji oraz ustawieniu środowiska, umożliwiłyśmy zbieranie danych z interesujących nas czujników w czasie rzeczywistym do bazy danych i odtworzenie ich w dowolnej chwili czasu.

1.2. Przedstawienie projektu badawczego symulującego proces przemysłowy

Celem projektu jest wykrycie oraz klasyfikacja sytuacji awaryjnych stanowiska laboratoryjnego. Ogólne założenie projektu są następujące. Zespół badawczy będzie utworzy algorytm bazujący na metodach sztucznej inteligencji, który przeznaczony będzie do wykrywania oraz klasyfikacji symulowanych uszkodzeń stanowiska laboratoryjnego zbudowanego wcześniej przez członków projektu ICS CyberSecurity koła AI-METH.

Opis stanowiska

Stanowisko laboratoryjne, na którym realizowany jest projekt to układ hydrotroniczny wykonujący przemysłowy proces regulacji parametrów cieczy (m.in. temperatury cieczy, przepływu cieczy, poziomu cieczy w zbiorniku oraz ciśnienia cieczy). Wszystkie procesy są monitorowane dzięki działaniu systemu pomiarowego, którego najważniejsze elementy zakupiliśmy m.in. dzięki wsparciu finansowemu zapewnionemu przez firmę APA Group. Poniżej zamieszczono opisane zdjęcie aktualnego stanowiska oraz jego poszczególnych elementów (elementy zaznaczone na czerwono zostały zakupione dzięki dofinansowaniu APA Group).



Rys. 1.4: Zdjęcie z frontu stanowiska laboratoryjnego z opisem widocznych elementów

Spis elementów stanowiska

1. Siemens S7-300 – sterownik PLC model CPU314C-2 PN/DP.
2. Zbiornik cieczy tank102 – zbiornik górny o oznaczeniu B102.
3. Zbiornik cieczy tank101 – zbiornik dolny o oznaczeniu B101.
4. Czujnik ultradźwiękowy B101 – odpowiada za pomiar poziomu w zbiorniku B102.
5. Piezorezystywny czujnik ciśnienia– odpowiada za pomiar ciśnienia w zbiorniku B103.
6. Elementy układu sterowania.
7. Przełącznik przelewowy S111 - zamontowany w zbiorniku B101, odpowiada za zabezpieczenie przed przekroczeniem maksymalnego poziomu cieczy w zbiorniku.
8. IFM UGT580 - czujnik ultradźwiękowy, umożliwia pomiar poziomu cieczy w zbiorniku B101.
9. IFM JN2200 – Czujnik nachylenia cieczy w zbiorniku B101.
10. Czujniki pojemnościowe B113 i B114 – dwie sztuki tego typu czujnika zostały zamontowane w zbiorniku B101. Są odpowiedzialne za pomiar dolnego i górnego poziomu cieczy.
11. Grzałka E104 – zamontowana w zbiorniku B101.
12. Pływak dający sygnał binarny w zbiorniku B101.
13. Przepływomierz elektromagnetyczny SM6400
14. Pompa wirowa.
15. Pompa wirowa zbiornika wyrównawczego.
16. Zbiornik wyrównawczy ciśnienia B103.
17. Zawory sterowane elektrycznie.
18. IFM TN2435 – czujnik temperatury Pt1000 z wyświetlaczem. Wspiera standard komunikacji IO-Link.
19. Zespół zaworu kulowego V102 oraz przełączniki pływakowe S112 i S117 – generujące sygnał binarny.
20. IFM AL1302 – jest to urządzenie typu master, odpowiedzialne za obsługę systemu czujników działających w standardzie IO-Link.
21. Zawór regulacji ciśnienia w układzie pneumatyki.
22. Czujnik temperatury P101 – rezystancyjny czujnik Pt100.
23. Elektroniczny czujniki poziomu cieczy w zbiornikach.
24. Zawór kulowy ręczny.

Opis najważniejszych elementów wchodzących w skład stanowiska laboratoryjnego

- pompa cieczy sterowana regulatorem PID, ma za zadanie wytworzyć zadany przepływ wody w obiegu oraz ciśnienie w zbiorniku B103



Rys. 1.5: Zdjęcie pompy sterowanej regulatorem PID

- dwa zbiorniki wody (B102 oraz B101)



Rys. 1.6: Zdjęcie zbiornika cieczy B101



Rys. 1.7: Zdjęcie zbiornika cieczy B102

- zbiornik ciśnieniowy (B103), ciśnienie utrzymywane jest w zbiorniku dzięki elektrozaworze V103, który po osiągnięciu odpowiedniego ciśnienia można natychmiast zamknąć.



Rys. 1.8: Zdjęcie zbiornika ciśnieniowego 103

- system pomiarowy (przepływomierz), jest to zaawansowany przepływomierz elektromagnetyczny SM6400 firmy Ifm, działający w technologii IO-link, który został zakupiony dzięki dofinansowaniu firmy APA Group.



Rys. 1.9: Zdjęcie zamontowanego przepływomierza

- system pomiarowy (czujniki poziomu cieczy) system wyposażono również w dwa elektroniczne czujniki poziomu cieczy KQ1000 firmy Ifm, działające w technologii IO-link, które również zostały zakupione dzięki dofinansowaniu firmy APA Group



Rys. 1.10: Zdjęcie I-O linkowego czujnika poziomu cieczy zamontowanego na zbiorniku cieczy 102



Rys. 1.11: Zdjęcie I-O linkowego czujnika poziomu cieczy zamontowanego na zbiorniku cieczy 101

- komputer Zotac ZBOX MI643 EDGE, służący jako lokalizacja serwera brzegowego NAZCA. Dzięki temu Core NAZCA zbierający dane może to robić bez przerwy, i pozwala odciążyć główny komputer, na którym sterujemy stanowiskiem. Został on zakupiony dzięki dofinansowaniu firmy APA Group.

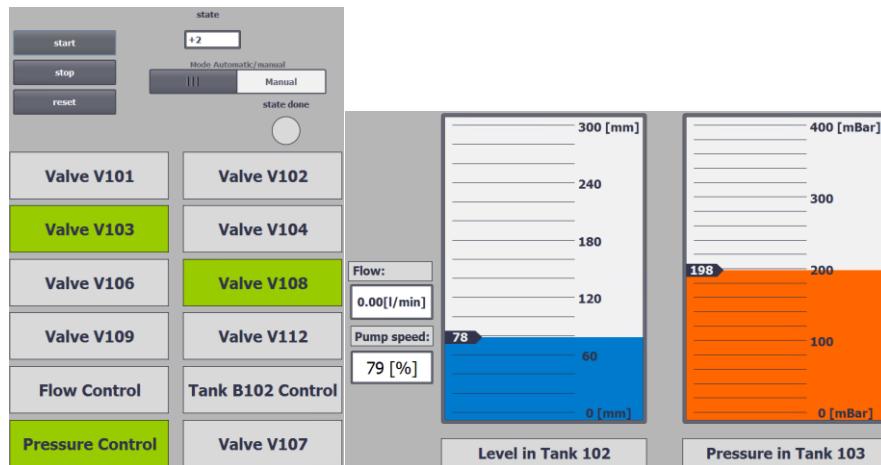


Rys. 1.12: Zdjęcie komputera Zotax ZBOX zbierającego dane z stanowiska

Zasada działania stanowiska laboratoryjnego

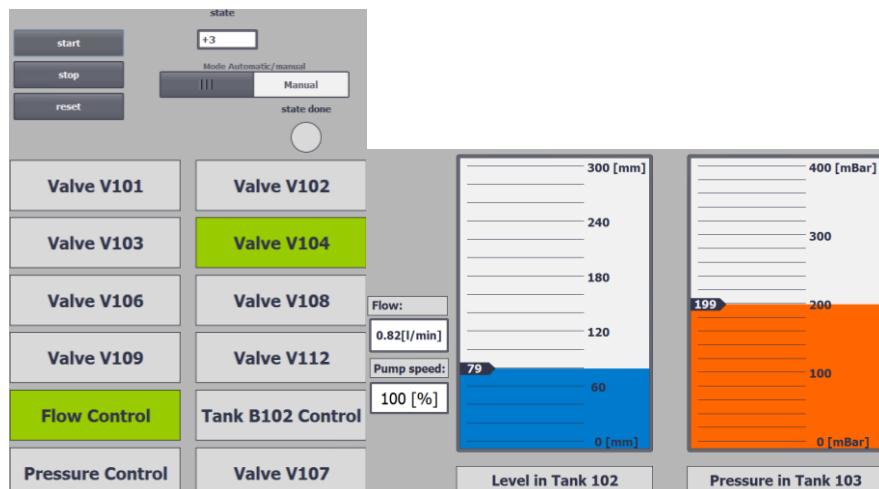
Badany układ ma za zadanie przejść przez 8 stanów funkcjonalnych (składających się w jeden cykl) mających na celu zmiennie wpływać na zachowanie się przedmiotowej cieczy. W każdym ze stanów funkcjonalnych stanowisko osiąga zadane parametry, które różnią się w zależności od sezonu pracy. Poniżej wytłumaczono zasadę działania każdego ze stanów funkcjonalnych układu oraz pokazano wizualizację ich parametrów w systemie SCADA.

- Stan funkcjonalny 1 to stan w którym osiągane są warunki początkowe pracy układu.
- Stan funkcjonalny 2 ma za zadanie kontrolować ciśnienie obrabianej cieczy w zbiorniku B103 na zadanym poziomie (w zależności od sezonu)



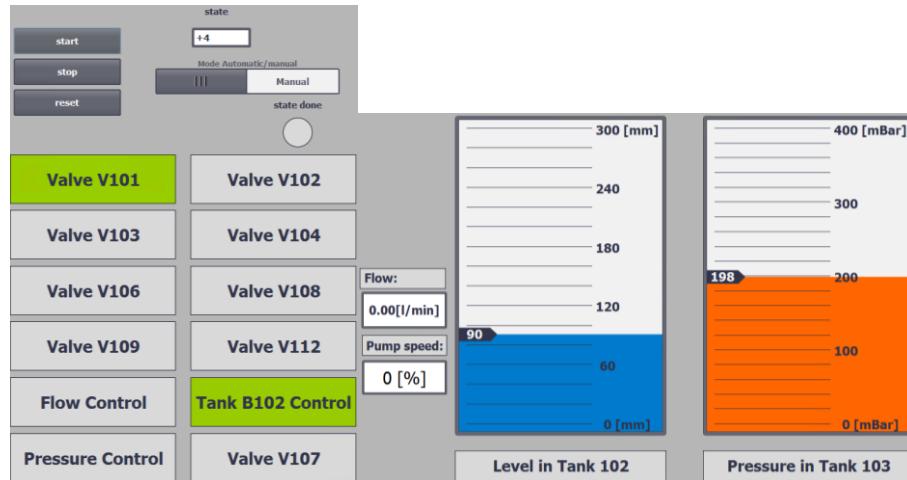
Rys. 1.13: Wizualizacja pracy stanowiska w stanie 2

- Stan funkcjonalny 3 ma za zadanie kontrolować przepływ cieczy na zadanym poziomie (w zależności od sezonu)



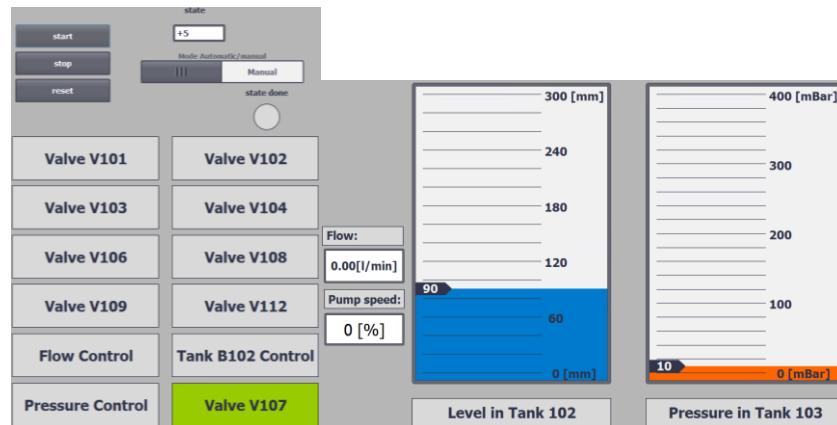
Rys. 1.14: Wizualizacja pracy stanowiska w stanie 3

- Stan funkcjonalny 4 ma za zadanie utrzymać wysokość słupa obrabianej cieczy w zbiorniku B102 na poziomie 90mm



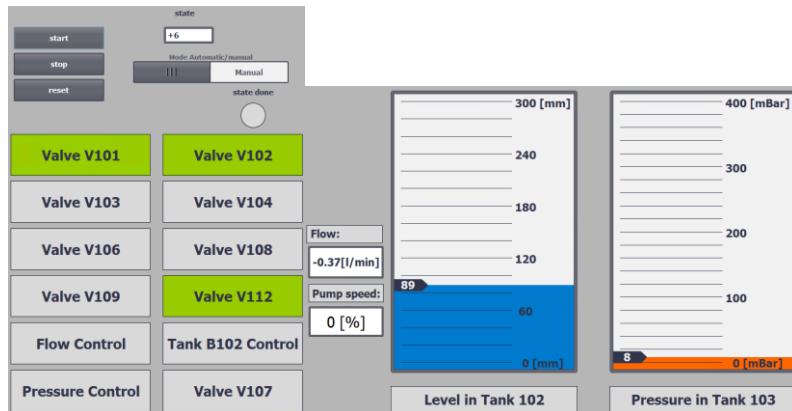
Rys. 1.15: Wizualizacja pracy stanowiska w stanie 4

- Stan funkcjonalny 5 ma za zadanie spuścić ciśnienie ze zbiornika B103 poprzez otwarcie zaworu pneumatycznego V102



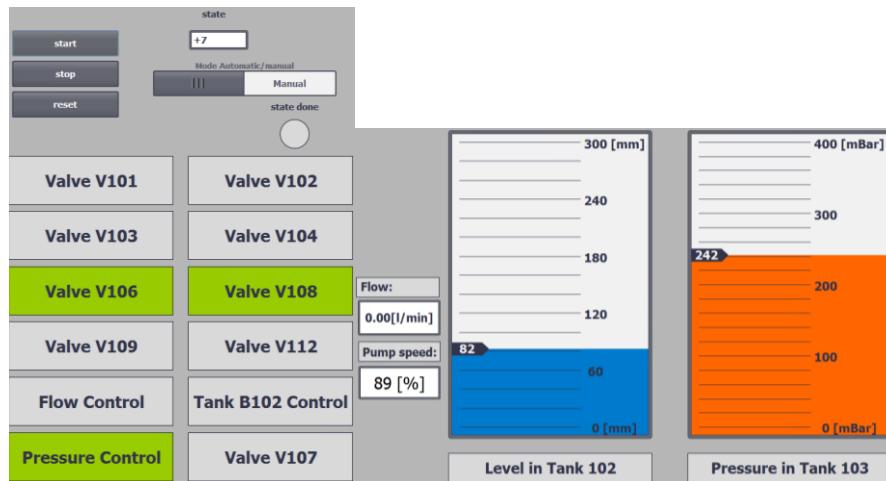
Rys. 1.16: Wizualizacja pracy stanowiska w stanie 5

- Stan funkcjonalny 6 ma za zadanie spuścić nadmiar cieczy ze zbiornika B102



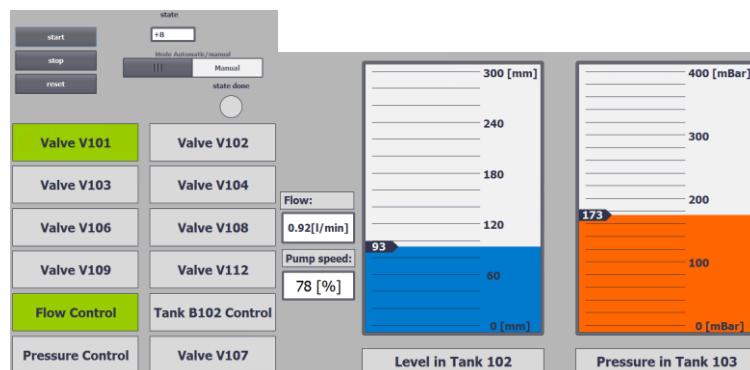
Rys. 1.17: Wizualizacja pracy stanowiska w stanie 6

- Stan funkcjonalny 7 ma za zadanie kontrolować ciśnienie obrabianej cieczy w zbiorniku B103 na zadanym poziomie wyższym niż w stanie 2 (w zależności od sezonu)



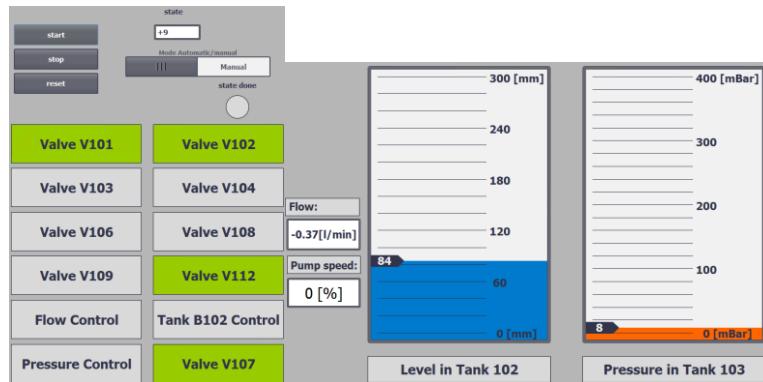
Rys. 1.18: Wizualizacja pracy stanowiska w stanie 7

- Stan funkcjonalny 8 ma za zadanie dopompować obrabianą ciecz do zbiornika B102 oraz zbiornika B103 przy utrzymaniu zadanego przepływu.



Rys. 1.19: Wizualizacja pracy stanowiska w stanie 8

- Stan funkcjonalny 9 ma za zadanie spuścić wodę ze zbiornika B102 do wysokości 80mm.

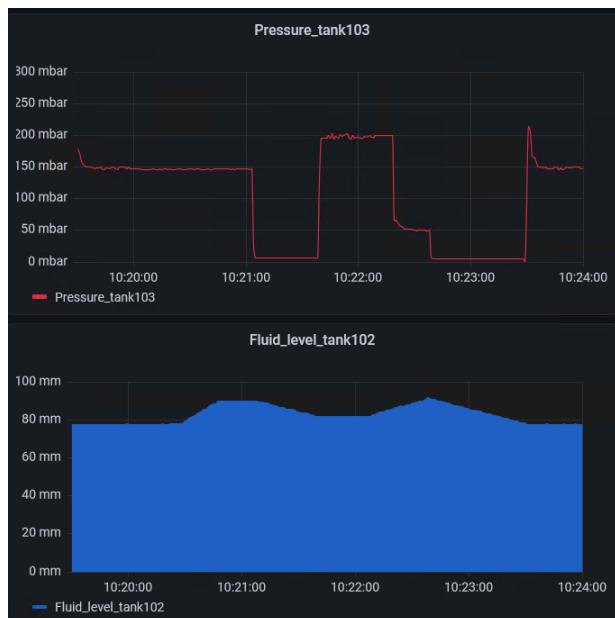


Rys. 1.20: Wizualizacja pracy stanowiska w stanie 9

Symulacja stanów technicznych

W projekcie symulowanych jest pięć stanów technicznych w tym cztery z uszkodzeniami, które algorytm musi wykryć i rozróżnić. Każde z nich powoduje niechciane zmiany w działaniu systemu, co można najlepiej zaobserwować po wykresach zagregowanych sygnałów: ciśnienia w zbiorniku B103, poziomu cieczy w zbiorniku B102, przepływu cieczy w układzie oraz zadanej prędkości pompy z jednego cyklu pracy stanowiska w danym uszkodzeniu. Poniżej przedstawiony jest wykaz stanów technicznych bez i z uszkodzeniem z wytlumaczeniem ich działania.

