



Faustyna Misiura
119041
Mechatronika

**Integracja cyberbezpieczeństwa i sztucznej inteligencji w predykcyjnym sterowaniu
systemami IoT infrastruktury krytycznej**

Praca magisterska

Praca wykonana pod kierunkiem
dr inż. Dariusz Bober

Rzeszów, 2025

Spis treści

Wstęp.....	4
1.1 Cel i zakres pracy	5
1.2. Metodologia badawcza	6
1.3 Struktura pracy.....	8
2. Przegląd literatury z zakresu przedmiotu pracy	9
2.1. IoT w infrastrukturze krytycznej – przegląd technologii.....	9
2.2. Zbieranie danych w systemach IoT oraz ich transfer.....	13
2.3. Dostępne metody predykcyjnego sterowania systemami IoT	14
2.4 Współczesne cyberzagrożenia systemów IoT	16
3. Zdalne sposoby przesyłania danych i ich przetwarzania.....	22
3.1.Zastosowanie podejścia typu Cloud AI.....	22
3.2 Porównanie podejścia typu TinyML z Cloud AI	24
3.3 Przetwarzanie danych oraz budowa modeli ML	25
4. Cyberbezpieczeństwo w IoT	27
4.1 Zagrożenia oraz zabezpieczenia dla systemów IoT	27
5. Opracowanie systemu predykcyjnego dla magazynu energii IoT	30
5.1 Cel i funkcjonalność systemu	30
5.1 Platformy i narzędzia wykorzystane w implementacji systemu	33
5.2 Inteligentne sterowanie IoT oparte na algorytmach AI.....	34
5.3 Wykorzystanie agentów AI do inteligentnego sterowania	43
5.4 Proces sterowania	45
6. Analiza anomalii w zbieranych danych z sensorów	52
6.1.Metody detekcji i klasyfikacji anomalii	55
6.2. Badania anomalii braku wartości i ich redukcja	55
7. Analiza i testowanie aplikacji pod względem bezpieczeństwa	57

7.1. Testowanie ataków lokalnie	57
7.1.1. Przebieg testów JWT i autoryzacji	58
7.1.2. Testy bezpieczeństwa przed atakami Cross-Site-Scripting	60
7.1.3. Testowanie na SQL Injection	61
7.1.4. Testy rejestracji i logowania	62
7.1.5. Testy logowania Google OAuth2	63
7.2. Testowanie aplikacji w wersji produkcyjnej	64
7.2.1. Testy przeprowadzone w programie OWASP ZAP	65
7.2.2. Testy fuzz endpointu create	69
7.2.3. Testy fuzz endpointu users	70
7.2.4. Wnioski z testowania serwera backendowego oraz jego endpointów ...	72
7.2.5. Badanie frontendu aplikacji pod względem bezpieczeństwa	72
7.2.6. Wnioski ze skanu front-endowej części aplikacji	73
7.2.7. Testy bezpiecznego połączenia backendu Rendera	73
7.3 Ocena wyników badań i propozycje rozwiązań	74
8. Dyskusja uzyskanych wyników	76
8.1. Ocena wdrożonych modeli i agentów	76
8.2. Wpływ zakłóceń jakoś predykcji i sterowanie systemem	77
8.3 Dalszy rozwój badań	77
9. Podsumowanie	78
Bibliografia	79
Spis rysunków, spis tabel	83

Wstęp

Problematyka niniejszej pracy jest interdyscyplinarnym połączeniem wyzwań jakie niesie zastosowanie Internet of Things w infrastrukturze krytycznej oraz możliwości jakie daje sztuczna inteligencja w rozwiązywaniu konkretnych problemów, w tym aspektów związanych z cyberbezpieczeństwem. Wyzwania z którymi zmierzono się w pracy to:

- ocena zastosowania sztucznej inteligencji w sterowaniu urządzeniami IoT przeznaczonymi do kontroli obiektów infrastruktury krytycznej;
- porównanie wybranych technik i technologii sterowania urządzeniami typu IoT opartego na AI, spośród opisanych w literaturze zagadnienia;
- ocena współczesnych zagrożeń i zabezpieczeń od strony cybernetycznej dla urządzenia IoT, w szczególności na warstwę aplikacyjną kontrolującą pracę prototypowego układu magazynu energii;
- ogólna problematyka w podejmowaniu decyzji przez użytkownika systemu, a podejmowanie tych decyzji przez agenta AI.

Wobec tak zarysowanych wyzwań i możliwości jakie niesie połączenie prostych układów IoT i technik sztucznej inteligencji do kontroli pracy elementów infrastruktury krytycznej, sformułowano następujące problemy badawcze:

- hipoteza 1: analiza i detekcja anomalii w danych gromadzonych w czasie rzeczywistym oraz opracowanie metody ich redukcji poprawi precyzję predykcji oczekiwanych wartości zbieranych przez czujniki IoT;
- hipoteza 2: opracowanie prototypowego układu magazynu energii i zastosowanie do jego kontroli agenta LLM jest możliwe i pozwoli zastąpić podejmowanie decyzji przez człowieka, a tym samym automatyzację pracy układu, w czasie gdy użytkownik go nie kontroluje (np, podczas urlopu, czy pobytu „off-line”);
- hipoteza 3: opracowanie modeli predykcyjnych oraz analiza skuteczności tych modeli trenowanych na danych z anomaliami i bez anomalii pozwoli wyłonić model optymalny do sterowania prototypem magazynu energii;
- hipoteza 4: przeprowadzenie testów podatności warstwy aplikacyjnej zarządzającej urządzeniami IoT pozwala ograniczyć ryzyko cyberataków na wybraną infrastrukturę krytyczną – problem mitygacji ryzyk warstwy cyfrowej.

Na potrzeby przeprowadzenia badań nad powyższymi problemami, opracowano stanowisko laboratoryjne, na które składa się: prototypowy układ magazynu energii podłączony do źródła fotowoltaicznego, czujników warunków otoczenia, obciążenia odbiornikiem energii – pracujących w warunkach symulujących off-grid (praca wyspowa) i zarządzanych w architekturze chmurowej IoT poprzez dedykowaną autorską aplikację, w której implementowano wybrane modele predykcji i wykonywano eksperymenty z AI LLM. Aplikację poddano testom mitygacji ryzyka zagrożenia cyberatakami.