- Stan techniczny F0 - stan pełnej zdatności, stanowisko działa poprawnie.



Rys. 1.21: Wykresy przedstawiające sygnały ciśnienia w zbiorniku B103 oraz poziomu cieczy w zbiorniku B102 zebrane ze stanu pełnej zdatności w czasie jednego cyklu pracy układu



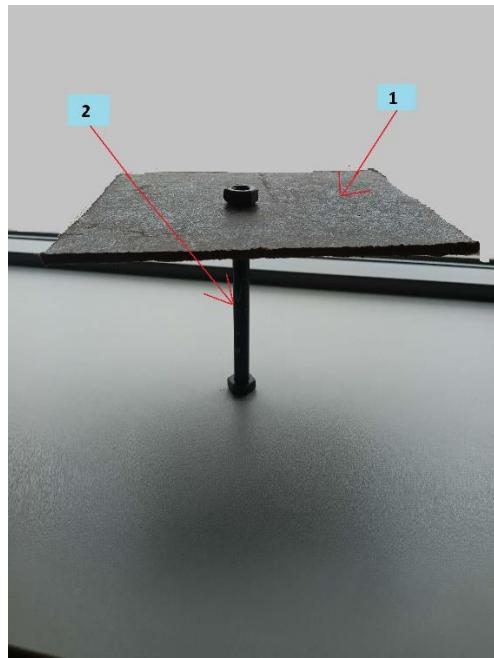
Rys. 1.22: Wykresy przedstawiające sygnały przepływu cieczy oraz zadanej prędkości pompy zebrane ze stanu pełnej zdatności w czasie jednego cyklu pracy układu

- Stan techniczny z uszkodzeniem F1 - pierwsze uszkodzenie, polega na zasymulowaniu zbierania się kożucha w zbiorniku cieczy B102 co zaburza odczyt z czujnika wysokości słupa cieczy.

Do symulacji tego uszczodzenia utworzono sztuczny kożuch, zrobiony z deski oraz śruby, która podpiera ją na odpowiedniej wysokości. Można ją ulokować w zbiorniku B102, co pozwala na oszukanie czujnika, który pokazuje wtedy wartość równą około 123mm, kiedy w rzeczywistości poziom cieczy jest mniejszy.



Rys. 1.23: Zdjęcie deski wykorzystywanej do symulacji uszkodzenia F1 zamontowanej w zbiorniku



Rys. 1.24: Widok deski wykorzystywanej do symulacji uszkodzenia F1, z widoczną śrubą podpierającą (1 - deska oszukująca czujnik, 2 - śruba podpierająca deskę)



Rys. 1.25: Wykresy przedstawiające sygnały ciśnienia w zbiorniku B103 oraz poziomu cieczy w zbiorniku B102 zebrane z pracy stanowiska podczas symulacji uszkodzenia F1 w czasie jednego cyklu pracy układu



Rys. 1.26: Wykresy przedstawiające sygnały przepływu cieczy oraz zadanej prędkości pompy z pracy stanowiska podczas symulacji uszkodzenia F1 w czasie jednego cyklu pracy układu

- Stan techniczny z uszkodzeniem F2 - drugie uszkodzenie polega na symulacji wycieku cieczy ze zbiornika B102, do jego symulacji otwierane są zawory podczas pracy stanowiska, co powoduje ubytek cieczy w trakcie każdego stanu.



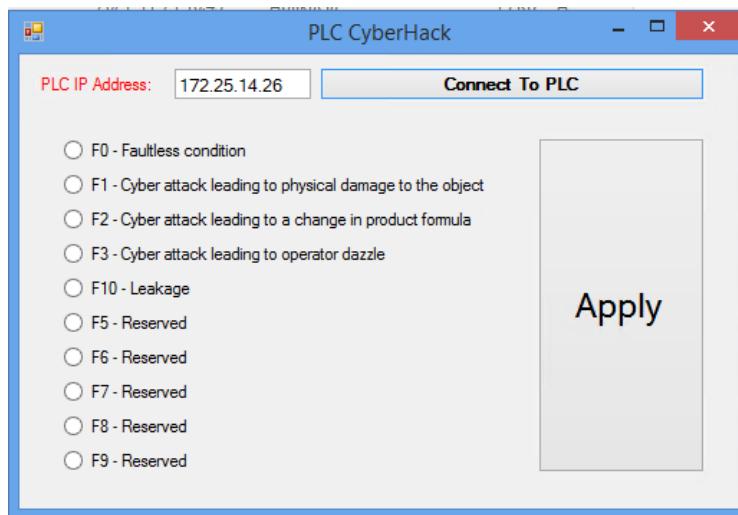
Rys. 1.27: Wykresy przedstawiające sygnały ciśnienia w zbiorniku B103 oraz poziomu cieczy w zbiorniku B102 zebrane z pracy stanowiska podczas symulacji uszkodzenia F2 w czasie jednego cyklu pracy układu



Rys. 1.28: Wykresy przedstawiające sygnały przepływu cieczy oraz zadanej prędkości pompy z pracy stanowiska podczas symulacji uszkodzenia F2 w czasie jednego cyklu pracy układu

- Stan techniczny z uszkodzeniem F3 - trzecie uszkodzenie symulowane jest poprzez wykonanie symulowanego cyber ataku na stanowisko, którego efektem jest niestabilna praca pompy.

Do jego symulacji wykorzystano aplikację „PLC Cyber Hack” utworzoną przez członka naszego koła w poprzednim projekcie dotyczących zagadnień cyberbezpieczeństwa.



Rys. 1.29: Zrzut ekranu przedstawiający aplikację „PLC Cyber Hack”



Rys. 1.30: Wykresy przedstawiające sygnały ciśnienia w zbiorniku B103 oraz poziomu cieczy w zbiorniku B102 zebrane z pracy stanowiska podczas symulacji uszkodzenia F3 w czasie jednego cyklu pracy układu



Rys. 1.31: Wykresy przedstawiające sygnały przepływu cieczy oraz zadanej prędkości pompy z pracy stanowiska podczas symulacji uszkodzenia F3 w czasie jednego cyklu pracy układu

- Stan techniczny z uszkodzeniem F4 - czwarte uszkodzenie polega na symulowaniu zbierania się osadu w układzie i przytykaniu rury, przez którą płynie ciecz podczas regulacji przepływu. Pompa cieczy musi pracować w tym przypadku mocniej aby osiągnąć zadany przepływ lub nie ma odpowiedniej mocy aby osiągnąć zadany przepływ i jest on mniejszy niż powinien.

Do symulacji tego uszkodzenia zamontowano na stanowisku ręczny zawór kulowy, dzięki któremu jest możliwość ograniczenia przepływu cieczy w odpowiednim zakresie.



Rys. 1.32: Zdjęcie zaworu wykorzystywanego do symulacji uszkodzenia F4



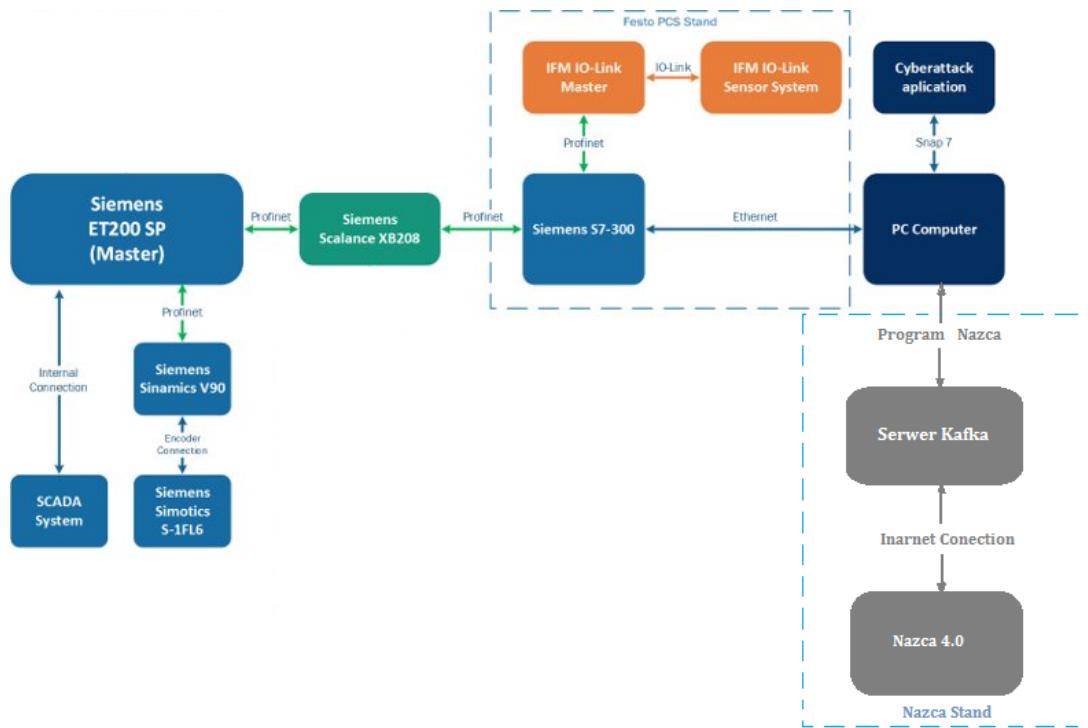
Rys. 1.33: Wykresy przedstawiające sygnały ciśnienia w zbiorniku B103 oraz poziomu cieczy w zbiorniku B102 zebrane z pracy stanowiska podczas symulacji uszkodzenia F4 w czasie jednego cyklu pracy układu



Rys. 1.34: Wykresy przedstawiające sygnały przepływu cieczy oraz zadanej prędkości pompy z pracy stanowiska podczas symulacji uszkodzenia F4 w czasie jednego cyklu pracy układu

1.3. Podłączenie badanego układu do środowiska NAZCA 4.0

Przed przystąpieniem do projektu zmienne procesowe gromadzone były w sterowniku Siemens S7-300, który wykonuje główny program sterujący całym układem. Do wizualizacji wykorzystywany jest system SCADA zaimplementowany w module ET-200SP połączony przez sieć profinet ze sterownikiem S7-300.



Rys. 1.35: Schemat blokowy stanowiska zmodernizowanego o funkcjonalność współpracy z serwerem Kafka oraz Nazca 4.0

Należało więc połączyć Sterownik S7-300 z konfiguratorem NAZCA, który zainstalowany był na tym samym komputerze, do którego podłączony jest sterownik:

- Do komunikacji PLC z rdzeniem NAZCA utworzono specjalny blok danych przechowujący wszystkie potrzebne sygnały ze sterownika.

The screenshot shows the 'Program blocks' tree on the left and a detailed table on the right.

Program blocks:

- Add new block
- CYC_INT5 [OB35]
- I/O_FLT1 [OB82]
- I/O_FLT2 [OB83]
- Main [OB1]
- OBNL_FLT [OB85]
- RACK_FLT [OB86]
- Manual_Mode_Level [FC5]
- Map [FC9]
- Move_bool_data [FC7]
- PID_Pump_Mode [FC4]
- Pump_Mode [FC1]
- RewriteInputValuesToOutput [FC6]
- Scale_Normalize [FC120]
- SetValues_Normalize [FC121]
- TankLids [FC8]
- Temperature Control [FC2]
- Valves [FC3]
- Automatic_Control [FB5]
- IO-Link_Measurements [FB4]
- Own_TONR [FB9]
- Tank101_Level_Auto [FB3]
- Tank101_Temperature_Auto [FB1]
- Tank101_Temperature_Manual [FB2]
- Automatic_Control_DB [DB10]
- Bool_Input_Output_Values [DB12]
- ConfigAutomaticControl [DB17]
- Functionality Conditions [DB14]
- HMI_automatic_control [DB16]
- IO-Link_Measurements_DB [DB11]
- Measurements [DB1]
- Regulator_PID [DB3]
- Scada_System [DB2]
- Sygnaly_NAZCA [DB18] (highlighted)
- Tank101_Level_Auto_DB [DB6]
- Tank101_Temperature_Auto_DB [DB4]
- Tank101_Temperature_Manual_DB [DB5]
- Valve_Control [DB13]
- Valves_Control2 [DB8]

Sygnaly_NAZCA

	Name	Data type
Static	V101	Bool
	V102	Int
	V103	Bool
	V104	Bool
	V106	Bool
	V107	Bool
	V108	Bool
	V109	Bool
	V112	Bool
	State	Int
	Fluid_level_tank102	Real
	Pressure_tank103	Real
	Fluid_lvl_MIN_tank101	Bool
	Fluid_lvl_MAX_tank101	Bool
	Flowmeter	Real
	Pump_speed	Real
	EStop	Bool
	Failure_F1	Bool
	Failure_F2	Bool
	Failure_F3	Bool
	Failure_F4	Bool
	Failure_F5	Bool
	Work	Bool
	tank102_set	Real
	Pressure_set	Real
	Flow_set	Real
	Fluid_lvl_tank102_IO...	Int
	Fluid_lvl_tank101_IO...	Int
	V102_bool	Bool
	cycle_counter	Int

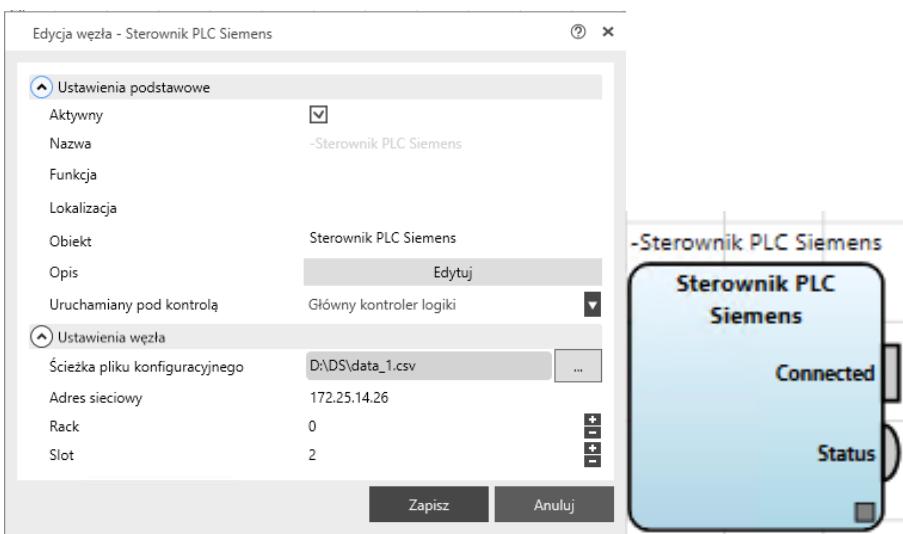
Rys. 1.36: Zrzut ekranu z wszystkimi funkcjami oraz blokami danych sterownika PLC z podświetlonym blokiem agregującym sygnały zbierane do Konfiguratora NAZCA, oraz zmiennymi zawartymi w tym bloku

- Następnie utworzono plik konfiguracyjny CSV, który wykorzystano do ustawienia bloku pobierającego dane w Konfiguratorze. Zawiera on informacje takie jak:
 - Name – nazwa zmiennej
 - Type – typ zmiennej
 - Address – adres zmiennej w bloku danych z PLC
 - DB[optional] – numer bloku, w którym zawarte są zmienne
 - Time[s] – określenie co jaki odstęp czasowy konfigurator ma pobierać dane
 - Scale – mnożnik zmiennej (domyślnie 1)
 - Read/Write – informacja dotycząca czy zmienność jest wartością odczytywaną z PLC czy do niego wysyłaną.

Tab. 1.1: Plik konfiguracyjny z ustawieniami zmiennych

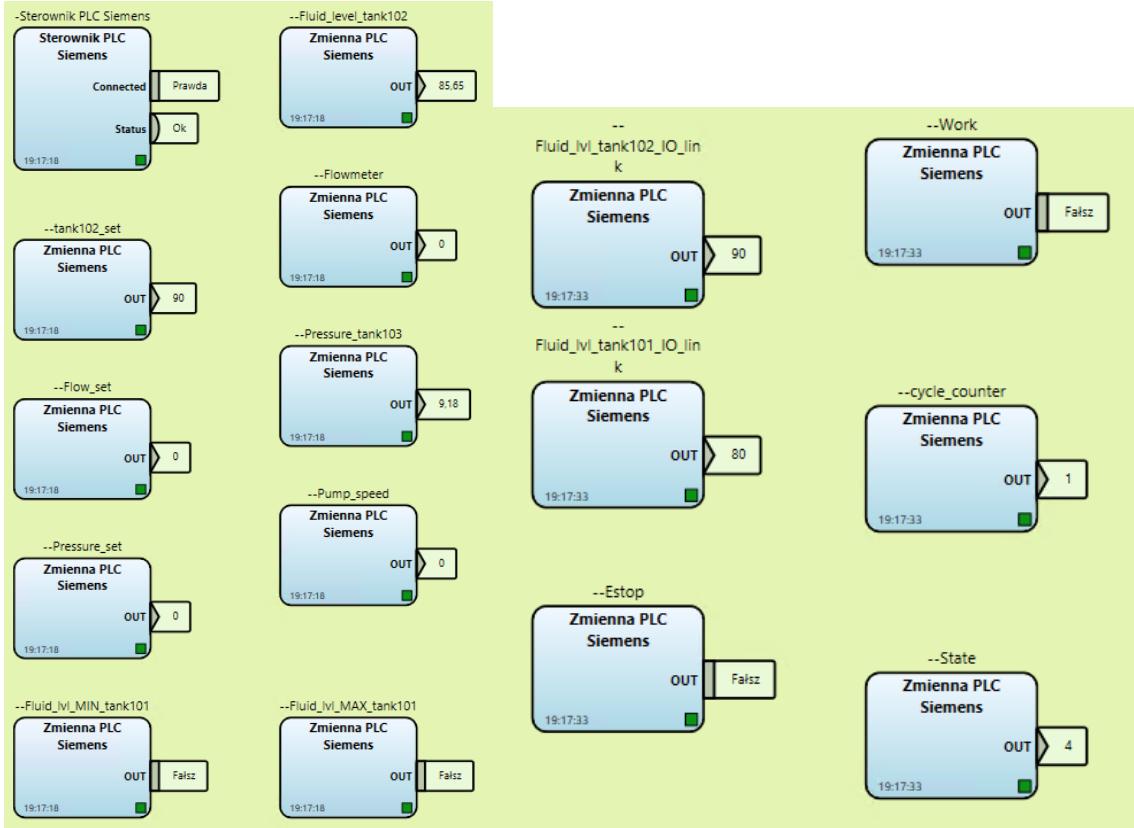
Name	Type	Address	Db [optional]	Time [s]	Scale	Read/Write
V101	Bool	DBD0.0		18 0.1		1 R
V102	Int	DBD2.0		18 0.1		1 R
V103	Bool	DBD4.0		18 0.1		1 R
V104	Bool	DBD4.1		18 0.1		1 R
V106	Bool	DBD4.2		18 0.1		1 R
V107	Bool	DBD4.3		18 0.1		1 R
V108	Bool	DBD4.4		18 0.1		1 R
V109	Bool	DBD4.5		18 0.1		1 R
V112	Bool	DBD4.6		18 0.1		1 R
State	Int	DBD6.0		18 0.1		1 R
Fluid_level_tank102	Real	DBD8.0		18 0.1		1 R
Pressure_tank103	Real	DBD12.0		18 0.1		1 R
Fluid_lvl_MIN_tank101	Bool	DBD16.0		18 0.1		1 R
Fluid_lvl_MAX_tank101	Bool	DBD16.1		18 0.1		1 R
Flowmeter	Real	DBD18.0		18 0.1		1 R
Pump_speed	Real	DBD22.0		18 0.1		1 R
Estop	Bool	DBD26.0		18 0.1		1 R
Failure_F1	Bool	DBD26.1		18 0.1		1 R
Failure_F2	Bool	DBD26.2		18 0.1		1 R
Failure_F3	Bool	DBD26.3		18 0.1		1 R
Failure_F4	Bool	DBD26.4		18 0.1		1 R
Failure_F5	Bool	DBD26.5		18 0.1		1 R
Work	Bool	DBD26.6		18 0.1		1 R
tank102_set	Real	DBD28.0		18 0.1		1 R
Pressure_set	Real	DBD32.0		18 0.1		1 R
Flow_set	Real	DBD36.0		18 0.1		1 R
Fluid_lvl_tank102_IO_link	Int	DBD40.0		18 0.1		1 R
Fluid_lvl_tank101_IO_link	Int	DBD42.0		18 0.1		1 R
V102_bool	Bool	DBD44.0		18 0.1		1 R
cycle_counter	Int	DBD46.0		18 0.1		1 R

Plik konfiguracyjny wykorzystywany jest przez blok w konfiguratorze, który służy do nawiązania odpowiedniej komunikacji między Core a sterownikiem PLC, jego ustawienie widoczne jest poniżej.

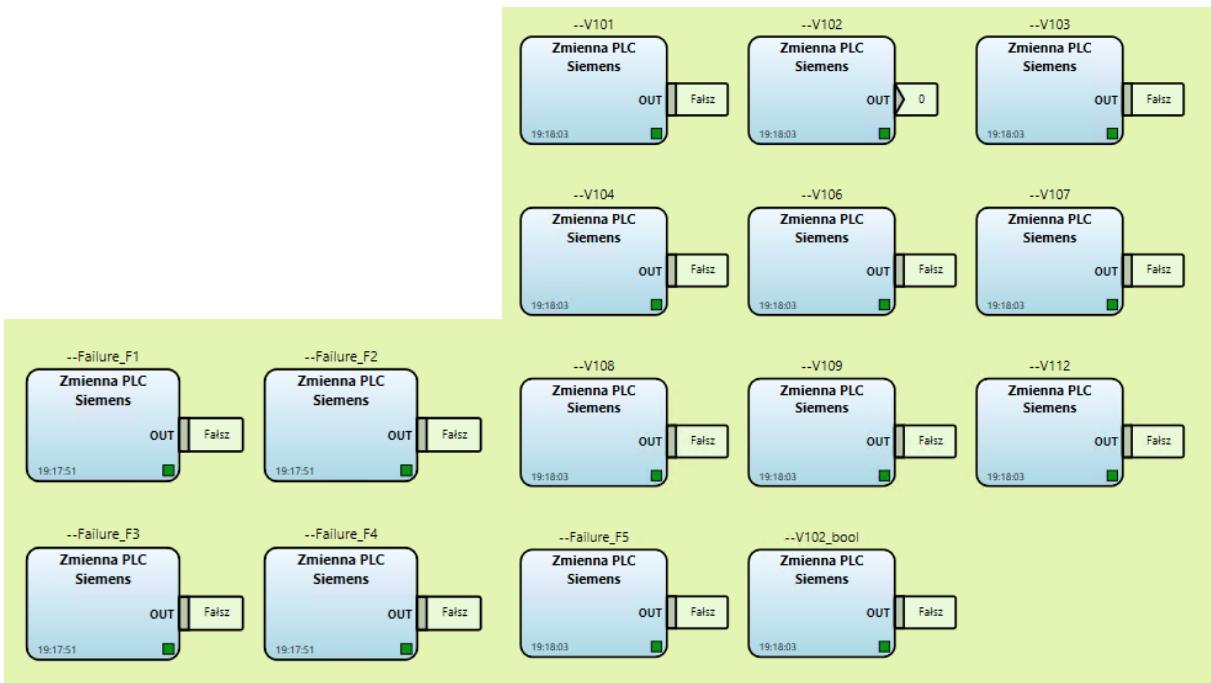


Rys. 1.37: Zrzut ekranu z zawartego w kontrolerze logiki konfiguratora NAZCA bloku połączenia ze sterownikiem PLC, oraz jego ustawień

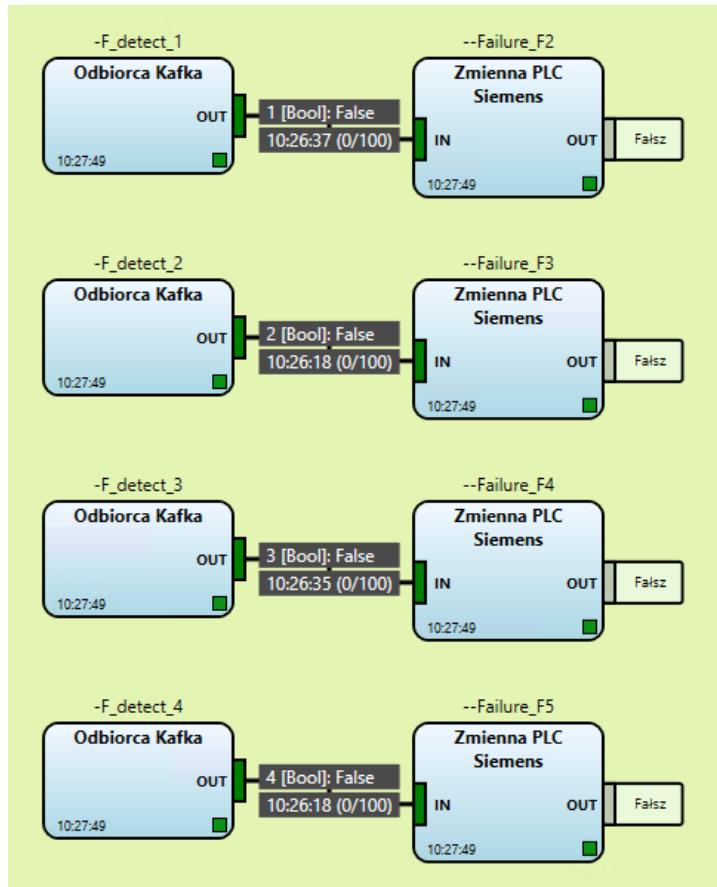
- Po skonfigurowaniu bloku pobierającego dane, dodano wszystkie sygnały do głównej logiki oraz sprawdzono czy połączenie działa prawidłowo i czy sygnały są na bieżąco pobierane.



Rys. 1.38: Zrzut ekranu z trybu diagnostycznego konfiguratora NAZCA, na którym można zaobserwować działanie ważniejszych sygnałów pobranych ze sterownika



Rys. 1.39: Zrzut ekranu z trybu diagnostycznego konfiguratora NAZCA, na którym można zaobserwować otwarcia wszystkich zaworów oraz wykrycia uszkodzeń stanowiska



Rys. 1.40: Zrzut ekranu z trybu diagnostycznego konfiguratora NAZCA, na którym można zaobserwować połączenie wykrytych przez sieć uszkodzeń do zmiennych w sterowniku PLC

Powyżej wykazano, że każdy sygnał poprawnie zwraca odczytywaną wartość ze sterownika PLC.

1.4. Zebranie danych pomiarowych

Wszystkie dane pomiarowe, na podstawie których uczy się algorytm sieci neuronowej, magazynowane są w bazie danych Kafka. Zarchiwizowane dane z okresu, który jest potrzebny do nauczania sieci, pozyskiwane są poprzez program „*Data_download*” [3] napisany w środowisku Jupyter. Wykorzystuje on bibliotekę SDK utworzoną przez ekspertów z firmy APA Group, dzięki której można pobrać dane i zapisać do np. dataframe'a. Po odpowiednim przygotowaniu danych (usunięciu niepotrzebnych sygnałów, dodaniu nowych zależności oraz zunifikowaniu typów zmiennych) dataframe zapisuje jako plik csv, który można wykorzystać do uczenia sieci.

measureTime	Fluid_level_tank102	Pressure_tank103	Pump_speed	Flowmeter	tank102_set	Pressure_set	Flow_set	Fluid_IM_tank101_IO...	Fluid_IM_tank102_IO...	Fluid_IM_MAX_tank101	Fluid_IM_MIN_tank101
2022-09-25T0...	77.91	171.41	78.68	0.12	0	150	0.0	90	80	0	0
2022-09-25T0...	77.91	159.17	73.1	0.0	0	150	0.0	90	80	0	0
2022-09-25T0...	77.91	154.83	72.3	0.0	0	150	0.0	90	80	0	0
2022-09-25T0...	77.91	152.28	71.98	0.0	0	150	0.0	90	80	0	0
2022-09-25T0...	77.91	150.49	71.66	0.0	0	150	0.0	90	80	0	0
2022-09-25T0...	77.91	150.49	71.26	0.0	0	150	0.0	90	80	0	0
2022-09-25T0...	77.91	148.45	71.22	0.0	0	150	0.0	90	80	0	0
2022-09-25T0...	77.91	147.18	71.16	0.0	0	150	0.0	90	80	0	0
2022-09-25T0...	77.91	148.71	71.22	0.0	0	150	0.0	90	80	0	0
2022-09-25T0...	77.91	148.45	70.9	0.0	0	150	0.0	90	80	0	0
2022-09-25T0...	77.91	149.98	71.16	0.0	0	150	0.0	90	80	0	0
2022-09-25T0...	77.91	147.18	71.21	0.0	0	150	0.0	90	80	0	0
2022-09-25T0...	77.91	146.67	70.98	0.0	0	150	0.0	90	80	0	0
2022-09-25T0...	77.91	149.22	71.18	0.0	0	150	0.0	90	80	0	0
2022-09-25T0...	77.91	148.45	71.24	0.0	0	150	0.0	90	80	0	0
2022-09-25T0...	77.91	148.2	71.04	0.0	0	150	0.0	90	80	0	0
2022-09-25T0...	77.91	150.49	71.11	0.0	0	150	0.0	90	80	0	0
2022-09-25T0...	77.91	146.92	71.05	0.0	0	150	0.0	90	80	0	0
2022-09-25T0...	77.91	146.67	71.3	0.0	0	150	0.0	90	80	0	0
2022-09-25T0...	77.91	150.49	70.91	0.0	0	150	0.0	90	80	0	0

Rys. 1.41: Zrzut ekranu z części danych przygotowanych do uczenia sieci - przykładowe 20 próbek danych związanych z zachowaniem cieczy zagregowanych w pliku csv

V101	V102	V103	V104	V106	V107	V108	V109	V112	cycle_counter	EStop	Work	State	Liquid_growth_B102	Pipe_clogging_factor	Flow_diff	Pressure_diff	Pressure_flow_ratio	Pressure_pumpspee...	Tanks_level_diff
0	0	1	0	0	0	1	0	0	1	0	1	2	0.06	0.0	-0.12	-21.41	20.57	13486.89	12.09
0	0	1	0	0	0	1	0	0	1	0	1	2	0.0	0.0	-9.17	0.0	11634.4	12.09	
0	0	1	0	0	0	1	0	0	1	0	1	2	0.0	0.0	-4.83	0.0	11193.48	12.09	
0	0	1	0	0	0	0	1	0	0	1	0	2	0.0	0.0	-2.28	0.0	10960.9	12.09	
0	0	1	0	0	0	0	1	0	0	1	0	2	0.0	0.0	-0.49	0.0	10784.15	12.09	
0	0	1	0	0	0	0	1	0	0	1	0	2	0.0	0.0	-0.49	0.0	10723.86	12.09	
0	0	1	0	0	0	1	0	0	1	0	1	2	0.0	0.0	0.0	1.55	0.0	10572.58	12.09
0	0	1	0	0	0	1	0	0	1	0	1	2	0.0	0.0	0.0	2.82	0.0	10473.49	12.09
0	0	1	0	0	0	1	0	0	1	0	1	2	0.0	0.0	0.0	1.29	0.0	10590.22	12.09
0	0	1	0	0	0	0	1	0	0	1	0	2	0.0	0.0	0.0	1.55	0.0	10525.74	12.09
0	0	1	0	0	0	0	1	0	0	1	0	2	0.0	0.0	0.0	0.02	0.0	10673.2	12.09
0	0	1	0	0	0	0	1	0	0	1	0	2	0.0	0.0	0.0	2.82	0.0	10480.41	12.09
0	0	1	0	0	0	0	1	0	0	1	0	2	0.0	0.0	0.0	3.33	0.0	10409.82	12.09
0	0	1	0	0	0	0	1	0	0	1	0	2	0.0	0.0	0.0	0.78	0.0	10621.37	12.09
0	0	1	0	0	0	0	1	0	0	1	0	2	0.0	0.0	0.0	1.55	0.0	10575.11	12.09
0	0	1	0	0	0	0	1	0	0	1	0	2	0.0	0.0	0.0	1.8	0.0	10528.1	12.09
0	0	1	0	0	0	0	1	0	0	1	0	2	0.0	0.0	0.0	-0.49	0.0	10702.22	12.09
0	0	1	0	0	0	0	1	0	0	1	0	2	0.0	0.0	0.0	3.08	0.0	10439.17	12.09
0	0	1	0	0	0	0	1	0	0	1	0	2	0.0	0.0	0.0	3.33	0.0	10457.25	12.09
0	0	1	0	0	0	0	1	0	0	1	0	2	0.0	0.0	0.0	-0.49	0.0	10671.05	12.09

Rys. 1.42: Zrzut ekranu z części danych przygotowanych do uczenia sieci - przykładowe 20 próbek danych związanych z otwarciami zaworów zagregowanych w pliku csv

2. Część związana z agregacją danych w NAZCA 4.0

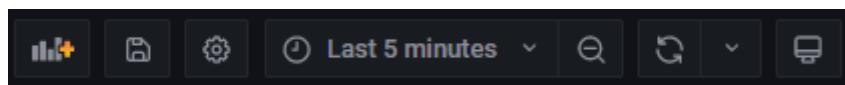
2.1. Agregacja danych z różnych czujników z własnego układu pomiarowego

W efekcie realizacji projektu zebrano dane z 16 różnych czujników (kolor pomarańczowy) oraz 8 ważniejszych sygnałów przetwarzanych w sterowniku PLC (kolor zielony). Poniżej przedstawiono legendę opisującą każdy sygnał.

Tab. 2.1: Opisy gromadzonych sygnałów

Estop	oznacza potwierdzenie awarii stanowiska i przejście do trybu serwisu
Flowmeter	zmierzona wartość przepływu
Fluid_level_tank102	zmierzona wysokość słupa cieczy w zbiorniku górnym 102
Pump_speed	zadana wartość prędkości pompy
Pressure_tank103	zmierzona wartość ciśnienia w zbiorniku 103
Flow_set	zadana wartość przepływu
tank102_set	zadana wysokość słupa cieczy w zbiorniku górnym 102
Pressure_set	zadana wartość ciśnienia
Fluid_lvl_tank101_IO_link	zmierzona wysokość słupa cieczy w zbiorniku górnym 101 przez nowy czujnik IO-link
Fluid_lvl_tank102_IO_link	zmierzona wysokość słupa cieczy w zbiorniku górnym 102 przez nowy czujnik IO-link
Fluid_lvl_MAX_tank101	wykrycie stanu maksymalnego wysokości słupa cieczy w zbiorniku 101
Fluid_lvl_MIN_tank101	wykrycie stanu minimalnego wysokości słupa cieczy w zbiorniku 101
V101	zawór v101 (1 - otwarty)
V102	zawór v102 (1 - otwarty)
V103	zawór v103 (1 - otwarty)
V104	zawór v104 (1 - otwarty)
V106	zawór v106 (1 - otwarty)
V107	zawór v107 (1 - otwarty)
V108	zawór v108 (1 - otwarty)
V109	zawór v109 (1 - otwarty)
V112	zawór v110 (1 - otwarty)
Work	zmienna informująca o działaniu stanu automatycznego
State	zmienna informująca o stanie w którym aktualnie znajduje się stanowisko
cycle counter	aktualny cykl stanowiska

Dzięki połączeniu systemów NAZCA 4.0 oraz Grafana, możliwe było utworzenie wizualizacji przebiegu każdego z sygnałów. Ułatwia to odczyt danych oraz pozwala na lepszą obserwację procesów zachodzących podczas działania układu. Poniżej przedstawiono wykresy przebiegu sygnałów zebranych z bieżącej pracy stanowiska laboratoryjnego.



Rys. 2.1: Ustawienie Grafana na pokazywanie danych sprzed 5 minut

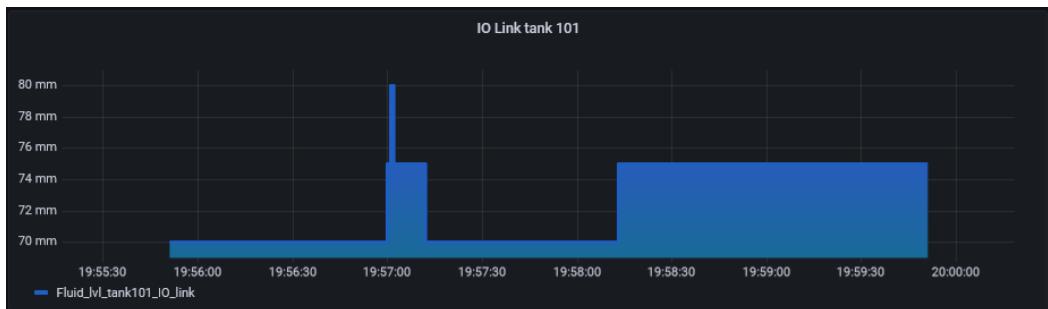
Wizualizacje przykładowych sygnałów związanych z zachowaniem się cieczy w układzie



Rys. 2.2: Wykresy ciśnienia cieczy w zbiorniku B103 oraz poziomu cieczy w zbiorniku B102 sprzed 5 minut



Rys. 2.3: Wykresy przepływu cieczy w układzie oraz zadanej prędkości pomp sprzed 5 minut



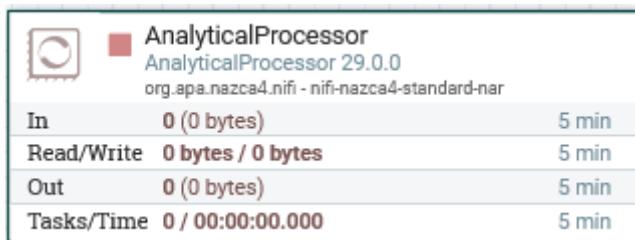
Rys. 2.4: Wykres wysokości słupa cieczy w zbiorniku B101 zebranego z czujnika KQ1000 sprzed 5 minut



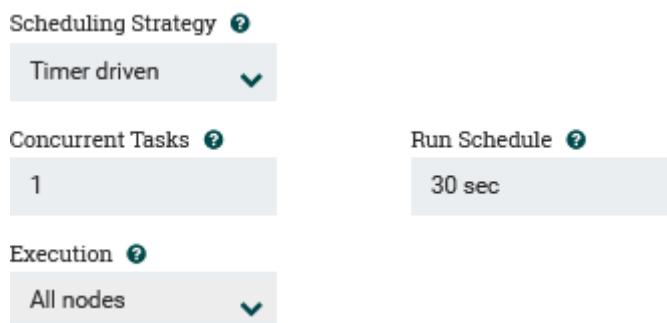
Rys. 2.5: Wykres wysokości słupa cieczy w zbiorniku B102 zebranego z czujnika KQ1000 sprzed 5 minut

2.2. Synchronizacja danych

Dane, które docelowo trafiają na wejście algorytmu sieci neuronowej, zostają pobrane z bazy danych Kafka, poprzez wywołanie funkcji SDK w docelowym programie wykrywającym i klasyfikującym uszkodzenia „*Damage_detection*” [2]. Algorytm pobiera dane z 240 ostatnich sekund, i na ich podstawie klasyfikuje uszkodzenie stanowiska. Program ten wywołany jest co 30 sekund przez blok funkcyjny „*AnalyticalProcessor*” w systemie Nifi.