Istotność stosowania takich rozwiązań jak to przedstawione w niniejszej pracy jest bardzo ważna w szczególności w kontekście systemów, które mają być zintegrowane z urządzeniami IoT. W szczególności stosowane w infrastrukturze krytycznej, które cechują się większą podatnością na ataki w przypadku stanów poważnego zagrożenia takich jak wojna lub kryzysy. Systemy od których zależy funkcjonowanie danej społeczności, muszą być w permanentnej kontroli bezpieczeństwa, zaś osoby za nie odpowiadające podlegają konkretnym obostrzeniom, jak wskazano w NIS2 [34].

Zapotrzebowanie na nowoczesne systemy oparte o IoT i AI jest duże. Dotyczy to nie tylko do magazynów energii lecz także do innych urządzeń, dla których IoT w pewnym stopniu tworzy warstwę komunikacyjną, i zapewnia automatyzację pozyskiwania danych w czasie rzeczywistym, w tym dla infrastruktury krytycznej. Co za tym idzie wzrasta ryzyko cyberzagrożeń i sparaliżowania funkcjonowania danego obszaru czy społeczności. Aby temu przeciwdziałać należy zachować zasadę minimalizacji zagrożeń w podążaniu za aktualnie stosowanymi trendami technologicznymi, które z biegiem czasu też staną się standardami.

Opracowanie rozwiązania jakie zostało przedstawione w tej pracy ma na celu sprostać zapotrzebowaniu ma autonomiczne zarządzanie magazynem energii w pracy wyspowej (off-grid) z wykorzystaniem układów IoT, jednocześnie inspirując do wdrożenia nowej techniki sterowania urządzeniami fizycznymi opartej na sterowaniu predykcyjnym z wykorzystaniem agentów AI lub chociażby samych w sobie algorytmów LLM.

1.1 Cel i zakres pracy

Celem pracy jest stworzenie bezpiecznej aplikacji do zarządzania elementami infrastruktury krytycznej, opartej na rozwiązaniach IoT i metodach sztucznej inteligencji zaangażowanej do zarządzania tą infrastrukturą.

Zakres pracy obejmuje część praktyczną i część badawczą. W ramach części praktycznej opracowano stanowisko laboratoryjne złożone z prototypu magazynu energii i układu mikrokontrolera IoT oraz aplikacją zarządzającą testowanym układem w czasie rzeczywistym.

W ramach części badawczej, podjęto następujące wyzwania:

- wykrywanie anomalii i ich wpływ na wytrenowane modele, uwzględniając je nie tylko od strony wyników odstających, wynikających z błędnego działania czujników, ale też traktowaniu ich jako ważny sygnał ostrzegawczy w pracy systemu;
- badania poprawności predykcji wartości dla wielkości mierzonych przez sensory układu prototypowego;
- badania skuteczności trenowanych modeli LLM jako agentów do podejmowania decyzji na temat aktualnego procesu ładowania akumulatora, ładowania zewnętrznych urządzeń, pracy lub wyłączenia całego systemu.

System nadzorujący pracę magazynu energii został również poddany badaniom odporności przeciwko możliwym atakom cybernetycznym. Wszystkie pomiary systemu są aktualizowane w czasie rzeczywistym w wykonanej w tym celu aplikacji webowej.

By zrealizować cele badawcze, wykonano implementacje zarówno modeli utworzonych na podstawie sieci neuronowych jak i wybranych współczesnych popularnych modelach językowych (agentów AI do podejmowania decyzji). Tworząc stanowisko laboratoryjne zapewniono warunki do zdalnego odczytywania wszystkich monitorowanych parametrów zbieranych przez system i wpływających na pracę magazynu energii.

1.2. Metodologia badawcza

Zastosowane w pracy metody przeprowadzania badań realizują podejście praktyczno-eksperymentalne, tj. zaplanowano i przeprowadzono eksperymenty na skonstruowanym prototypie magazynu energii. W trakcie eksperymentów zbierano dane, poddano je następnie analizie, interpretacji i prezentacji w aplikacji webowej. Ponadto przeprowadzono przegląd literatury pod kątem zastosowania podobnych rozwiązań w przemyśle, wykonano zapytania o aktualne badania na temat inteligentnych magazynów energii, możliwości wdrożenia AI w energetyce oraz zabezpieczeń cybernetycznych. Poniżej zostały wyodrębnione poszczególne etapy realizacji metody eksperymentalnej jak i teoretycznej oraz zastosowanych narzędzi badawczych.

Metoda eksperymentalna składała się z następujących etapów:

1. zaprojektowanie części elektrycznej systemu: w tym celu użyto akumulatora żelowego 5Ah 12V oraz panelu fotowoltaicznego o mocy 20W, kontrolera do panelu, a także czujników, natężenia prądu(INA219), nasłonecznienia(TEMT6000) , wilgotności, temperatury.(DHT22);
2. zaimplementowanie algorytmów modeli predykcyjnych oraz wdrożenie automatycznego podejmowanie decyzji przez system na podstawie wyników z analizy przeprowadzonych przez modele uczenia maszynowego tak aby wyniki badań mogły być przedstawione w dowolnym momencie i łatwo dostępne dla użytkowników takiego magazynu w tym celu wdrożono aplikację webową napisaną we frameworku Django w części backendowej oraz Stramlite dla części webowej oraz modeli predykcyjnych opartych na algorytmach takich jak: GradientBoost, LinearRegression, LGBM Regressor, jak również skonfigurowano i zaprogramowano mikrokontrolery ESP32 DOIT DEVKIT do zbierania danych

wysyłania decyzji i wystawiania ich do bazy czasu rzeczywistego InfluxDB oraz sterowania systemem na podstawie przetworzonych danych przez modele uczenia maszynowego;

3. badanie odporności systemu na zagrożenia cybernetyczne poprzez przeprowadzenie testów penetracyjne systemu (tj. fazowanie aplikacji, ogólne badania na podatność na ataki typu wstrzykiwanie przez aplikacje symulacje ataków DoS, MITM, próby przejęcia danych z czujników);
4. ocena wyników uzyskanych w zrealizowanych testach.