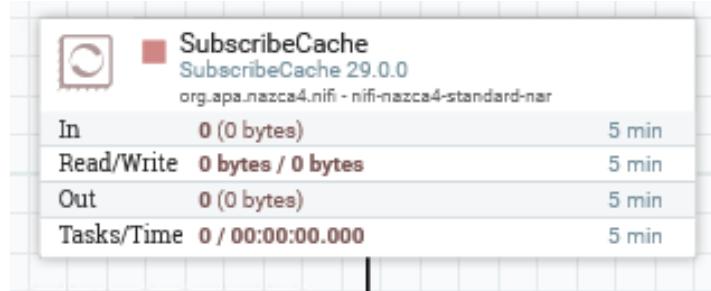
Rys. 2.6: Blok wywołujący algorytm do detekcji i rozróżniania uszkodzeń



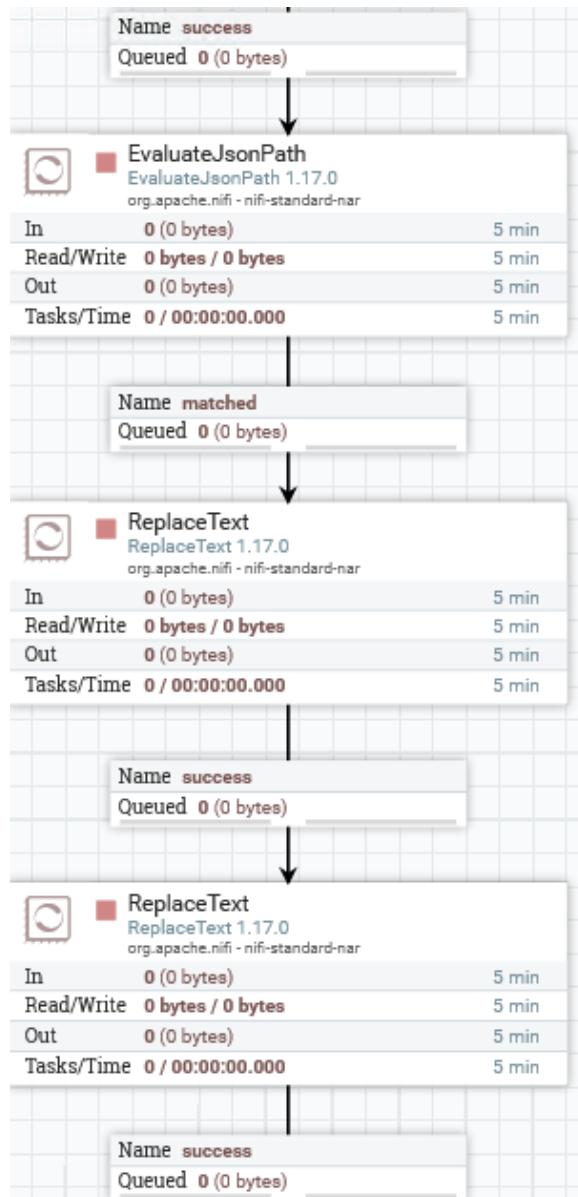
Rys. 2.7: Ustawienia wewnętrzne timera bloku

Aby wysłać informacje o uszkodzeniu do sterownika PLC utworzony został blok o nazwie „Nazca to Edge”. W bloku tym znajdują się procesory o nazwie „SubscribeCache”, których zadaniem jest pobranie informacji o sygnale danego uszkodzenia. Zadaniem kolejnych procesorów jest zmiana wartości True/False na wartość do odczytania przez blok „EdgeBroker”, które składają się z małych

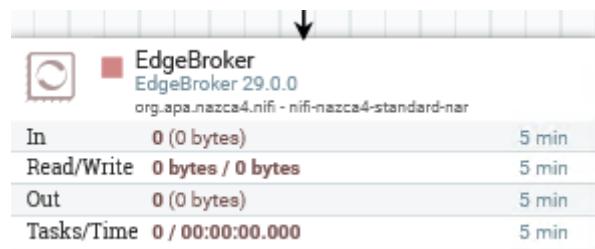
liter. Zadaniem procesora „EdgeBroker” jest wysłanie sygnału o stanie uszkodzenia do konfiguratora NAZCA, który następnie sygnał ten przesyła do sterownika PLC. Wysłanie takiego sygnału sprawia, iż można wyświetlić informacje w SCADA jakie uszkodzenie wystąpiło, oraz zatrzymać cykl programu w celu naprawy uszkodzenia.



Rys. 2.8: Blok pobierający zadaną wartość z bazy danych Kafka



Rys. 2.9: Bloki konwertujące zmienną do możliwości odczytu przez blok „EdgeBroker”



Rys. 2.10: Blok „EdgeBroker” wysyłający sygnał o zmianie stanu uszkodzenia do konfiguratora NAZCA

Przykładowa próbka danych wejściowych sieci.

Fluid_level_tank102	Pressure_tank103	Pump_speed	Flowmeter	Fluid_lvl_tank101_IO_link	Fluid_lvl_tank102_IO_link	V101	V102	V103	V104	V106	V107	V108	V109	V112
78.58	8.16	0.00	0.00	80	80	0	0	0	0	0	0	0	0	0
78.58	8.16	0.00	0.00	80	80	0	0	0	0	0	0	0	0	0
78.58	8.16	0.00	0.00	80	80	0	0	0	0	0	0	0	0	0
78.58	8.16	0.00	0.00	80	80	0	0	0	0	0	0	0	0	0
78.58	8.16	0.00	0.00	80	80	0	0	0	0	0	0	0	0	0
...
78.58	197.43	96.79	0.86	80	80	0	0	0	1	0	0	0	0	0
78.58	197.68	98.03	0.86	80	80	0	0	0	1	0	0	0	0	0
78.58	197.68	99.28	0.86	80	80	0	0	0	1	0	0	0	0	0
78.75	197.43	100.00	0.85	80	80	0	0	0	1	0	0	0	0	0
78.58	197.43	100.00	0.85	80	80	0	0	0	1	0	0	0	0	0

Rys. 2.8: Dane wejściowe sieci zebrane w czasie testów cz.1

cycle_counter	State	Liquid_growth_B102	Pipe_clogging_factor	Flow_diff	Pressure_diff	Pressure_flow_ratio	Pressure_pumpspeed_ratio	Tanks_level_diff	Fluid_level_dependency
11	4	0.00	0.00	0.00	-8.16	0.00	0.00	1.42	0.00
11	4	0.00	0.00	0.00	-8.16	0.00	0.00	1.42	0.00
11	4	0.00	0.00	0.00	-8.16	0.00	0.00	1.42	0.00
11	4	0.00	0.00	0.00	-8.16	0.00	0.00	1.42	0.00
11	4	0.00	0.00	0.00	-8.16	0.00	0.00	1.42	0.00
...
1	3	0.43	4.84	0.05	-197.43	169.79	19108.88	1.42	65.41
1	3	0.43	4.90	0.05	-197.68	170.01	19378.69	1.42	66.25
1	3	0.43	4.96	0.05	-197.68	170.01	19625.79	1.42	67.09
1	3	0.43	6.00	0.06	-197.43	167.81	19742.61	1.25	66.94
1	3	0.43	6.00	0.06	-197.43	167.81	19742.61	1.42	66.79

Rys. 2.9: Dane wejściowe sieci zebrane w czasie testów cz.1

Aby zagregować dane potrzebne jest skonfigurowanie nowego modułu w systemie NAZCA 4.

Zmienna	Wartość
Fluid_level_tankI02	92,885
Pressure_tankI03	7,652
Pump_speed	0
Flowmeter	0
State	9
cycle_counter	1
tankI02_set	0
Pressure_set	0
Flow_set	0
Fluid_lvl_tankI01_IO_link	70
Fluid_lvl_tankI02_IO_link	95
Work	false
Fluid_lvl_MAX_tankI01	false
Fluid_lvl_MIN_tankI01	false
VI01	false
VI02	false
VI03	false
VI04	false
VI06	false
VI07	false
VI08	false
VI09	false
VII2	false
EStop	false
Failure_F1	false
Failure_F2	false
Failure_F3	false
Failure_F4	false
Failure_F5	false

Rys. 2.10: Zrzut ekranu z modułu ze zagregowanymi danymi w bazie Kafka

2.3. Wyciągnięcie cyklu z sygnału

Praca stanowiska pozwala na wyciągnięcie cyklu z każdego zmierzonego sygnału. Jeden cykl polega na przejściu stanowiska przez 8 stanów funkcjonalnych pracy, od 2 do 9. Poniżej przedstawiono po jednym cyklu wszystkich sygnałów, które zebrano:

- Sygnały związane z zachowaniem cieczy w układzie



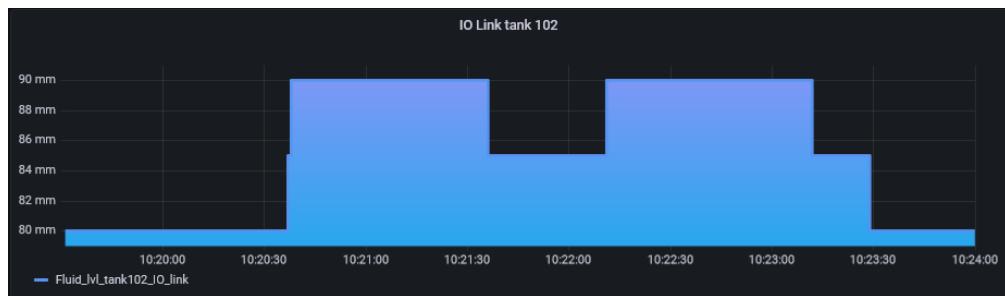
Rys. 2.11: Wykresy ciśnienia cieczy w zbiorniku B103 oraz poziomu cieczy w zbiorniku B102



Rys. 2.12: Wykresy przepływu cieczy w układzie oraz zadanej prędkości pompy



Rys. 2.13: Wykres wysokości słupa cieczy w zbiorniku B101 zebranego z czujnika KQ1000

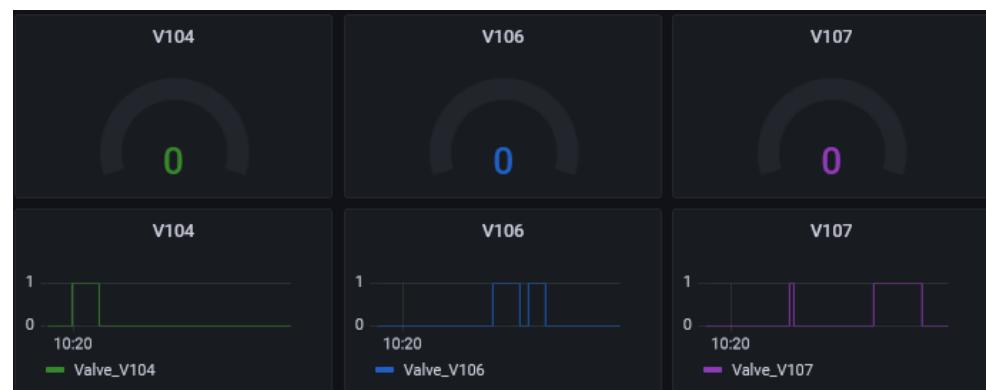


Rys. 2.14: Wykres wysokości słupa cieczy w zbiorniku B102 zebranego z czujnika KQ1000

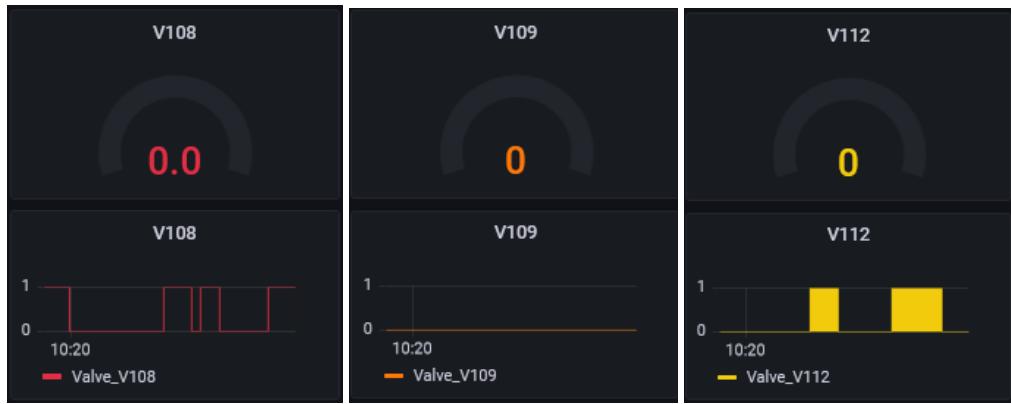
- Sygnały związane z otwarciami zaworów układu



Rys. 2.15: Otwarcia zaworów V101 – V103



Rys. 2.16: Otwarcia zaworów V104 – V107



Rys. 2.17: Otwarcia zaworów V108 – V112

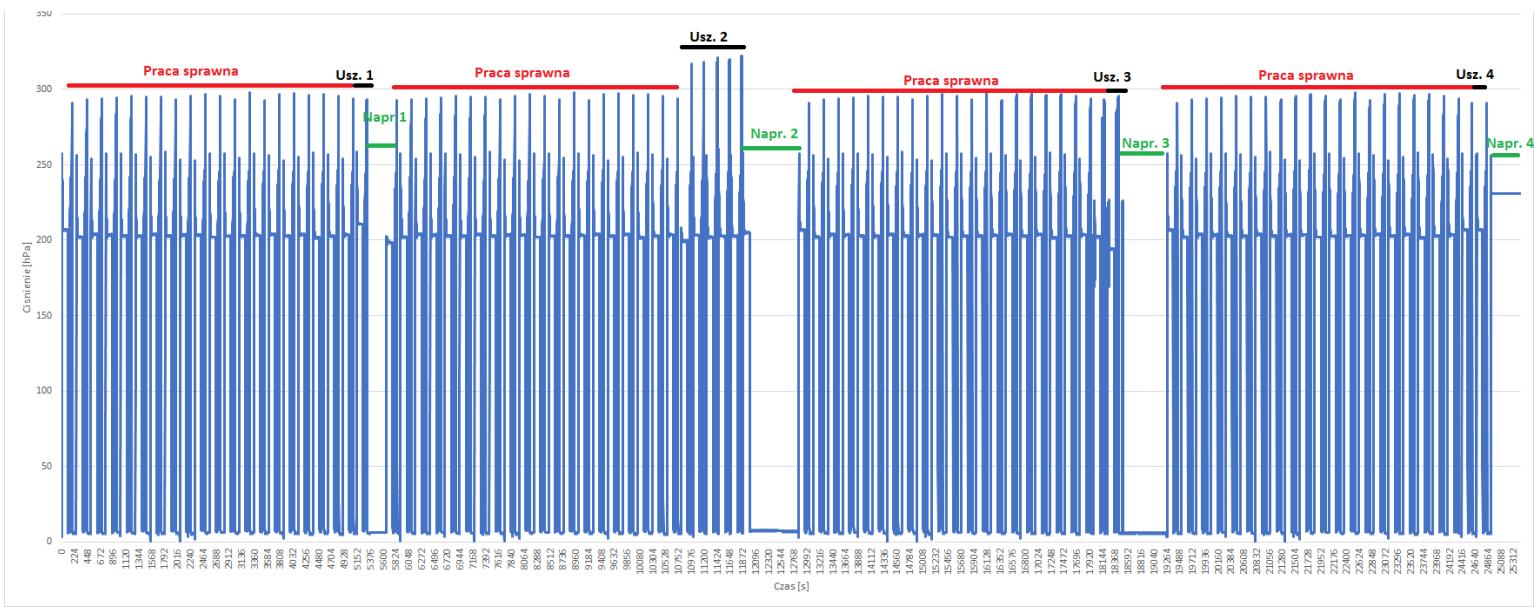
2.4. Wyznaczenie trendu i sezonowości

W momencie gdy zostały zebrane dane do wyznaczenia trendu i sezonowości stanowisko pracowało jeszcze na starym przepływomierzu oraz symulowanym innym uszkodzeniu F4, które polegało na niesprawnej pracy czujnika maksymalnego poziomu cieczy w zbiorniku B101. Dane pozyskane były bezpośrednio ze sterownika dzięki bibliotece snap7, poprzez utworzoną dodatkowo aplikację w języku Python, aby można było w razie potrzeby zbierać dane ze stanowiska również bez dostępu do NAZCA. Na podstawie zebranych danych wyznaczono następujące sezony.