Na potrzeby badań opracowano stanowisko laboratoryjne, złożone z:

- akumulatory (żelowy 12V 5Ah 1,5A , 18W lub ogniwa litowo-jonowe 3,6V, 3000mAh (tutaj stosowane wraz z przetwornicą step-down i dodatkowym modulem ładowania TP4056);
- zaimplementowana aplikacja Streamlite i Django;
- panele fotowoltaiczne solarne moduły krystaliczne 20W 12V;
- kontroler do paneli solarnych (maks. napięcie wejściowe 12/24V, regulacja PWM);
- czujniki natężenia prądu INA219;
- czujniki natężenia światła TEMA6000, BH1750;
- czujniki wilgotności i temperatury DHT22;
- mikrokontrolery ESP32 (platformy rozwojowe ESP32 DOIT DEVKIT);

jak i oprogramowania oraz bibliotek niezbędnych do opracowania aplikacji obsługującej stanowisko laboratoryjne, w tym:

- Python (Scikit-Learn, TensorFlow, XGBoost);
- Visual Studio Code (z rozszerzeniem PlatformIO);
- framework StreamLit;
- baza danych czasu rzeczywistego InfluxDB oraz baza danych Postgres Neon console;
- Wireshark do analizy ruchu sieciowego i testów bezpieczeństwa.

Przebieg badań zrealizowano w następujący sposób. Początkowo zbierane dane z czujników były wysłane przez mikrokontroler ESP32 do bazy danych InfluxDB, następnie dane te były pobierane adekwatnie do zarejestrowanych szeregów czasowych oraz przetwarzane przez wybrane algorytmy predykcyjne na serwerze Python, który wysłała wyniki swoich obliczeń w osobny punkt pomiarowy do bazy, drugi mikrokontroler, na podstawie wyznaczonych wyników predykcyjnych przez stworzone modele AI, decyduje o włączeniu lub wyłączeniu przekaźnika przełączających stan pracy akumulatora. Kolejnym istotnym ostatecznie w element badań wchodzi też testy systemów poprzez ataki cybernetyczne. Badania wykonane w pracy można podzielić na poszczególne serie testów:

1. Seria I – polegała na kalibracji wszystkich czujników z mikrokontrolerem, napisaniu mechanizmu odczytu danych z czujników oraz wysyłaniu ich do bazy danych czasu rzeczywistego w chmurze.
2. Seria II – składała się z projektowania modeli AI do przewidywania przyszłych wskazań, jednakże modele nie były trenowane na danych historycznych z długiego okresu, które mogłyby już zostać uznane za mało przydatne ze względu na

- zmieniający się klimat, pory roku, warunki atmosferyczne lokalnie zmienne, podjęto decyzję o modelowaniu predykcji wartości mierzonych w krótkim horyzoncie.
3. Seria III – to ogólne testy systemu na, które składały się User Experience aplikacji Webowej, poprawność przesyłanych danych, niezawodność systemu. Analizowano, jak AI podejmuje decyzje o przełączaniu między ładowaniem akumulatora a zasilaniem urządzenia. Mierzono efektywność systemu przy różnych warunkach pogodowych i obciążeniu pracą badanego układu.
 4. Seria IV - zawiera testy cyberbezpieczeństwa wykonane na w pełni gotowej i już działającej aplikacji oraz omówienie wyników testów.

Powyższe serie stanowią chronologiczny przebieg etapów prowadzonych badań. Poszczególne fragmenty rozwiązania problemów jak i wyzwań z jakim zmierzono się podczas w/w serii testów, odpowiadają w skali laboratoryjnej wyzwaniom, z jakimi zmaga się współczesny rynek producentów magazynów energii, czy szerszej systemów stosowanych do SmartHome czy w infrastrukturze krytycznej bazującej na układach IoT, gdzie bezpieczeństwo stanowi warunek dla rozwoju nowoczesnych rozwiązań opartych na sztucznej inteligencji.

1.3 Struktura pracy

Praca została podzielona na część wstępną, w której omawiany jest przegląd i porównanie obecnego stanu techniki, jaka jest zastosowana w podobnych systemach do zarządzania energią w SmartHome. Przeprowadzono ogólne rozeznanie dostępnej literatury oraz wyjaśnienie podstawowych pojęć, ważnych terminologii, jak i zasad, które należy przestrzegać przy konstrukcji systemów bazujących na IoT. W drugiej części pracy znajduje się ocena implementacji uczenia maszynowego na danych przesłanych z czujników oraz mikrokontrolera do bazy danych w chmurze. Przeprowadzono ocenę wdrożonych modeli, wydajności sterowania systemu jak i testów związanych z potencjalnymi atakami cybernetycznymi.

Kończącym etapem w pracy jest dyskusja wyników, które zostały przedstawione w części badawczej pracy oraz porównanie autorskiego podejścia do zaprojektowanego systemu z rozważaniami przedstawionymi w literaturze. Pracę zakończono podsumowaniem wyników przedstawionych badań i perspektywą ich praktycznego zastosowania.

2. Przegląd literatury z zakresu przedmiotu pracy

W tym rozdziale została przeanalizowana dostępna literatura, której kwerenda objęła publikacje ostatnich lat swojej tematyka zaczynając od cyberbezpieczeństwa w aplikacjach do SmartHome, poprzez wskazanie dostępnych nowych technologii z obszaru SmartGrids, po zastosowania rozwiązań opartych na IoT i szczególnym nacisku na obszar infrastruktury krytycznej. Poniższe podrozdziały stanowią opracowanie wybranych przeglądów technologii mających odzwierciedlenie w późniejszym autorskim etapie badań zamieszczonych w niniejszej pracy.

2.1. IoT w infrastrukturze krytycznej – przegląd technologii

Internet of Things (IoT) jest technologią, która umożliwia sterowanie fizycznymi urządzeniami takimi jak np. różnego rodzaju czujniki lub maszyny - przez Internet, co umożliwia zdalną kontrolę nad tymi obiektami fizycznymi oraz komunikację pomiędzy nimi a panelem sterowniczym [14]. Komunikacja użytkownika z układami IoT następuje poprzez dedykowaną stronę internetową lub aplikację mobilną. Cechy charakteryzujące technologię IoT to m.in: sterowanie lub odczyt danych z czujników w czasie rzeczywistym (bez opóźnień), możliwość wysyłania zebranych danych z czujników do bazy oraz możliwość analizowania gromadzonych danych historycznych przez użytkownika, a ponadto dane te mogą zostać wykorzystane do predykcji przyszłych trendów. Kolejną ważną cechą jaką powinien charakteryzować się niezawodny system działający w oparciu o technologię IoT jest zapewnienie bezpieczeństwa i prywatności, w szczególności gdy IoT jest stosowane w infrastrukturze krytycznej[11].

Popularne platformy chmurowe, które stosowane są w rozwiązaniach IoT to m.in:

- AWS IoT - platforma chmurowa stworzona przez Amazon Web Services, która pozwala na zarządzanie urządzeniami IoT w czasie rzeczywistym AWS IoT obsługuje przesyłanie dużych ilości danych, ich analizę i integrację z chmurą, co jest często stosowane w energetyce, produkcji i transporcie[35] ;
- Microsoft Azure IoT-platforma chmurowa do integracji z systemami IoT oraz wystawianiu ich m.in sterowaniu podzespołami elektroniki skojarzonej z IoT oraz zbieraniu danych, analizie zebranych danych umożliwia także integrację z innymi systemami i aplikacjami , które działają w chmurze[35];
- IBM Watson IoT – platforma Watson IoT od IBM wspiera analizę danych i zarządzanie urządzeniami IoT, z naciskiem na sztuczną inteligencję;
- Google Cloud IoT – platforma Google obsługująca urządzenia IoT, zapewniająca narzędzia do analizy i przetwarzania danych w chmurze.[35]

Natomiast sprzętowe zintegrowane mikrokontrolery, które stanowią sprzęt bazowy do budowy takich systemów IoT to m.in: mikrokontrolery ESP32, ESP8266, Raspberry Pi, Orange Pi [36] oraz protokoły komunikacyjne takie jak: Zigbee, MQTT czy WebSocket. [23]

Infrastruktura krytyczna to określenie, które wynika z roli pełnionej przez dany system lub układ współpracujących urządzeń dla społeczeństwa. Infrastrukturę krytyczną stanowi zespół złożony z zasobów obejmujących systemy oraz obiekty będące fundament

funkcjonowania społeczeństwa oraz gospodarki i zarządzanie nimi [23]. Infrastruktura krytyczna obejmuje takie sektory m.in jak energetyka, wodociągi, transport, telekomunikacja.

Zastosowanie IoT w infrastrukturze krytycznej można zobrazować następującymi przykładami, które są podzielone względem sektorów jakich mogą dotyczyć:

- sektora energetyki- przykładem zastosowania IoT w energetyce jest możliwość monitorowania oraz zarządzaniem wykorzystanej energii w czasie rzeczywistym, a także przewidywanie jej przyszłych wartości, co może przekładać się na wykluczenie błędów lub prób nielegalnego poboru tej energii [9]. Opisywane wykorzystanie IoT w tym przykładzie znane jest jako Smart Grids. sektor energetyki odnawialnej działający w oparciu o systemy IoT może być również wykorzystywany bardziej wydajnie gdyż sieć dzięki pomiarom zgromadzonym z inteligentnych systemów opartych na IoT może optymalizować oraz umożliwiając automatyczne dostosowywanie produkcji energii do zapotrzebowania.[24]
- sektora wodociągów - systemy IoT również jak i w sektorze energetycznym mogą usprawniać zarządzanie strukturą wodociągów m.in monitorować przepływ wody i wykrywać wycieki, co pozwala na oszczędność zasobów oraz zmniejszenie kosztów eksploatacji, jak i również mogą zostać wykorzystywane celem monitorowania jakości wody (pH, zanieczyszczeń) w czasie rzeczywistym, co pozwala na szybsze wykrywanie problemów i natychmiastowe reakcje.[25]
- sektora transportu - IoT umożliwia zbieranie danych z czujników zamontowanych w pojazdach, co pozwala na monitorowanie ruchu, optymalizację tras oraz przewidywanie korków [26].
- sektora telekomunikacji- gdzie IoT może być wykorzystywane do monitorowania stanu urządzeń i infrastruktury, co pozwala na przewidywanie awarii oraz optymalizację działania sieci [26].
- sektor bezpieczeństwa publicznego – systemy IoT mogą zostać wykorzystywane w ramach inteligentnego monitoringu miejsc publicznych co zwiększa bezpieczeństwo kontroli służb ochrony nad obywatelami [45]

Z punktu widzenia przedmiotu badań, tu IoT w obszarze energetyki, przykładami infrastruktury krytycznej mogą być SmartGrids i SmartHome.

SmartGrids definiowane są jako inteligentne sieci energetyczne, w praktyce jednak chodzi o szerzej pojęte systemy, które wykorzystują technologię komunikacyjną i informacyjną, w tym również IoT do zarządzania dostaw energii elektrycznej w sposób mający na celu jej większą optymalizację [26]. Cechy Smart Grids to m.in; możliwość zdalnego monitorowania i zarządzania, interaktywność w trakcie trwania monitoringu co pozwala na zarządzanie poborem energii w czasie rzeczywistym, automatyzacja procesów co ma na celu zredukowanie awarii.[27]

Na system SmartGrids składają się różne urządzenia współpracujące ze sobą tak aby stworzyć pełen funkcjonalny, elastyczny i bezpieczny system do dystrybucji energii zarówno dla producentów jak i konsumentów będących niekiedy też prosumentami. W tabeli 1 zostały wyszczególnione ogólne założenia jakie muszą spełniać komponenty wchodzące w skład systemu SmartGrids [27].

Tabela 1. Komponenty w IoT[27]

Komponent	opis	główne funkcje	Przykłady komponentów
Smart Meters	inteligentne liczniki czyli urządzenia, które monitorują zużycie energii w czasie rzeczywistym i przesyłają te dane do dostawcy energii oraz użytkowników.	Umożliwiają zdalne odczyty, bieżące monitorowanie zużycia energii, a także dynamiczne taryfikowanie w zależności od pory dnia i godziny zapotrzebowania.	Itron - Model Itron OpenWay Riva. Landis+Gyr - Model E350. Sensus - Model FlexNet.
sensory pomiarowe	różne czujniki monitorujące parametry sieci, takie jak napięcie, prąd, przepustowość i jakość energii.	pozwalają na wczesne wykrywanie problemów, takich jak przeciążenia czy awarie	Schneider Electric - czujniki jakości energii, takie jak Model PowerLogic ION9000. Siemens - czujniki pomiarowe w systemie Spectral Power.
Energy Management System	Oprogramowanie do analizy danych z inteligentnych liczników i czujników	Optymalizuje działanie systemu energetycznego, umożliwia prognozowanie zapotrzebowania oraz zarządzanie generacją energii.[10]	Siemens - System EnergyIP. Schneider Electric - EcoStruxure Energy Management. General Electric - Digital Energy Solutions.[10]
magazyny energii	Systemy, które przechowują energię (np. akumulatory) w celu jej późniejszego wykorzystania.	Pomagają w stabilizacji sieci, umożliwiają magazynowanie energii z źródeł odnawialnych oraz zarządzanie szczytami zapotrzebowania[10]	Tesla - Tesla Powerpack i Powerwall. LG Chem - LG RESU (Residential Energy Storage Unit). Sonnen - SonnenBatterie.
odnawialne źródła energii	Technologie wykorzystujące energię odnawialną, takie jak panele fotowoltaiczne i turbiny wiatrowe.	Dostarczają zasilanie do sieci oraz integrują się z systemem zarządzania energią.	First Solar - panele słoneczne. Vestas - turbiny wiatrowe. Siemens Gamesa - turbiny wiatrowe (onshore i offshore)

Smart Home jest to technologia, która wykorzystuje Internet Rzeczy (IoT) umożliwiając zdalne sterowanie urządzeniami i systemami znajdującymi się w budynku, aby zwiększyć komfort, wygodę użytkownika i efektywność energetyczną dla jego użytkowników.