Tab. 2.2: Przedstawienie działania wyznaczonych sezonów

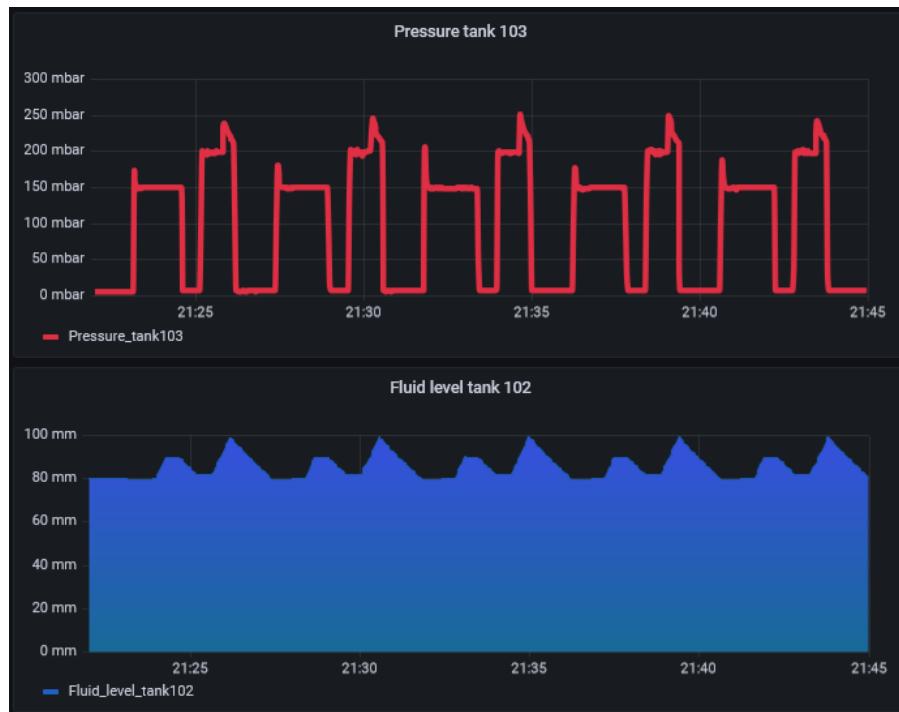
Sezon 1 (przepływ - 1,5 l/min, ciśnienie 200 mbar)	Sezon 2 (przepływ 0.7 l/min, ciśnienie - 150 mbar)
1. Tryb sprawnej pracy - 20 cykli 2. Tryb z pierwszym uszkodzeniem (kożuch) - 1 cykli 3. Brak pracy stanowiska (naprawa uszkodzenia) - 1 cykli 4. Tryb sprawnej pracy - 20 cykli 5. Tryb z drugim uszkodzeniem (wyciek) - 5 cykle 6. Brak pracy stanowiska (naprawa uszkodzenia) - 3 cykli 7. Tryb sprawnej pracy - 20 cykli 8. Tryb z trzecim uszkodzeniem (cyberatak) - 2 cykle 9. Brak pracy stanowiska (naprawa uszkodzenia)- 3 cykle 10. Tryb sprawnej pracy - 20 cykli 11. Tryb z czwartym uszkodzeniem (uszkodzenie czujnika poziomu) - 2 cykle 12. Brak pracy stanowiska (naprawa uszkodzenia) - 2 cykle	1. Tryb sprawnej pracy - 20 cykli 2. Tryb z pierwszym uszkodzeniem (kożuch) - 1 cykli 3. Brak pracy stanowiska (naprawa uszkodzenia) - 1 cykli 4. Tryb sprawnej pracy - 20 cykli 5. Tryb z drugim uszkodzeniem (wyciek) - 5 cykle 6. Brak pracy stanowiska (naprawa uszkodzenia) - 3 cykli 7. Tryb sprawnej pracy - 20 cykli 8. Tryb z trzecim uszkodzeniem (cyberatak) - 2 cykle 9. Brak pracy stanowiska (naprawa uszkodzenia)- 3 cykle 10. Tryb sprawnej pracy - 20 cykli 11. Tryb z czwartym uszkodzeniem (uszkodzenie czujnika poziomu) - 2 cykle 12. Brak pracy stanowiska (naprawa uszkodzenia) - 2 cykle

Przykładowo, można zaobserwować pracę stanowiska w jednym z sezonów na podstawie sygnału z czujnika ciśnienia w zbiorniku B103.



Rys. 2.18: Przebieg sygnału ciśnienia w całym sezonie z opisem

Jako, że ciężko zauważać różnice między stanem poprawnej pracy stanowiska a uszkodzeniami w całym sezonie na jednym wykresie, poniżej przedstawiono wykresy z 5 cykli pracy.

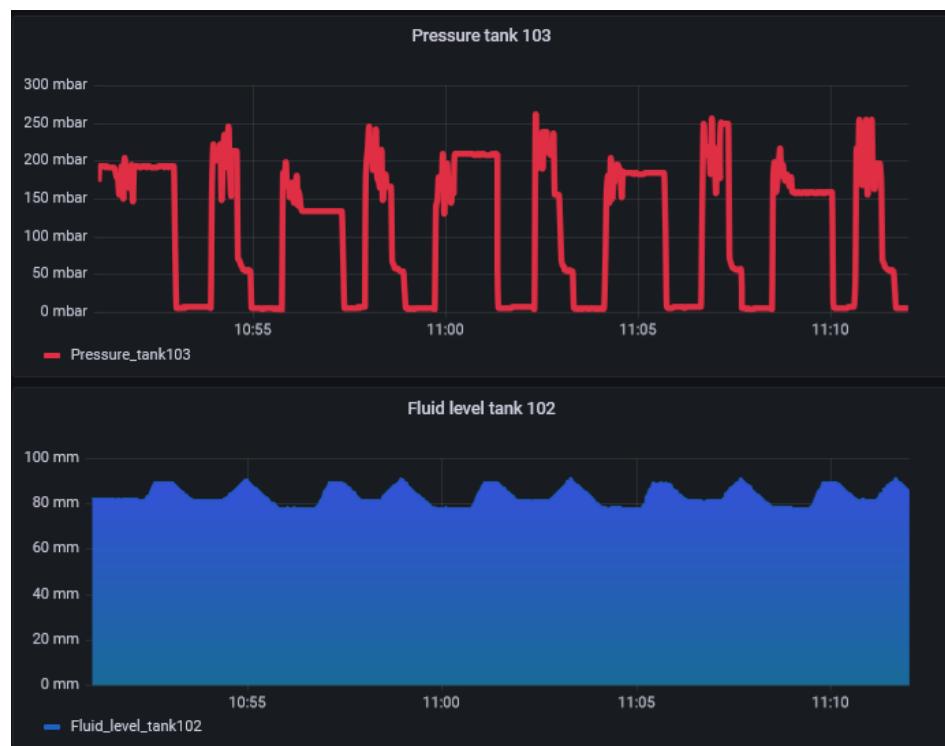


Rys. 2.19: Wykresy przedstawiające ciśnienie w zbiorniku B103 oraz poziom cieczy w zbiorniku B102 dla 5 cykli pracy stanowiska

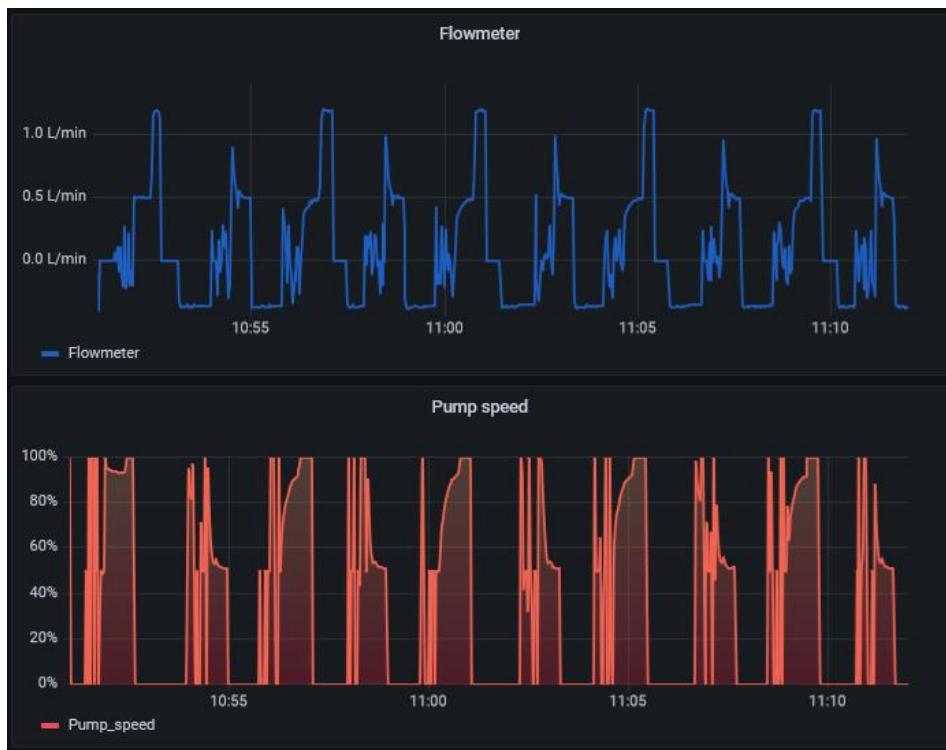


Rys. 2.20: Wykresy przedstawiające przepływ oraz prędkość pomy dla 5 cyklów pracy stanowiska

Dla porównania, przedstawiono wykresy powyższych sygnałów, również dla 5 cykli pracy, lecz tym razem z uszkodzeniem F3.



Rys. 2.21: Wykresy przedstawiające ciśnienie w zbiorniku B103 oraz poziom cieczy w zbiorniku B102 dla 5 cyklów pracy stanowiska przy uszkodzeniu F3



Rys. 2.22: Wykresy przedstawiające przepływ oraz prędkość pomy dla 5 cyklów pracy stanowiska przy uszkodzeniu F3

2.5. Określenie współczynników

Współczynniki jakie wyznaczono to:

- **MTTR** – średni czas potrzebny do naprawy systemu oraz przywrócenia go do pełnej funkcjonalności. Do obliczenia tego współczynnika wykorzystano wzór:

$$MTTR = \frac{\sum_{i=1}^N T_i}{N}$$

N – liczba awarii

T_i – czas awarii

- **MTBF** – średni czas jaki upływał między naprawialną awarią sprzętu a następnym jej wystąpieniem. Do obliczenia tego współczynnika wykorzystano wzór:

$$MTBF = MTTR + MTTF$$

- **MTTF** – średni czas jaki upływał między nienaprawialnymi awariami sprzętu. Do obliczenia tego współczynnika wykorzystano wzór:

$$MTTF = \frac{T_D - \sum_{i=1}^N T_i}{N + 1}$$

T_D – czas dostepny maszyny lub grupy maszyn

N – liczba awarii

T_i – czas awarii

- **MTTA** – średni czas potrzebny organizacji na reakcję na reklamacje, przerwy w działaniu lub incydenty we wszystkich działach. Do obliczenia tego współczynnika wykorzystano wzór:

$$MTTA = \frac{T_r}{N}$$

T_r - całkowity czas między ostrzezeniem a potwierdzeniem

N – liczba awarii

2.6. Wyliczenie wskaźników za pomocą narzędzia JUPYTER

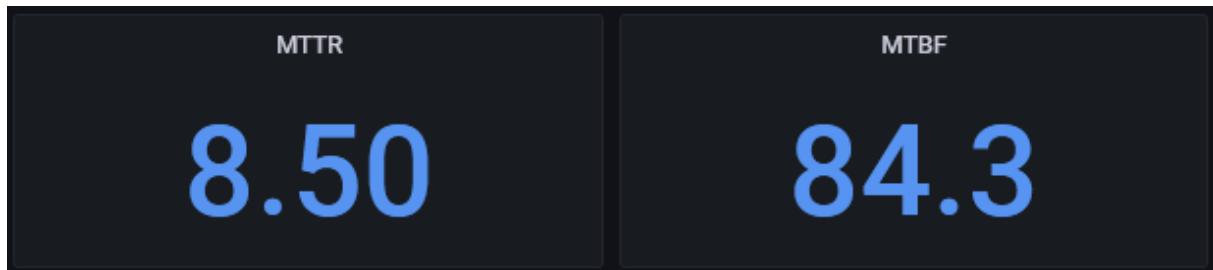
Do obliczenia wyznaczonych wskaźników stworzono program w języku Python, który na podstawie danych w pliku csv, oblicza potrzebne wskaźniki [1]:

- Wyniki obliczonych współczynników:

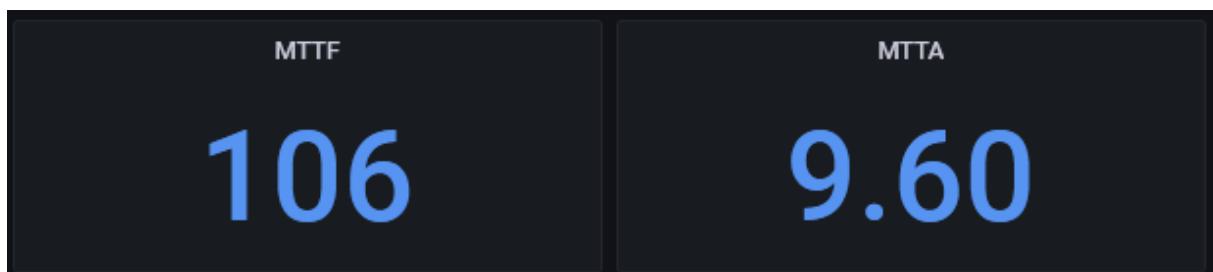
Tab. 2.3: Wykaz współczynników pracy stanowiska

Wskaźnik	Wynik [minuty]
MTTR	8,525
MTBF	84,314
MTTF	105,589
MTTA	9,606

Utworzony algorytm wykorzystano później w docelowym programie wywoływanym w NiFi, co pozwala na wyliczanie wskaźników w czasie rzeczywistym. Poniżej przedstawiono wizualizacje wyznaczonych wskaźników po godzinie pracy układu.



Rys. 2.23: Obliczone wskaźniki MTTR oraz MTBF przez algorytm wywoływany w programie docelowym



Rys. 2.24: Obliczone wskaźniki MTTF oraz MTTA przez algorytm wywoływany w programie docelowym

3. Część związana z uczeniem maszynowym – analiza eksploracyjna danych (EDA) wraz z analizą statystyczną

3.1. Wyznaczenie zależności pomiędzy zmiennymi

Po analizie 24 sygnałów, które wchodzą na wejście sieci, można było zauważyc, że najbardziej wpływające na zachowanie się sieci będą sygnały z czujników przepływu, poziomu cieczy w zbiornikach, ustawionej prędkości pompy oraz ciśnienia. Na ich podstawie zostały wyznaczone następujące zależności:

- Liquid_growth_B102 – Zależność pokazująca przewidywany przyrost cieczy w mm/s w zbiorniku B102 na podstawie zmierzonego przepływu oraz wymiarów zbiornika.
- Pipe_clogging_factor – Zależność dzięki której można zaobserwować stopień zatkania rur w sekcji gdzie mierzony jest przepływ cieczy do zbiornika B101
- Flow_diff – Różnica między zadanym przepływem a zmierzonym
- Pressure_diff – Różnica między ciśnieniem zadanym a zmierzonym
- Pressure_flow_ratio – Iloczyn zmierzonego ciśnienia oraz przepływu
- Pressure_pumpspeed_ratio – Iloraz zmierzonego ciśnienia i zadanej prędkości pompy
- Tanks_level_diff – Różnica między poziomami cieczy w zbiorniku B101 a B102
- Fluid_level_dependancy – Iloczyn zmierzonego poziomu cieczy, przepływu oraz zadanej prędkości pompy

Poniżej pokazano przykłady wyliczonych zależności podczas poprawnej pracy stanowiska zebrane z 40 sekund pracy.

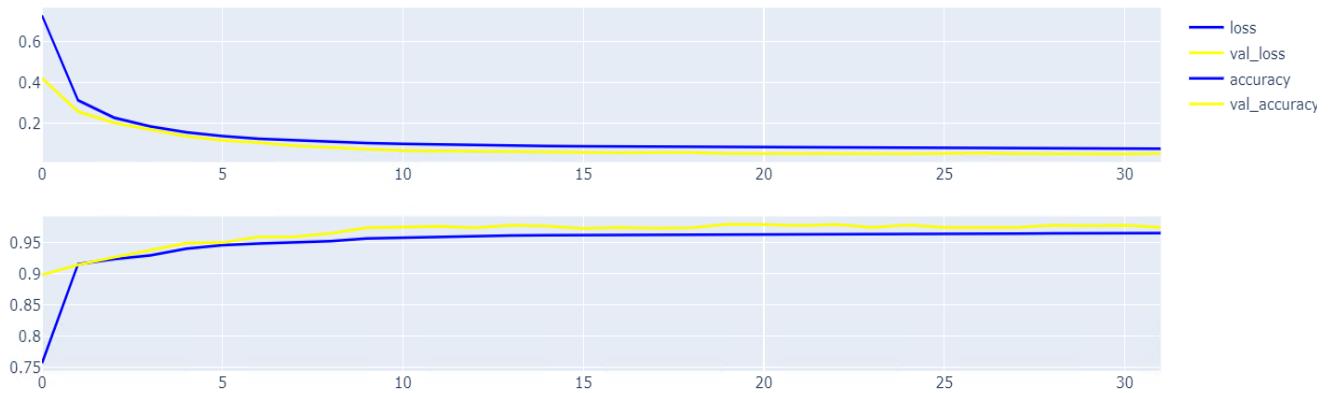
Liquid_growth_B102	Pipe_clogging_factor	Flow_diff	Pressure_diff	Pressure_flow_ratio	Pressure_pumpspeed_ratio	Tanks_level_diff	Fluid_level_dependency
0.06	0.0	-0.12	-21.41	20.57	13486.89	12.09	7.36
0.0	0.0	0.0	-9.17	0.0	11634.4	12.09	0.0
0.0	0.0	0.0	-4.83	0.0	11193.48	12.09	0.0
0.0	0.0	0.0	-2.28	0.0	10960.9	12.09	0.0
0.0	0.0	0.0	-0.49	0.0	10784.15	12.09	0.0
0.0	0.0	0.0	-0.49	0.0	10723.86	12.09	0.0
0.0	0.0	0.0	1.55	0.0	10572.58	12.09	0.0
0.0	0.0	0.0	2.82	0.0	10473.49	12.09	0.0
0.0	0.0	0.0	1.29	0.0	10590.22	12.09	0.0
0.0	0.0	0.0	1.55	0.0	10525.74	12.09	0.0
0.0	0.0	0.0	0.02	0.0	10673.2	12.09	0.0
0.0	0.0	0.0	2.82	0.0	10480.41	12.09	0.0
0.0	0.0	0.0	3.33	0.0	10409.82	12.09	0.0
0.0	0.0	0.0	0.78	0.0	10621.37	12.09	0.0
0.0	0.0	0.0	1.55	0.0	10575.11	12.09	0.0
0.0	0.0	0.0	1.8	0.0	10528.1	12.09	0.0
0.0	0.0	0.0	-0.49	0.0	10702.22	12.09	0.0
0.0	0.0	0.0	3.08	0.0	10439.17	12.09	0.0
0.0	0.0	0.0	3.33	0.0	10457.25	12.09	0.0
0.0	0.0	0.0	-0.49	0.0	10671.05	12.09	0.0
0.0	0.0	0.0	0.02	0.0	10662.09	12.09	0.0
0.0	0.0	0.0	-0.75	0.0	10715.93	12.09	0.0
0.0	0.0	0.0	-0.75	0.0	10727.59	12.09	0.0
0.0	0.0	0.0	2.82	0.0	10466.9	12.09	0.0
0.0	0.0	0.0	-1.0	0.0	10776.68	12.09	0.0
0.0	35.5	0.5	-148.2	0.0	10521.02	12.09	0.0
0.11	17.44	0.28	-146.92	32.32	9150.29	11.92	10.7
0.16	11.9	0.18	-146.92	47.01	9711.53	12.09	16.48
0.19	9.25	0.13	-146.92	54.36	10457.89	12.09	20.52
0.19	8.81	0.12	-146.92	55.83	10784.06	12.09	21.73
0.19	8.92	0.12	-146.67	55.73	10900.27	12.09	22.0
0.2	7.43	0.1	-146.67	58.67	10900.27	12.09	23.16
0.22	5.62	0.07	-146.67	63.07	11768.54	12.09	26.88
0.22	5.69	0.07	-146.67	63.07	11912.27	12.09	27.21
0.22	5.81	0.07	-146.67	63.07	12183.6	12.09	27.83
0.23	3.38	0.04	-146.67	67.47	12375.74	12.09	30.24
0.23	3.4	0.04	-146.92	67.58	12475.13	12.09	30.43
0.24	1.72	0.02	-146.67	70.4	12632.4	12.09	32.21
0.23	3.48	0.04	-146.67	67.47	12774.67	12.09	31.21

Rys. 3.1: Zrzut ekranu z zapisu danych zależności w pliku csv w środowisku JUPYTER

3.2. Selekcja i ekstrakcja cech sygnałowych na podstawie, których można przeprowadzić wnioskowanie

Po utworzeniu działającej sieci neuronowej, wykonano analizę, które dane najbardziej wpływają na wyniki, jakie przewiduje sieć. Wiedząc, że proces uczenia sieci nie trwa długo, oraz że istnieje możliwość skrócenia go wykorzystując do obliczeń kartę graficzną, wykorzystano metodę prób i błędów. Z 32 danych wejściowych sieci po kolej odrzucano pojedyncze sygnały i sprawdzano na podstawie danych walidacyjnych oraz macierzy pomyłek jak uczy się sieć oraz kiedy poprawniej przewiduje uszkodzenia.