Poniższa tabela 2 stanowi zestawienie kluczowych komponentów jakie mogą wchodzić w podstawowy system SmartHome.

Tabela 2. Komponenty w SmartHome [37, 38, 39]

komponent	opis	główne funkcje	przykłady
inteligentne oświetlenie	zdalnie sterowane systemy oświetleniowe, sterowane za pomocą aplikacji webowych i mobilnych [37]	Możliwość zdalnego włączania i wyłączania światła za pomocą aplikacji mobilnej lub poleceń głosowych. Umożliwiają regulację jasności i kolorów (w przypadku żarówek RGB). Programowanie harmonogramów oświetlenia (np. włączanie/wyłączanie w określonych godzinach). Integracja z systemami bezpieczeństwa (np. symulacja obecności przez automatyczne włączanie światła).[37]	Philips Hue - inteligentne żarówki i systemy oświetleniowe, które można kontrolować za pomocą aplikacji lub głosowo. LIFX - inteligentne żarówki LED z możliwością zmiany kolorów i jasności. Sengled - inteligentne żarówki, które mogą działać jako czujniki ruchu.[37]
inteligentne termostaty [38]	Urządzenia do zdalnego zarządzania temperaturą w budynku, które uczą się preferencji użytkownika [38]	Automatyczne dostosowywanie temperatury w zależności od pory dnia i obecności użytkowników. Możliwość zdalnego sterowania temperaturą za pomocą aplikacji mobilnej. Funkcje prognozowania pogody, które wpływają na ustawienia grzania/chłodzenia [38]	Nest (Google) - inteligentny termostat, który uczy się preferencji użytkownika i optymalizuje zużycie energii. Ecobee - inteligentny termostat z czujnikami obecności, który może zarządzać temperaturą w różnych pomieszczeniach. Honeywell Home - różne modele termostatów do zarządzania ogrzewaniem i chłodzeniem.[38]
systemy zabezpieczeń	Systemy monitorujące, które chronią dom przed włamaniami i	Zdalne powiadomienia o wykrytych ruchach, otwarciu drzwi czy rozpoznaniu twarzy. Wideo na żywo i	Ring - inteligentne dzwonki wideo i systemy monitoringu, które pozwalają na zdalne obserwowanie posesji.

	innymi zagrożeniami[39]	nagrywanie, które można przeglądać zdalnie.[39]	Arlo - kamery monitorujące, które oferują funkcje detekcji ruchu i nagrywania wideo. SimpliSafe - systemy alarmowe i monitorujące, które można konfigurować i kontrolować zdalnie.[39]
--	-------------------------	---	---

Oprócz wymienionych wyżej komponentów wchodzących w skład systemu inteligentnych domów wyróżnia się jeszcze oprogramowania najczęściej działające w chmurze do odczytu danych przez urządzenia pomiarowe oraz wizualizacji zebranych informacji a także jako interfejs sterowania, przykładem takich programów mogą być[11]:

- platformy do zarządzania podzespołami urządzeń IoT w domu takie jak **Apple Home Kit**;
- usługi, które pozwalają tworzyć proste automatyzacje między różnymi aplikacjami oraz urządzeniami w celu kontroli różnych procesów, przykładem tak funkcjonującej usługi jest IFTTT z ang. If This Then That;
- aplikacje personalizowane pod firmę producenta takich systemów, jak Samsung SmartThings, czyli dedykowany przez firmę Samsung hub do integracji różnych urządzeń i automatyzacji procesów w domu.

2.2. Zbieranie danych w systemach IoT oraz ich transfer

W artykule opublikowanym przez blog e-magazyny na temat smartmeteringu [40] opisane zostały systemy inteligentnego opomiarowania, które są kluczowym elementem infrastruktury IoT w energetyce do, których m.in należą:

- inteligentne liczniki energii - słowo "inteligente" ma znaczenie w kontekście automatycznego odczytu zużycia energii oraz monitorowania parametrów sieci w czasie rzeczywistym;
- aktualnie stosowane liczniki w przemysłowych wariantach jak i też tych dla gospodarstwa domowego zestawiono w poniższej tabeli nr 3.

Tabela 3. Aktualne liczniki w przemysłowych wariantach dla gospodarstw domowych[40]

zastosowanie	model	technologia transmisji
Gospodarstwa domowe	Landis+Gyr E450	PLC, GSM
Gospodarstwa domowe	Kamstrup OMNIPOWER	LoRaWAN, NB-IoT
Przemysł	Siemens PAC2200	Modbus TCP/IP
Przemysł	ABB A44 212-100	GSM, światłowody
OZE / PV	Elster A1700	Smart Grid, SCADA

- infrastruktura komunikacyjna czyli dwukierunkowa transmisja danych między licznikiem a systemem zarządzania, co jest ważne dla szybkiego transferu danych w integracji energetyki z IoT;
- oprogramowanie do zarządzania danymi, którego celem jest gromadzenie danych, analiza, przetwarzanie oraz ewentualna wizualizacja pozwalająca na optymalizację zużycia energii oraz wczesnego wykrywania awarii.

Z wcześniej omawianego artykułu na blogu [40] można wyodrębnić następujące metody komunikacji liczników energii z systemami zarządzania są to:

- Power Line Communication (PLC) – umożliwiające przesył danych przez linie elektroenergetyczne;
- Sieci bezprzewodowe-GSM/4G/5G, ZigBee, LoRaWAN;
- Moduły IoT z Wi-Fi – dla lokalnych zastosowań Smart Home.

Zalety wyżej przedstawionych rozwiązań to m.in: eliminacja ręcznego odczytu, możliwość wczesnego wykrywania awarii, możliwość zdalnego odcięcia i przywrócenia dostaw energii elektrycznej.[40]

Infrastruktura komunikacyjna jest realizowana poprzez systemy SCADA, Data Concentrator Unit, DCU) z tym, że niektóre z systemów używane są do analizy danych m.in SCADA, pozostałe to różnego rodzaju chmury obliczeniowe i Big Data (dane są przetwarzane w chmurze (np. Azure IoT, AWS IoT)), oraz AI i Machine Learning (algorytmy uczące się analizują dane i przewidują zużycie).

Kolejne publikacje [4, 10 i 41] pokazują nowoczesne metody pracy z danymi oraz ich przetwarzania, co jest istotne do prawidłowego działania nowoczesnych systemów inteligentnych. Do tych metod m.in należą: Edge Computing, Cloud Computing, Big Data AI, a różnice między nimi głównie polegają na przesyłaniu danych. Edge Computing wyróżnia się tym, że dane są przetwarzane lokalnie na urządzeniu IoT (np. w liczniku energii), co zmniejsza opóźnienia. Cloud Computing charakteryzuje się natomiast tym, że dane przesyłane do chmury, gdzie są analizowane i przechowywane. Big Data & AI analiza dużych zbiorów danych pozwala na wykrywanie anomalii i przewidywanie zużycia energii.[4] W publikacji „*Nowoczesne metody magazynowania energii*” [10] nie brakuje również wymienionych rozwiązań jakie stosuje się w SmartHome do zabezpieczenia danych podczas transferu. Dla zapewnienia prywatności danych, wyodrębnia się główne metody[41]:

- szyfrowanie danych – AES-256, TLS/SSL chronią przesyłane informacje;
- autoryzacji dostępu – stosowanie kluczy kryptograficznych do uwierzytelniania urządzeń IoT;
- monitorowaniu sieci – wykrywanie anomalii i nieautoryzowanych działań.

2.3. Dostępne metody predykcyjnego sterowania systemami IoT

Predykcja danych polega na przewidywaniu wskazań mierników i czujników za pomocą przeróżnych algorytmów AI, których skuteczność i dopasowanie zostaną również poddane jako obiekt badań niniejszej pracy. Do popularnych metod sterowania poprzez wyuczone algorytmy AI, które stosowane są w systemach IoT zalicza się:

- model Predictive Control czyli model predykcyjnego sterowania, jest to podejście, w którym wykorzystuje się matematyczny model systemu celem prognozowania jego przyszłych stanów i wyznaczania jego optymalnych decyzji sterujących w określonym horyzoncie czasowym, tego typu podejście jest często stosowane w zarządzaniu rozładowaniem i naładowaniu np. magazynu energii.

W kolejnej publikacji omówiono zastosowanie MPC do sterowania magazynami energii z uwzględnieniem prognoz zużycia i generacji z OZE. Metoda ta zapewnia optymalne zarządzanie energią i ogranicza straty.[5]

- sterowanie rozmyte gdy w danym obszarze danych trudno jest stworzyć na podstawie danych jednoznaczny model matematyczny stosowane są systemy rozmyte czyli wykorzystuje się w takich przypadkach logikę rozmytą do podejmowania decyzji w sytuacjach niepewności. Logika rozmyta to technika sterowania i podejmowania decyzji, która naśladuje ludzki sposób myślenia – nie działa w trybie 0 lub 1, tylko mówi: „trochę”, „średnio”, „bardzo”.

Sterowanie rozmyte w literaturze zostało przedstawione jako tzw. fuzzy logic, technika, która była wykorzystywana była m.in. do sterowania HVAC (ogrzewanie, wentylacja, klimatyzacja) oraz do integracji z OZE, np. w systemach Smart Home.

- reinforcement learning (uczenie przez wzmocnianie) - to popularne rozwiązanie dla systemów IoT bardzo złożonych polega ono na tym, że system sam uczy się optymalnych decyzji przez próby i błędy inaczej ujmując sterownik „nagradza się” za dobre decyzje)[5];
- Kalman Filter jest to metoda klasyfikowana do tzw. metod estymacji stanu, które są wykorzystywane wszędzie tam gdzie nie da się bardzo dokładnie (w danej chwili, lub bardzo krótkim przedziale czasowym) dokładnie zmierzyć wielkości mierzonych i potrzebne jest ich oszacowanie czyli estymacja. Takie podejście w budowlach magazynu energii mogłoby mieć bardzo kluczowe znaczenie gdyż są parametry, które nie da się z odpowiednio dużą dokładnością zmierzyć (choćby rzeczywisty poziom naładowania baterii mając jedynie mierzony prąd i napięcie) co zostało pokazane również w różnych publikacjach w literaturze m.in. w publikacji Pletta z 2004 roku w, której autor pokazuje zastosowanie EKF do precyzyjnej estymacji SOC i SOH (state-of-health) akumulatorów w pojazdach elektrycznych [13].
- Neural Network Predictive Control jest wykorzystaniem sieci neuronowych zamiast klasycznego modelu predykcyjnego opartego na modelowaniu matematycznym.[5]

Jednym z przykładów literatury, który omawia zastosowanie AI w systemach energii jest publikacja w, której zaprezentowano zastosowanie algorytmów uczenia maszynowego (w tym sieci neuronowe) do optymalizacji sterowania mikro siecią i zarządzania magazynami energii w oparciu o dynamiczne warunki pogodowe i zużycie energii[8].

- Metody adaptacyjno -hybrydowe, metody takie stanowią połączenie wszystkich wyżej wymienionych metod celem osiągnięcia najbardziej odpowiedniejszej optymalizacji. Często znajdują zastosowanie w rzeczywistych aplikacjach np. połączenie fuzzy logic z MPC, lub MPC oraz AI, by połączyć intuicję sterowania z dokładnością predykcji. Przykładem może być mikro sieć z baterią, fotowoltaiką i wiatrakami, używając fuzzy logic do dostosowania parametrów

optymalizacji, w literaturze takie sterowanie zostało przedstawione w publikacji z 2016 roku autorstwa Aghajani, G. R., & Kalantar, M. *"Optimal energy management of a smart microgrid with hybrid renewable energy systems using new fuzzy adaptive modified particle swarm optimization."* przez wydawnictwo Energy.[7]

Powyższe metody nie są tylko przykładem metod, które mają zastosowanie w magazynach energii ale także mają zastosowanie m.in w stacjach ładowania pojazdów EV czy innych systemach zarządzania energią.