- Dokładność sieci oraz przebieg uczenia bez usuwania zmiennych



Rys. 3.2: Wykres przebiegu uczenia się sieci neuronowej bez usuwania zmiennych

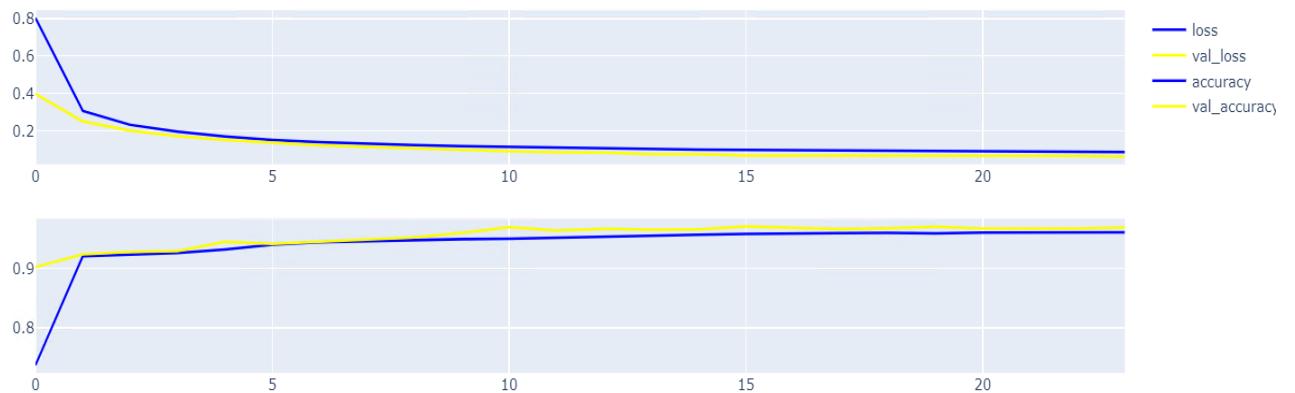
Macierz 1.1: Macierz pomylek sieci neuronowej, dokładność wynosi 97,39 %

2573	0	5	1	0
0	380	0	0	0
86	0	378	0	0
1	0	0	129	0
1	0	0	0	54

Tab. 3.1: Dokładność z jaką sieć neuronowa z wszystkimi zmiennymi rozpoznaje i klasyfikuje stany działania zdalnego stanowiska

Nazwa klasy	Dokładność działania sieci [%]
Brak uszkodzeń	99,77
Korzuch	100
Wyciek	81,46
Cyberatak	99,23
Przytykanie się rur	98,18

- Dokładność sieci oraz przebieg uczenia po usunięciu dodanych zależności



Rys. 3.3: Wykres przebiegu uczenia się sieci neuronowej po usunięciu zależności

Macierz 3.2: Macierz pomyłek sieci neuronowej, dokładność całkowita wynosi 97,39 %

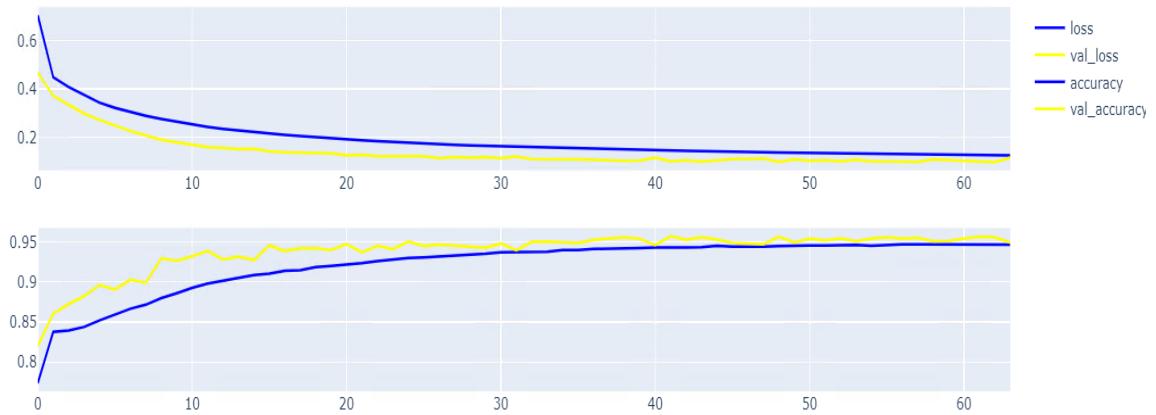
2571	0	5	3	0
0	380	0	0	0
79	0	385	0	0
21	0	0	109	0
1	0	0	0	54

Tab. 3.2: Dokładność z jaką sieć neuronowa bez dodanych zależności rozpoznaje i klasyfikuje stany działania zdalnego stanowiska

Nazwa klasy	Dokładność działania sieci [%]
Brak uszkodzeń	99,68
Korzuch	100
Wyciek	82,97
Cyberatak	83,84
Przytykanie się rur	98,18

Po usunięciu dodatkowych sygnałów, sieć zaczęła gorzej rozpoznawać uszkodzenie 3, pozostałe uszkodzenia wykrywała bez większych zmian.

- Dokładność sieci oraz przebieg uczenia po usunięciu zmiennych otwarcia zaworów



Rys. 3.4: Wykres przebiegu uczenia się sieci neuronowej po zmiennych otwarcia zaworów

Macierz 3.3: Macierz pomyłek sieci neuronowej bez zmiennych otwarcia zaworów, dokładność całkowita wynosi 94,90 %

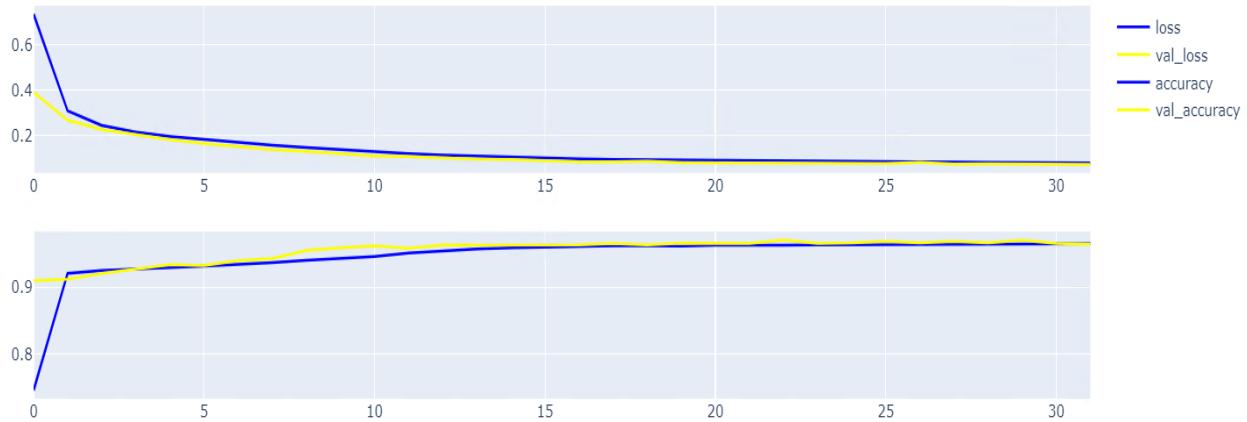
2572	0	4	3	0
0	380	0	0	0
173	0	291	0	0
3	0	0	127	0
1	0	0	0	54

Tab. 3.3: Dokładność z jaką sieć neuronowa bez zmiennych otwarcia zaworów rozpoznaje i klasyfikuje stany działania zdalnego stanowiska

Nazwa klasy	Dokładność działania sieci [%]
Brak uszkodzeń	99,72
Korzuch	100
Wyciek	62,71
Cyberatak	97,69
Przytykanie się rur	98,18

Po usunięciu otwarć zaworów sieć znacznie gorzej zaczęła wykrywać uszkodzenie 2

- Dokładność sieci oraz przebieg uczenia po usunięciu zmiennych z czujników maksymalnego i minimalnego poziomu cieczy w zbiorniku B101, czujników poziomu cieczy IO-LINK oraz zmiennej Estop i Work



Rys. 3.5: Wykres przebiegu uczenia się sieci neuronowej po usunięciu sygnałów z MAX i MIN poziomu cieczy B101, czujników poziomu cieczy KQ1000 oraz zmiennej „Estop” i „work”

Macierz 3.4: Macierz pomyłek sieci neuronowej bez zmiennych z czujników maksymalnego i minimalnego poziomu cieczy w zbiorniku B101, czujników poziomu cieczy IO-LINK oraz zmiennej „Estop” i „work”, dokładność całkowita wynosi 94,90 %

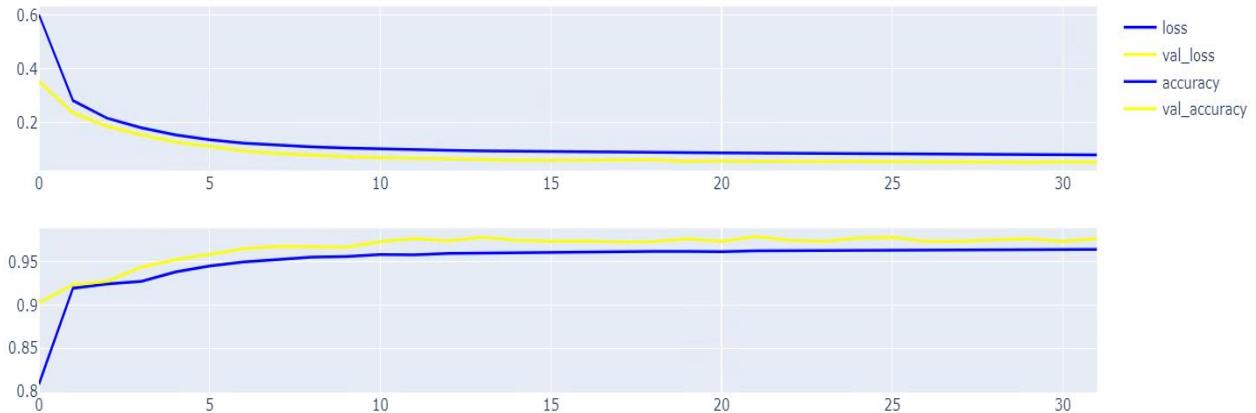
2555	0	20	4	0
0	380	0	0	0
79	0	385	0	0
23	0	0	107	0
2	0	0	0	53

Tab. 3.4: Dokładność z jaką sieć neuronowa bez zmiennych otwarcia zaworów rozpoznaje i klasyfikuje stany działania zdalnego stanowiska

Nazwa klasy	Dokładność działania sieci [%]
Brak uszkodzeń	99,06
Korzuch	100
Wyciek	82,97
Cyberatak	82,30
Przytykanie się rur	96,36

Usunięcie tych zmiennych spowodowało, że sieć nieco lepiej zaczęła rozpoznawać uszkodzenie 2 ale również nieco gorzej uszkodzenie 3.

- Dokładność sieci oraz przebieg uczenia po usunięciu zmiennych Estop, Work, poziomów cieczy MIN/MAX w zbiorniku B101



Rys. 3.6: Wykres przebiegu uczenia się sieci neuronowej po usunięciu sygnałów Estop i work oraz poziomów MIN/MAX cieczy B101

Macierz 3.5: Macierz pomyłek sieci neuronowej bez zmiennych z czujników maksymalnego i minimalnego poziomu cieczy w zbiorniku B101, czujników poziomu cieczy IO-LINK oraz zmiennej „Estop” i „work”, dokładność całkowita wynosi 94,90 %

2555	0	20	4	0
0	380	0	0	0
79	0	385	0	0
23	0	0	107	0
2	0	0	0	53

Tab. 3.5: Dokładność z jaką sieć neuronowa bez zmiennych otwarcia zaworów rozpoznaje i klasyfikuje stany działania zdalnego stanowiska

Nazwa klasy	Dokładność działania sieci [%]
Brak uszkodzeń	99,06
Korzuch	100
Wyciek	82,97
Cyberatak	82,30
Przytykanie się rur	96,36

Po usunięciu tych zmiennych, można zauważyc, że nie mają one wpływu na uczenie się sieci, więc można przyjąć, że uczenie sieci bez tych sygnałów będzie lepszym rozwiązaniem.

- Dokładność sieci oraz przebieg uczenia po usunięciu zmiennych zadanych wartości sygnałów (FlowSet, LvLtanSet, PressureSet)



Rys. 3.7: Wykres przebiegu uczenia się sieci neuronowej po usunięciu zmiennych zadanych wartości

Macierz 3.6: Macierz pomyłek sieci neuronowej bez zmiennych zadanych wartości sygnałów „FlowSet”, „LvLtanSet”, „PressureSet”, dokładność całkowita wynosi 97,28 %

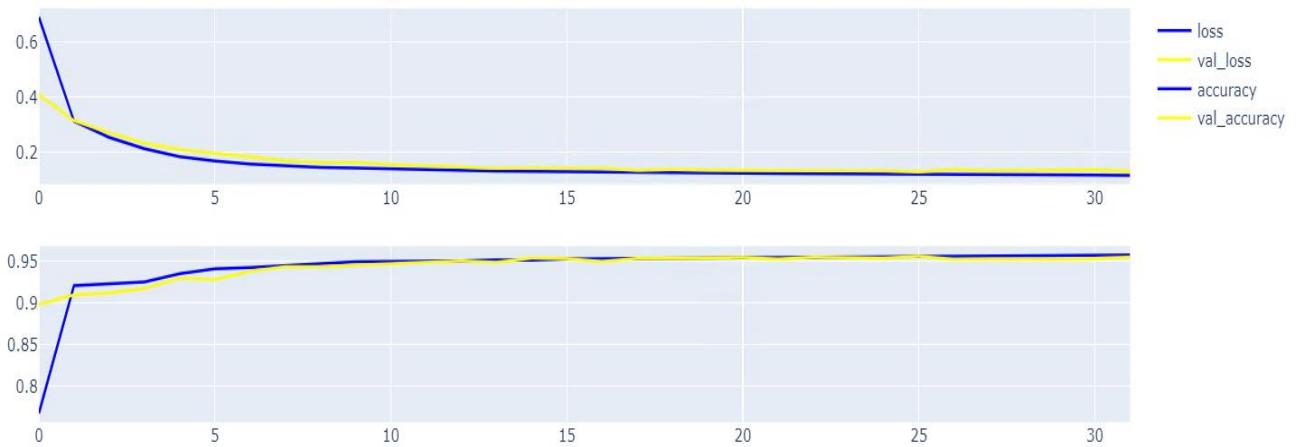
2571	0	6	2	0	
0	380	0	0	0	
88	0	376	0	0	
1	0	0	129	0	
1	0	0	0	54	

Tab. 3.6: Dokładność z jaką sieć neuronowa bez zmiennych zadanych wartości sygnałów „FlowSet”, „LvLtanSet”, „PressureSet”, rozpoznaje i klasyfikuje stany działania zdalnego stanowiska

Nazwa klasy	Dokładność działania sieci [%]
Brak uszkodzeń	99,68
Korzuch	100
Wyciek	81,03
Cyberatak	99,23
Przytykanie się rur	98,18

Usunięcie tych sygnałów również nie wpłynęło znacząco na dokładność sieci.

- Dokładność sieci oraz przebieg uczenia po usunięciu zmiennej Flowmeter



Rys. 3.8: Wykres przebiegu uczenia się sieci neuronowej po usunięciu danych z przepływu

Macierz 3.7: Macierz pomyłek sieci neuronowej bez zmiennej „Flowmeter”, dokładność całkowita wynosi 95,50 %

2573	0	2	2	2	
0	380	0	0	0	
130	0	334	0	0	
23	0	0	107	0	
3	0	0	0	52	

Tab. 3.7: Dokładność z jaką sieć neuronowa bez zmiennej „Flowmeter”, rozpoznaje i klasyfikuje stany działania zdalnego stanowiska

Nazwa klasy	Dokładność działania sieci [%]
Brak uszkodzeń	99,76
Korzuch	100
Wyciek	71,9
Cyberatak	82,3
Przytykanie się rur	94,54

Po usunięciu tej zmiennej spada dokładność wykrywania uszkodzenia 2 oraz 3.

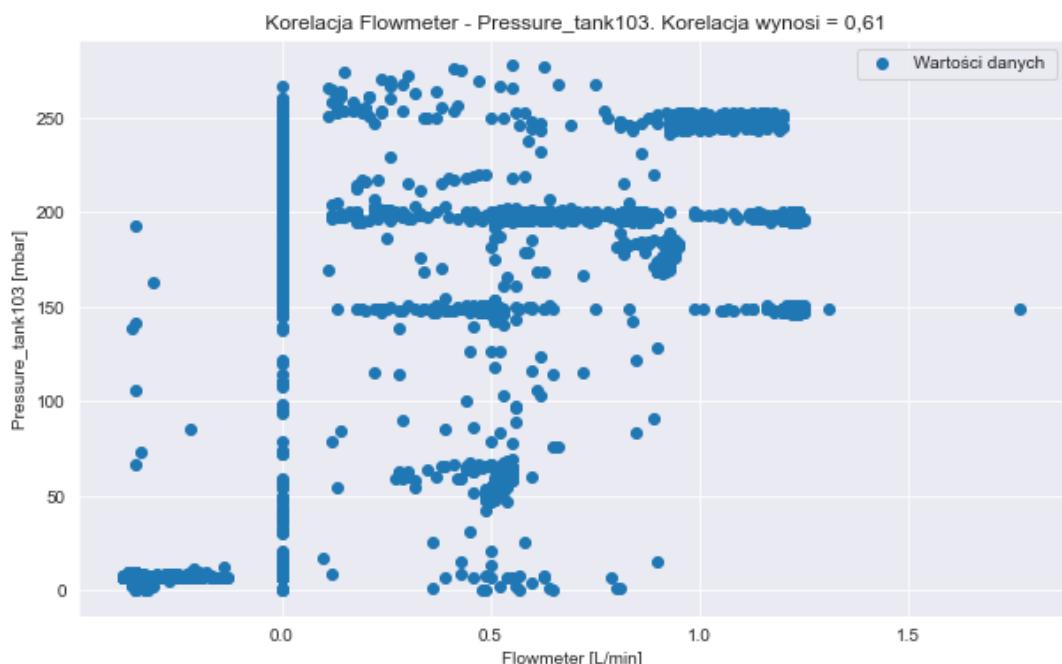
Końcowo zdecydowano, że najlepiej sieć działa przy wariancie gdy odrzuci się zmienne:

- Fluid_lvl_MAX_tank101
- Fluid_lvl_MIN_tank101
- EStop
- Work
- Flow_set
- Tank102_set
- Pressure_set

3.3. Analiza statystyczna zmiennych

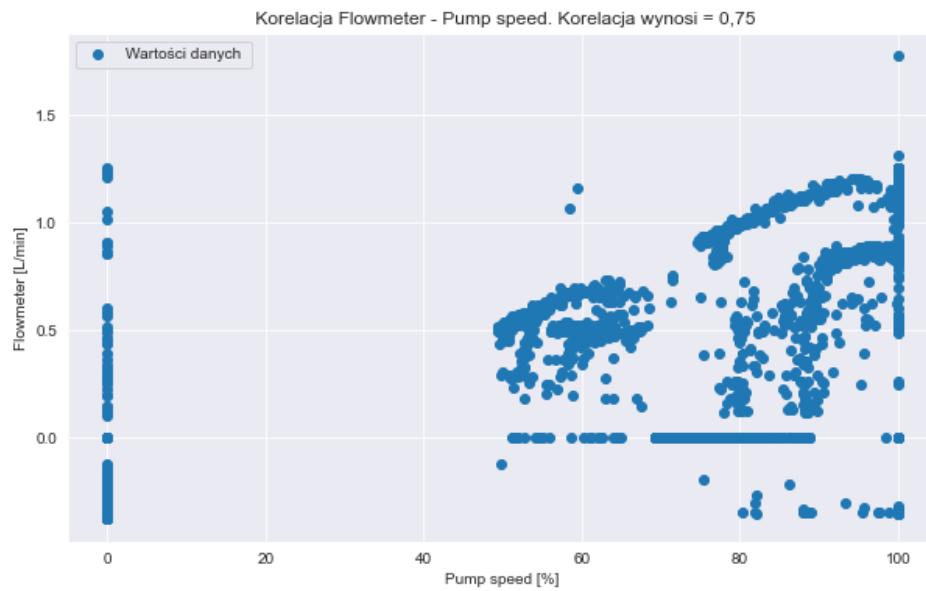
Aby lepiej zrozumieć zmienne procesowe oraz zależności między ważniejszymi zmiennymi policzono ich odchylenia standardowe oraz korelacje, wykreślono wykresy oraz wyliczono współczynniki korelacji. Wyniki analizy przedstawiono poniżej.