2.4 Współczesne cyberzagrożenia systemów IoT

W tej części rozdziału zostały opisane standardowe zagrożenia z jakimi systemy IoT najczęściej są narażone oraz najpopularniejsze typy ataków wykorzystywane przez hackerów celem przejęcia kontroli lub wyłudzenia informacji zbieranych przez urządzenia i oprogramowanie oparte na IoT.[5]

Do najbardziej znanych metod ataków na jakie mogą być zagrożone systemy IoT należą:

- ataki typu DDoS (z ang. Distributed Denial of Service) ataki tego typu polegają na przeciążeniu urządzenia lub sieci IoT dużą ilością fałszywych żądań w wyniku czego sieć nie odpowiada w sposób właściwy na rzeczywiste żądania pochodzące od użytkownika pierwotnego;[15]
- przejmowanie urządzeń;
- ataki typu Man in the middle w tego typu atakach haker przechwytuje i manipuluje danymi przesyłanymi między urządzeniem IoT a aplikacją zarządzającą [15]
- ataki typu firmware exploit polegające na przejęciu kontroli nad inteligentnymi urządzeniami przez atak na luki w ich firmware;
- przełamywanie słabych haseł i uwierzytelnianie aplikacji stanowiących interfejs do zdalnego sterowania urządzeniami IoT[15]

Wyzwania jakie dotyczą bezpieczeństwa aplikacji IoT wynikają między innymi ze specyfiki urządzeń oraz komponentów użytych w tych systemach takich jak:

- ograniczona moc obliczeniowa urządzeń IoT, urządzenia wykorzystywane jako baza do działania w aplikacjach IoT takie jak mikrokontrolery ESP32 mają zwykle niewielką moc obliczeniową, co utrudnia stosowanie zaawansowanych zabezpieczeń, takich jak szyfrowanie danych [16] ;
- zróżnicowane poziomy ochrony i standardów dla urządzeń IoT, które wynikają z braku wyznaczenia jednolitych standardów dla tej grupy urządzeń w tym zakresie- różne protokoły zabezpieczające w systemach SmartHome takie jak MQTT-S, CoAP[17];
- złożoność zarządzania bezpieczeństwem przy dużej liczby urządzeń wykorzystywanych w systemach IoT[16];
- przesyłanie danych do chmury zwiększa ryzyko związane z ich bezpieczeństwem oraz prywatnością[17].

Ponadto oprócz wyżej przytoczonych popularnych ataków i zagrożeń w literaturze jest wiele publikacji w których omawiane są zagrożenia w kontekście spersonalizowanym co do systemu IoT oraz rodzaju infrastruktury do, której ten system został przygotowany [14].

Poniżej znajdują się publikacje, które reprezentują opisy badań lub rozprawy na temat wdrażania systemów IoT w infrastrukturę krytyczną oraz dokładnie przytoczone najważniejsze wnioski i wyniki z badań na temat bezpieczeństwa;

Pierwszym artykułem, który zostanie przedstawiony jest *"Zagrożenia wynikające z implementacji koncepcji Internetu rzeczy w działalności przedsiębiorstw"* autorzy Artur Rot i Bartosz Blaike wskazują, że w 8 na 10 badanych urządzeń IoT wykryto podatności związane z gromadzeniem danych osobowych, takich jak imię i nazwisko, e-mail, adres czy informacje o stanie zdrowia [20]. Również z tego samego artykułu wynika, że 80% urządzeń IoT nie wymagało odpowiedniej długości i specyfiki haseł zakładanych do kontroli systemów IoT co umożliwiało używanie trywialnych haseł, zwiększając ryzyko nieautoryzowanego dostępu. Autorzy zwracają również uwagę na niezabezpieczone interfejsy WWW w urządzeniach IoT, co stanowi istotne zagrożenie dla bezpieczeństwa systemów.[20]

W publikacji z Uniwersytetu Ekonomicznego we Wrocławiu pod tytułem *„Bezpieczeństwo Internetu rzeczy. Wybrane zagrożenia i sposoby zabezpieczeń na przykładzie systemów produkcyjnych”* z 2017 autorstwa Artura Rota[42] pokazano dokładnie jak luki i błędy w oprogramowaniu urządzeń IoT mogą być wykorzystywane przez cyberprzestępców do przeprowadzania ataków na systemy produkcyjne. Z wyżej przytoczonych przykładów pozycji dostępnych w literaturze wynikają też propozycje ich rozwiązań. Celem przeciwdziałania powyższym zagrożeniom, proponowane są następujące rozwiązania[20]:

- Wzmocnienie mechanizmów autoryzacji i uwierzytelnienia rozwiązane poprzez wprowadzenie silniejszych haseł oraz dwuskładnikowego uwierzytelniania w urządzeniach IoT.
- Szyfrowanie transmisji danych poprzez zapewnienie poufności i integralności danych przesyłanych między urządzeniami IoT poprzez stosowanie protokołów szyfrujących.
- Regularne aktualizacje oprogramowania poprzez usuwanie luk i błędów w oprogramowaniu i przeprowadzanie regularnych aktualizacji , co minimalizuje ryzyko wykorzystania znanych podatności
- Zabezpieczanie interfejsów użytkownika: stosowanie bezpiecznych praktyk programistycznych w celu ochrony interfejsów WWW przed potencjalnymi atakami.

Analiza literatury z tej dziedziny wykazuje, że mimo dostępnych rozwiązań ich implementacja w rzeczywistości jest niewystarczająca lub nie sprawdza się dostatecznie dobrze, wiele producentów urządzeń IoT nie zwraca też uwagi na kwestie zabezpieczenia tego typu systemów i aplikacji, zazwyczaj ograniczają się tylko do podstawowej częściowej ochrony. [18]

2.5 Sposoby zastosowania AI w inteligentnym sterowaniu i ochronie systemów IoT przed atakami cybernetycznymi

Zintegrowanie zaawansowanej sztucznej inteligencji z systemami struktury Internetu rzeczy przed atakami cybernetycznymi jest w obecnym czasie bardzo wysokim obszarem zainteresowań, które potwierdzają najnowsze publikacje naukowe jak i artykuły opisujące różnego rodzaju przedsięwzięcia w realizacji rozbudowanych technik wdrażania m.in algorytmów uczenia maszynowego w celu identyfikacji i neutralizacji zagrożeń w sieciach IoT, poddawane zostały różne techniki ML, które w przyszłości mogą znaleźć zastosowanie m.in w wykrywaniu anomalii oraz ochrony przed atakami.