- Zależność między przepływem a zmianą ciśnienia, ponieważ występuje ona tylko w dwóch stanach pracy stanowiska (2 i 7).



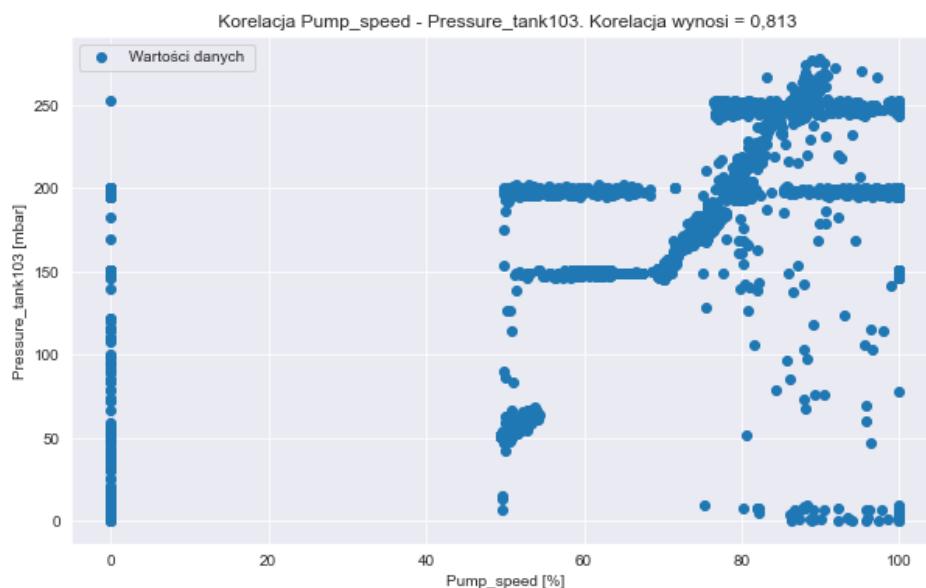
Rys. 3.9: Wykres korelacji przepływu oraz ciśnienia w zbiorniku B103 na podstawie danych z stanu pełnej zdatności

- Na poniższym wykresie można zauważać średnią zależność między prędkością pompy a przepływem cieczy, ponieważ w niektórych stanach pompa wytwarza ciśnienie np. do zbiornika B103, przez co przepływ nie występuje.



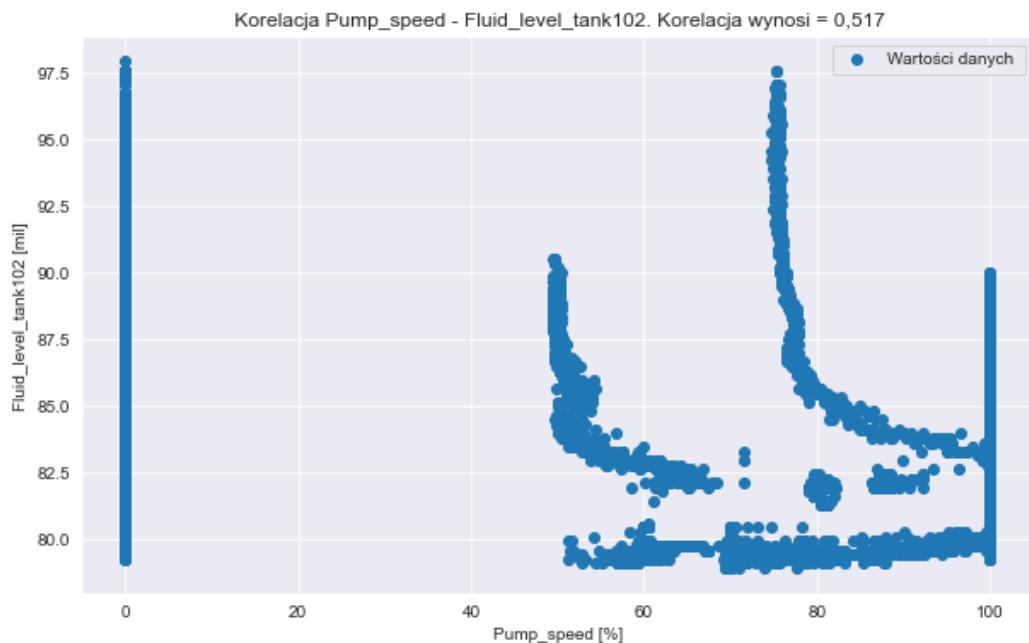
Rys. 3.10: Wykres korelacji zadanej prędkości pompy oraz przepływu na podstawie danych z stanu pełnej zdatności

- Zależność między ciśnieniem w zbiorniku B103 a prędkością pompy jest silniejsza, ponieważ każda zmiana ciśnienia związana jest z pracą pompy w układzie, poza stanami gdy ciśnienie jest wypuszczane ze zbiornika.



Rys. 3.11: Wykres korelacji zadanej prędkości pompy oraz ciśnienia na podstawie danych z stanu pełnej zdatności

- Między prędkością pompy a poziomem cieczy w zbiorniku B102 również występuje mała zależność, polega ona na dopompowywaniu cieczy do tego zbiornika i regulacji poziomu słupa cieczy.



Rys. 3.12: Wykres korelacji zadanej prędkości pompy oraz wysokości poziomu cieczy w zbiorniku B102 na podstawie danych z stanu pełnej zdatności

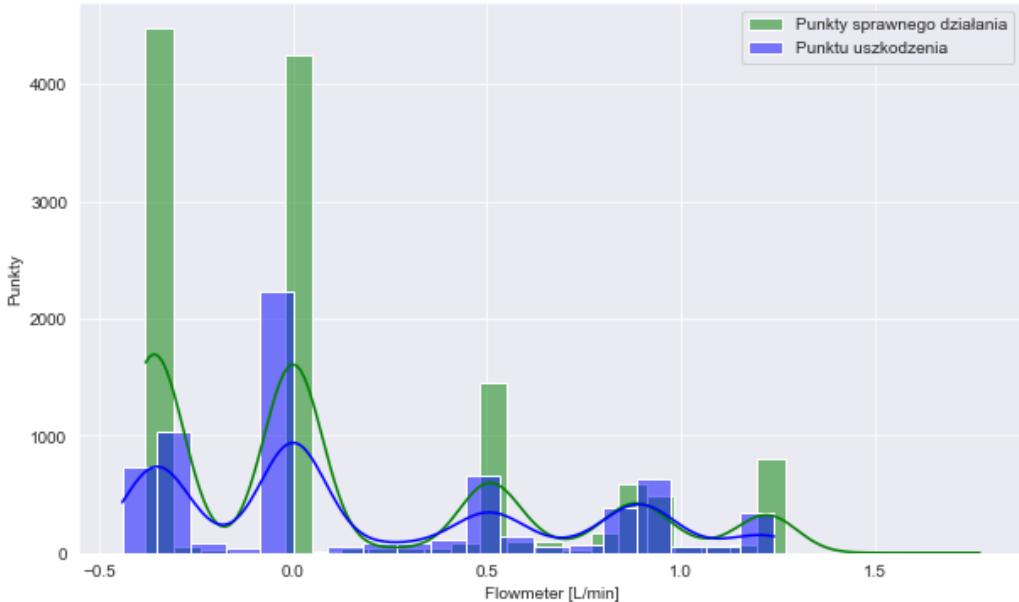
Odchylenia standardowe potrzebne do wyliczenia korelacji obliczono w programie Jupyter:

Tab. 3.8: Wyniki programu do obliczania odchylenia standardowego

Odchylenie standardowe zmiennej:	Wynik
„Flowmeter”	0.5022
„Pump_speed”	40.5118
„Pressure_tank103”	89.6900
„Fluid_level_tank102”	4.4659

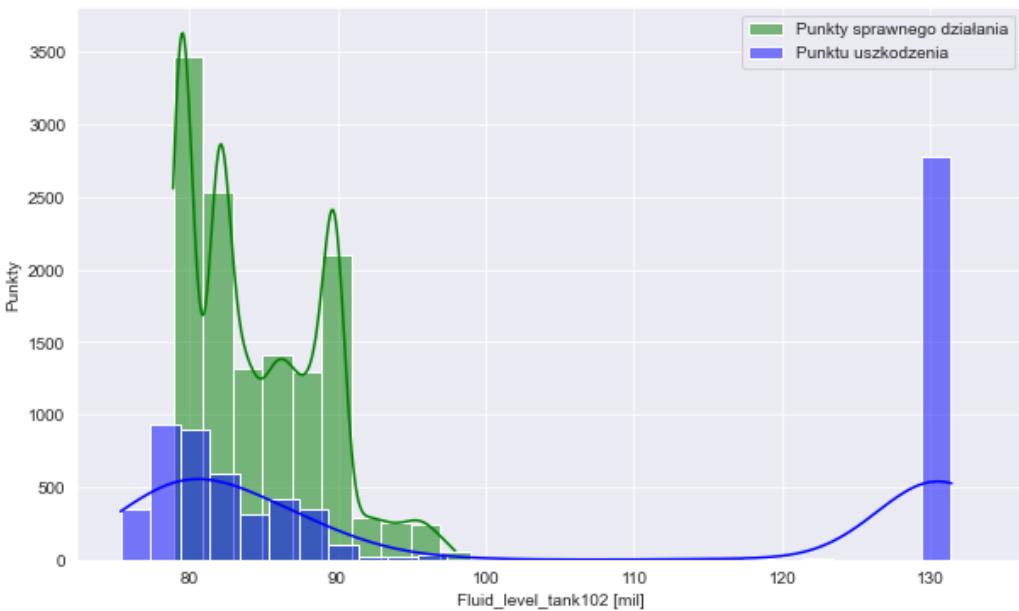
Aby lepiej zaobserwować rozkłady zmiennych oraz ich liczebność wyznaczono również histogramy dla ważniejszych zmiennych. Zielone kolumny na wykresie przedstawiają dane z stanu pełnej zdatności natomiast na niebiesko przedstawiono dane tej samej zmiennej ze stanów gdy występuje uszkodzenie. Danych ze stanu pełnej zdatności jest zdecydowanie więcej. Największe różnice widoczne są na wykresach z "Fluid_level_tank102" oraz "Flowmeter", w których wartości ze stanów z uszkodzeniem znaczco różniły się od danych stanu pełnej zdatności. W pozostałych wykresach wartości w większości się pokrywają.

- Na histogramie przepływu można zaobserwować mniejszy przepływ w czasie gdy stanowisko pracowało podczas awarii niż przy całkowicie sprawnej pracy.



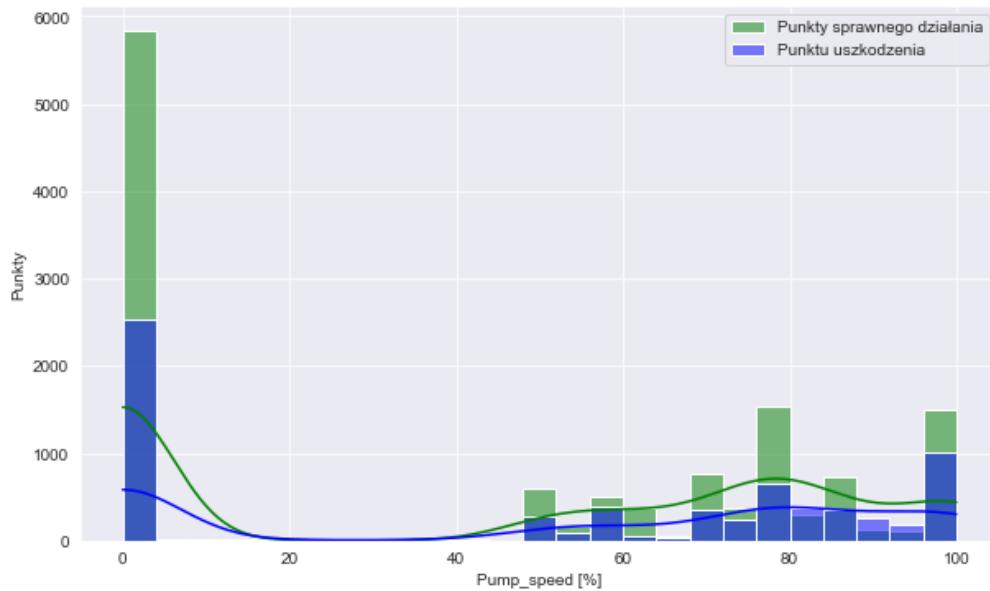
Rys. 3.13: Histogram przepływu

- Na histogramie poziomu cieczy, można zauważyć, że przy uszkodzeniu stanowiska przepływ cieczy w dużych ilościach pokazywał wartość znacznie przekraczającą normę (około 130 mm).



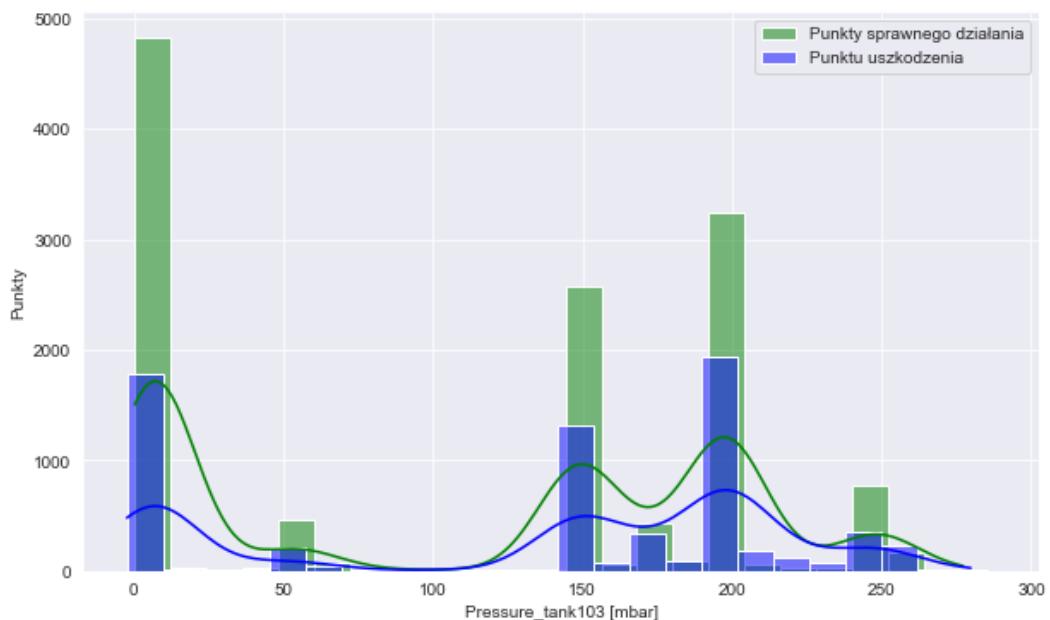
Rys. 3.14: Histogram poziomu cieczy w zbiorniku B102

- Na histogramie prędkości pompy, można zauważać bardzo małe różnice, jedynie krótszą pracę i momentami mniej stabilną.



Rys. 3.15: Histogram zadanej prędkości pompy

- Obserwując histogram ciśnienia w zbiorniku B103, można zaobserwować, że przy uszkodzeniu zmiany były gwałtowniejsze oraz wykraczały poza normę.



Rys. 3.16: Histogram ciśnienia

3.4 Dobór algorytmu do klasy problemu

W dalszej części raportu przedstawiono wyniki w zakresie zaproponowania algorytmu sztucznej inteligencji do problemu klasyfikacji lub predykcji.

Stanowisko wykonuje proces wolnozmienny, możliwe jest zatem sprawdzanie czy system działa poprawnie co dłuższy okres czasu, w naszym przypadku algorytm ma wykrywać uszkodzenie na podstawie danych sprzed 4 minut (tyle około trwa jeden cykl stanowiska) oraz sklasyfikować jaki to typ uszkodzenia (jedno z czterech symulowanych).

Do tego typu rozwiązania, zaproponowano sieć neuronową typu FeedForward oraz CNN. Aby sprawdzić, która sieć bardziej nadaje się do problemu, utworzono obie od podstaw używając gotowych bibliotek Tensorflow Keras [4].

Następnie porównano obie sieci neuronowe sprawdzając wyniki macierzy pomyłek.

Macierz 3.8: Macierz pomyłek sieci neuronowej CNN, dokładność całkowita wynosi 96,72 %

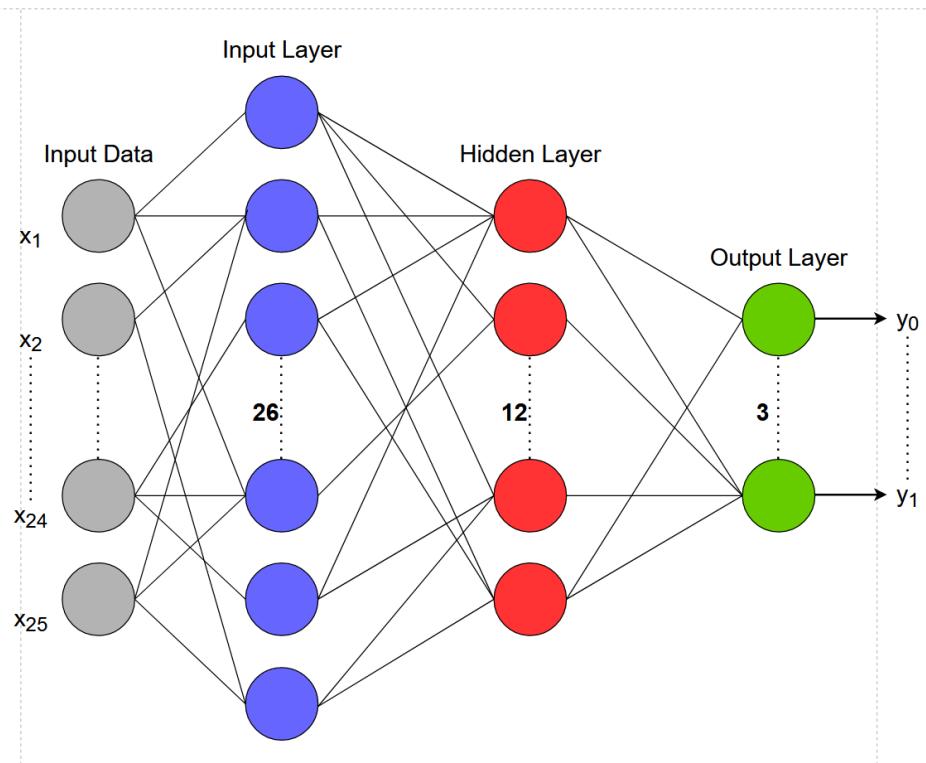
2549	0	26	2	2
0	380	0	0	0
61	0	403	0	0
26	0	0	104	0
3	0	0	0	54

Macierz 3.9: Macierz pomyłek sieci neuronowej FeedForward, dokładność całkowita wynosi 96,78 %

2573	0	5	1	2
0	380	0	0	0
87	0	377	0	0
22	0	0	108	0
3	0	0	0	54

Wybrano sieć typu FeedForward, ponieważ wykorzystanie bardziej skomplikowanych algorytmów oraz typów sieci mijałoby się z celem, dokładności obu sieci były bardzo podobne a skomplikowanie sieci CNN oraz czas uczenia się były większe.

Poniżej przedstawiono schemat budowy utworzonej sieci.



Rys. 3.17: Uproszczony schemat budowy wykorzystywanej sieci neuronowej

Utworzona sieć składa się z trzech warstw. Warstwą pierwszą jest warstwa wejściowa, jest ona typu „dense”, posiada 32 neurony, oraz aktywator typu „relu”. Danymi wejściowymi jest wektor o rozmiarze 1×25 . Kolejną warstwą jest warstwa ukryta, będąca typu „dense”, która posiada 16 neuronów i aktywator „relu”. Ostatnią warstwą jest warstwa wyjściowa, która jest typu „dense”, posiadająca 5 neuronów, oraz aktywator typu „softmax”. Wybranym algorytmem uczącym sieci jest „Adam”, a wybrana metryka jest typu „accuracy”. Na podstawie wykresu funkcji strat obliczanej dla danych trenujących oraz walidacyjnych wyznaczona została odpowiednia liczba epok uczenia modelu, która wynosi 32 epoki.

3.5 Przygotowanie danych do procesu uczenia i weryfikacji algorytmu

Do uczenia sieci zebrano nowe dane zapisane w plikach csv, trenujące po 40 cykli na sezon dla stanu F0 oraz po 10 cykli z każdego uszkodzenia na sezon. Wykorzystano do tego program „Data_download” [3] napisany w JUPYTER udostępniony przez firmę APA.

/ AI-Meth-konkurs /	
Name	Last Modified
Damage_detection.ipynb	an hour ago
Data_download.ipynb	3 days ago

Rys. 3.18: widok programu utworzonego w środowisku JUPYTER

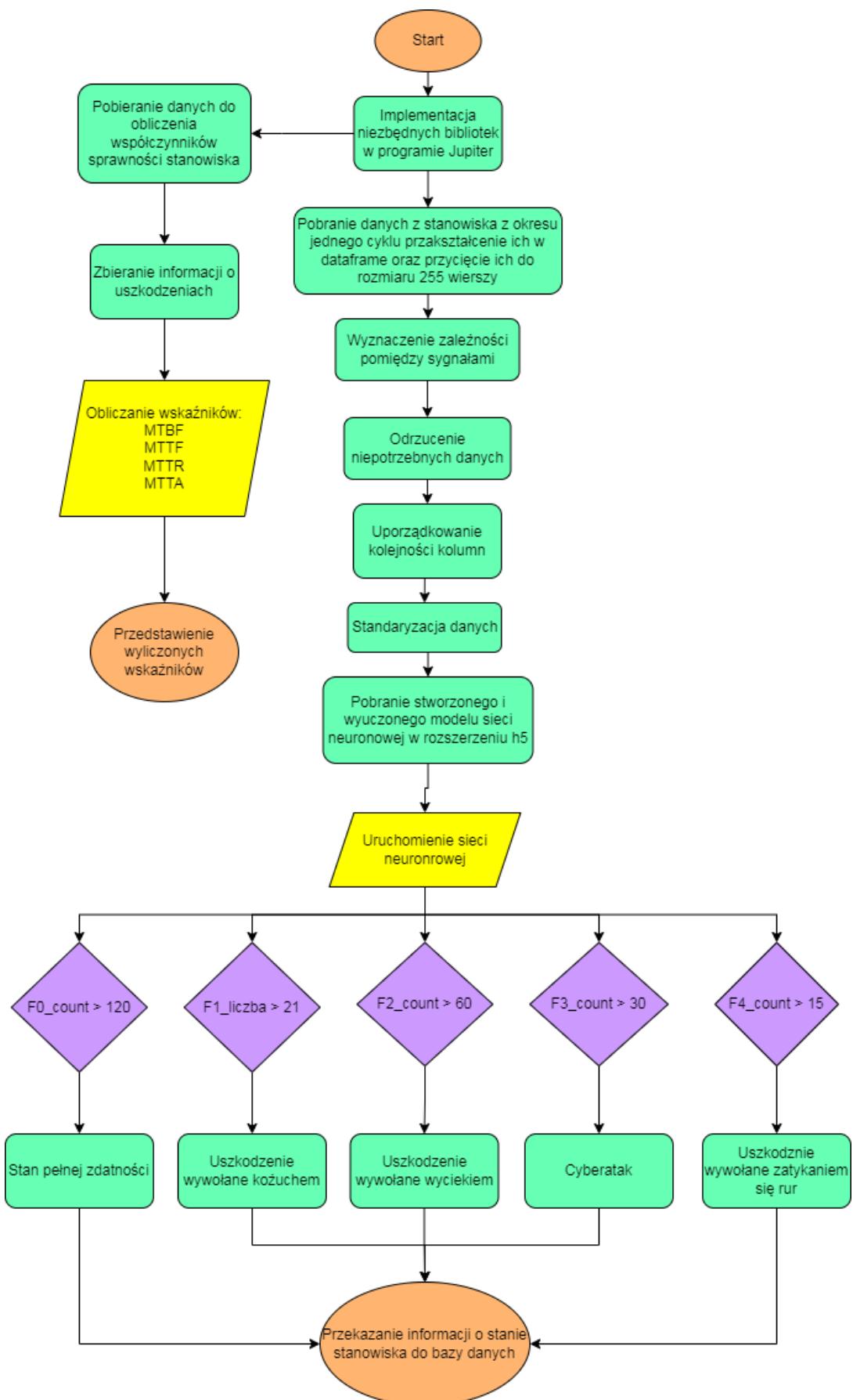
Program po pobraniu surowych danych dzięki bibliotece SDK, na początku zapisuje je do Dataframe. Później usuwa zmienne, które nie są potrzebne do uczenia sieci. Następnie dla zebranych danych obliczane są wyznaczone wcześniej zależności między zmiennymi, ponieważ pomagają one w uzyskaniu lepszej dokładności sieci. Tak przygotowane dane zapisywane są do pliku csv, po czym dzielone są na zbiory: trenujący, testowy, oraz walidacyjny. Ze względu na charakterystykę pracy stanowiska oraz zebranych danych (tryb pracy w całych cyklach, przy procesie wolnozmiennym) dane podzielono ręcznie, tak aby uzyskać dokładne próbki z całych cykłów (np. w danych walidacyjnych znajduje się próbka po 1 cyklu z uszkodzeniem dla sezonu i 4 cykle poprawnej pracy dla sezonu). Zachowane proporcje przedstawiono w Tab. 3.9.

Tab. 3.9: proporcje podzielności danych

Rodzaj zbioru:	Całkowity procent objętości danych
trenujący	80%
testowy	15%
walidacyjny	5%

3.6 Analiza uzyskanych rezultatów

W programie Jupiter utworzony został program diagnostyczny stanowiska laboratoryjnego. Z obiektu pobierane zostają dane z czujników poprzez system NAZCA, który przekazuje dane do serwera Kafka skąd informacje pozyskuje NAZCA 4.0. Program pobiera liczbę wierszy odpowiadającą czasowi jednego cyklu stanowiska następnie dane są obrabiane. Wczytana zostaje wyuczona sieć neuronowa typu FeedForward. Sieć sprawdza wcześniej przygotowane zmienne wiersz po wierszu a wyniki zapisywane są do tablicy. W części postprocessingu zliczone zostają wartości tablicy a następnie program w zależności od liczby zliczonych poszczególnych cyfr reprezentujących stany podaje aktualny wyznaczony ostatecznie stan stanowiska. Schemat programu przedstawiono poniżej.



Rys. 3.19: Schemat blokowy działania programu detekcji uszkodzeń

Poniżej przedstawiono weryfikację poprawności pracy algorytmu i wyciągnięcie odpowiednich wniosków na podstawie nowego zbioru danych.

Po przetestowaniu sieci na docelowych danych wejściowych zbieranych co 4 minuty, można zobaczyć, że algorytm zwraca liczbę odpowiadającą przewidywanemu uszkodzeniu lub stanu pełnej sprawności, dla każdego zebranego wektora danych.

Rys. 3.20: Lista 1 wykrywania stanu pełnej zdatności

Rys. 3.21: Lista 2 wykrywania uszkodzenia 1

Rys. 3.22: Lista 3 wykrywania uszkodzenia 2

Rys. 3.23: Lista 4 wykrywania uszkodzenia 3

Rys. 3.24: Lista 5 wykrywania uszkodzenia 4

Jak widać z rysunków zamieszczonych powyżej, sieć popełnia błędy podczas wykrywania uszkodzenia, lecz jeżeli weźmie się pod uwagę większość zwracanych informacji to na ich podstawie można stwierdzić, które uszkodzenie wystąpiło. Takie dane zmuszają do zastosowania postprocessingu, dzięki któremu można określić, jakie uszkodzenie zostało przewidziane dla całego cyklu wykonanego przez stanowisko.

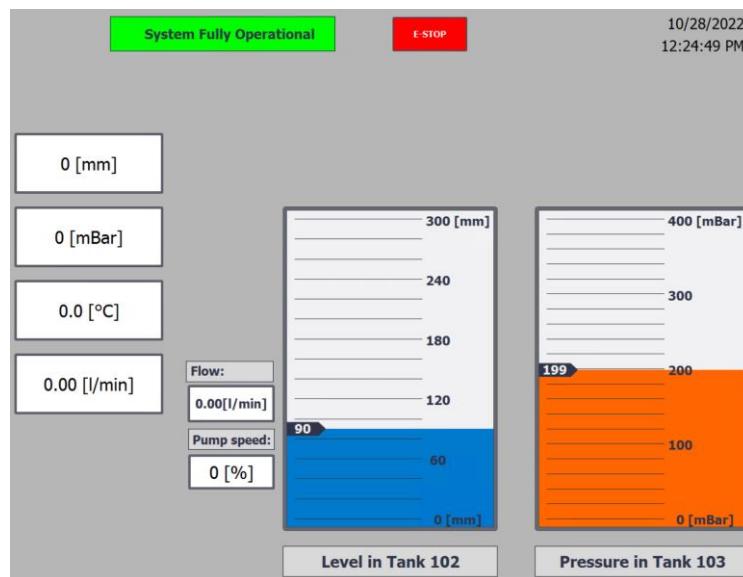
Do głównego kodu programu „*Damage_detection*” [2] dodano algorytm, który na podstawie liczby zwracanych identyfikatorów odpowiadających uszkodzeniu określa, które uszkodzenie powinna przewidzieć sieć.

- Po wykonaniu postprocessingu danych wyjściowych sieci uzyskano następujące rezultaty:

Tabela 3.10: Wykrycie przez sieć neuronową stanu pełnej zdolności

Wyniki macierzy wykryć sieci		Wynik rozpoznanego stanu pracy	
Nazwa stanu	Ilość wykryć	Nazwa stanu	Rozpoznanie stanu
	Seria 1		Seria 1
F0_count	189	Failure_F0	1
F1_count	5	Failure_F1	0
F2_count	34	Failure_F2	0
F3_count	9	Failure_F3	0
F4_count	2	Failure_F4	0

Wynik algorytmu dzięki połączeniu Sterownika PLC z bazą danych NAZCA można wyświetlić w systemie SCADA, dzięki czemu łatwo zauważyc uszkodzenie stanowiska i szybko je wyłączyć aby zapobiec dalszym stratom.



Rys.3.25: Zrzut ekranu z wizualizacji stanu pełnej zdolności



Rys.3.26: Zrzut ekranu z przedstawienia stanu pełnej zdolności w grafana

Tabela 3.11: Wykrycie przez sieć neuronową stanu uszkodzenia spowodowanego przez kożuch

Wyniki macierzy wykryć sieci		Wynik rozpoznanego stanu pracy	
Nazwa stanu	Ilość wykryć	Nazwa stanu	Rozpoznanie stanu
	Seria 1 Seria 2		Seria 1 Seria 2
F0_count	173 181	Failure_F0	0 0
F1_count	30 31	Failure_F1	1 1
F2_count	36 27	Failure_F2	0 0
F3_count	0 0	Failure_F3	0 0
F4_count	0 0	Failure_F4	0 0



Rys.3.27: Zrzut ekranu z wizualizacji kożucha (uszkodzenia F1)



Rys.3.26: Zrzut ekranu z przedstawienia uszkodzenia F1 (kożuch) w grafana

Tabela 3.12: Wykrycie przez sieć neuronową stanu uszkodzenia spowodowanego wyciekiem z zbiornika 102

Wyniki macierzy wykryć sieci		Wynik rozpoznanego stanu pracy	
Nazwa stanu	Ilość wykryć	Nazwa stanu	Rozpoznanie stanu
	Seria 1		Seria 1
F0_count	149	Failure_F0	0
F1_count	17	Failure_F1	0
F2_count	61	Failure_F2	1
F3_count	12	Failure_F3	0
F4_count	0	Failure_F4	0



Rys.3.27: Zrzut ekranu z wizualizacji wycieku (uszkodzenia F2)



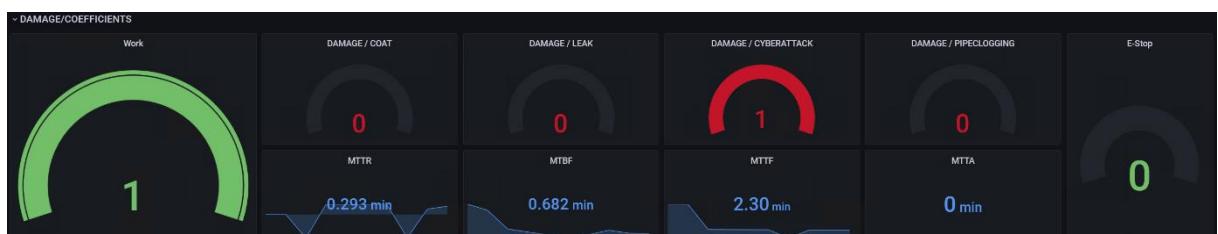
Rys.3.26: Zrzut ekranu z przedstawienia uszkodzenia F2 (wyciek) w grafana

Tabela 3.13: Wykrycie przez sieć neuronową stanu uszkodzenia spowodowanego cyberatakiem

Wyniki macierzy wykryć sieci			Wynik rozpoznanego stanu pracy		
Nazwa stanu	Ilość wykryć		Nazwa stanu	Rozpoznanie stanu	
	Seria 1	Seria 2		Seria 1	Seria 2
F0_count	160	161	Failure_F0	0	0
F1_count	9	9	Failure_F1	0	0
F2_count	27	27	Failure_F2	0	0
F3_count	43	42	Failure_F3	1	1
F4_count	0	0	Failure_F4	0	0



Rys.3.28: Zrzut ekranu z wizualizacji cyberataku (uszkodzenia F3)



Rys.3.26: Zrzut ekranu z przedstawienia uszkodzenia F3 (cyberatak) w grafana

Tabela 3.14: Wykrycie przez sieć neuronową stanu uszkodzenia spowodowanego zatykaniem się rur

Wyniki macierzy wykryć sieci		Wynik rozpoznanego stanu pracy		
Nazwa stanu	Ilość wykryć	Nazwa stanu	Rozpoznanie stanu	
	Seria 1		Seria 1	
F0_count	202	179	Failure_F0	0
F1_count	5	15	Failure_F1	0
F2_count	13	24	Failure_F2	0
F3_count	0	0	Failure_F3	0
F4_count	19	21	Failure_F4	1



Rys.3.29: Zrzut ekranu z wizualizacji zatykania się rur (uszkodzenia F4)



Rys.3.26: Zrzut ekranu z przedstawienia uszkodzenia F4 (zatkanie rur) w grafana

4. Wnioski

Przy pomocy programu NAZCA można zebrać dane ze sterownika PLC, które następnie zostają zapisane w bazie danych Kafka. Za pomocą modułu Jupyter, można dokonać eksploracji danych która jest potrzebna do utworzenia działającego algorytmu sztucznej inteligencji do wykrycia nieprawidłowego działania stanowiska. Z użyciem NAZCA 4.0 oraz Nifi, główny kod programu jest wykonywany cyklicznie, a dane zapisywane są w bazie danych, z której można je wysłać z powrotem do sterownika PLC, lub przy pomocy modułu Grafana, można odczytać, oraz przedstawić na wykresach.

Zaproponowany algorytm sztucznej inteligencji z pomocą postprocessingu wykrywa wszystkie uszkodzenia. Osiąga dużą skuteczność oraz ogranicza liczbę fałszywych alarmów. W przypadku wszelkich zmian na stanowisku wymagane jest zebranie nowych danych, za pomocą których można douszczyć sieć, co sprawi, że sieć osiągnie większą dokładność wykrywania uszkodzeń.

5. Bibliografia

- [1] Pierwszy program wykorzystywany do obliczania wskaźników sprawności Stanowiska:
http://157.158.62.247:10128/user/nazca/lab/tree/AI-Meth-konkurs/Wyliczanie_wskaznikow.ipynb
- [2] Program główny zawierający wszystkie potrzebne algorytmy do działania systemu detekcji i klasyfikacji uszkodzeń: http://157.158.62.247:10128/user/nazca/lab/tree/AI-Meth-konkurs/Damage_detection.ipynb
- [3] Program, wykorzystywane do pobrania danych w celu uczenia/douczania sieci neuronowej: http://157.158.62.247:10128/user/nazca/lab/tree/AI-Meth-konkurs/Data_download.ipynb
- [4] Dokumentacja tensorflow keras: https://www.tensorflow.org/api_docs/python/tf/keras