W publikacji z 2020 roku pt. Machine Learning Based Solutions for Security of Internet of Things (IoT) wskazano konieczność opracowania bardziej efektywnych i lekkich modeli ML, które mogłyby działać na urządzeniach o ograniczonych zasobach m.in chodzi tutaj o urządzenia charakterystyczne dla obszaru IoT czyli mikrokontrolery MCU, czujniki IoT, kamery niskiej mocy itp. [19]

Wynikiem badań na temat stworzenia bardziej adekwatnych rozwiązań przedstawionych w niniejszym opracowaniu jest to, że większość klasycznych algorytmów do detekcji anomalii jest zbyt ciężka dla urządzeń IoT i musi zostać poddana optymalizacji, przykładowe rozwiązania zaprojektowane w tym celu dotyczą m.in uproszczonych drzew decyzyjnych w sieciach neuronowych (TinyML) oraz postawienie na Edge AI, czyli podejście do przeprowadzania analizy danych lokalnie na urządzeniu bez wysyłania danych do chmury.[12]

Istnieją specjalistyczne algorytmy do uczenia ML dedykowane dla mikrokontrolerów, przykładem jest TensorFlow Lite for Microcontrollers inaczej też funkcjonująca pod nazwą TFLite Micro[2]], kolejna pozycja to Simple Neural Networks, ale tylko 1 lub 2 warstwowy, algorytm często używany w TinyML.[2]

Przykładem zintegrowania jednego z takich algorytmów mianowicie modelu uproszczonego sieci neuronowej z TFLite Micro jest wykrywanie anomalii w sieci czujników IoT. Opisany case-study został przedstawiony następująco:

- urządzenia, które zostały wykorzystane do zaimplementowania takiego systemu to: płytki rozwojowe ESP32 (popularnie dostępny model DOIT DEVKIT, CZUJNIK DO POMIARU TEMPERATURY DHT22 oraz akcelerometr;
- zadaniem tego systemu miało być wykrywanie nietypowego wzrostu temperatury i drgań wynikających z ataku lub awarii, natomiast efekt takiej implementacji miał skupiać się głównie na odpowiednio szybkiej reakcji czyli np wysyłka danych do brokera MQTT oraz włączenie sygnału ostrzegawczego poprzez zapalenie diody LED;

w tabeli nr 4 znajduje się zestawienie najbardziej częstych algorytmów, które znajdują zastosowanie w urządzeniach o ograniczonych zasobach pamięciowych i mocy

Tabela 4. Algorytmy dla urządzeń o ograniczonym zasobie[3]

Algorytm	Dokładność	RAM/Flash	czas reakcji	złożoność implementacji
Decision Tree	70 %	niska 2 do 10 KB	szybki dla średniej ilości próbek *	stosunkowo prosta w porównaniu z innymi **
K-Nearest Neighbors (KNN)	60-85%	średnia i bardzo zależna od danych	wolny w porównaniu do pozostałych, dla średniej ilości próbek*	średnia**
Naive Bayes	60-85%	Niska (~5–20 KB)	szybki *	prosta **
SVM z ang. Support Vector Machine	80–95%	Wysoka (>40 KB modelu)	Wolniejszy*	Trudna **
Algorytm	Dokładność	RAM/Flash	czas reakcji	złożoność implementacji

* szybkość została ujęta w sposób miary jakościowej jako szybki, średni i wolny, dokładne wartości dla szybki :5 – 15 ms, dla średnio:15 – 50 ms, wolny: 50 – 200+ ms;

** miary te zostały ujęte w sposób jakościowy, interpretacja ilościową składa się cały proces implementacji w tym wzięte zostały pod uwagę aspekty takie jak: istnienie gotowych już bibliotek, wymagania wsparcia sprzętowe, trudność wytrenowania danego modelu i przeniesienie go na MCU.

Trenowanie modeli, które później są implementowane w mikrokontrolerach może być wykorzystane w gotowych platformach do trenowania modeli, większość z tych platform zawiera już funkcje eksportu wytrenowanego modelu do pliku w formacie cpp(dla języka C++), który można bezpośrednio już wgrać do mikrokontrolera. Dostępne tego rodzaju platformy to m.in:

- Edge Impulse - jest to chmurowa platforma stworzona do tworzenia modeli ML na mikrokontrolery, platforma ta została stworzona w 2019 roku przez Zacha Shelby (znany z Arm i IoT) i Jana Jongboom i funkcjonująca oraz stale rozwijana do dziś. Edge Impulse bardzo szybko zdobyło popularność wśród hobbystów, firm i uczelni pracujących z urządzeniami typu Arduino, ESP32, STM32 itp.[43]
- Teachable Machine od Google - Teachable Machine swój początek miało w 2019 roku i pojawiło się wtedy wówczas jako wersja próbna, a właściwie eksperymentalna, natomiast już w 2020 roku już oficjalnie wyszło. Platforma ta skupia się głównie na przystępnym i intuicyjnym interfejsie dla użytkowników, często jest wykorzystywana dla edukacji i szybkiego tworzenia modeli ML bez kodowania np. tworzenie modelu ML z dostarczonych zdjęć celem ich klasyfikacji według podanych kryteriów [1].

Na zrzucie ekranu widocznym na rys. 1 przedstawione zostały tzw. developments kits, które są wspierane przez Edge Impulse, na te pozycje można wygenerować i wgrać firmware przygotowany przez Edge Impulse, a przed wgraniem oprogramowania można też przetestować modele ML na żywo.[43]

EDGE AI HARDWARE

Arducam Pico4ML TinyML Dev Kit

Arduino Nano 33 BLE Sense

Arduino Nicla Sense ME

Arduino Nicla Vision

Arduino Portenta H7

Blues Wireless Swan

Espressif ESP-EYE

Himax WE-I Plus

Infineon CY8CKIT-
062-BLE Pioneer KitInfineon CY8CKIT-
062S2 Pioneer Kit **Powered by GitBook**

Rysunek 1. Zestaw wspieranych mikrokontrolerów przez Edge Impulse[43]

W przypadku zastosowania algorytmów ML celem ochrony systemów IoT przed atakami cybernetycznymi w literaturze istnieje kilka kluczowych pozycji [15] prezentujących działanie różnych metod i technik, należy jednak zaznaczyć że żadna z tych metod, która opiera się na ML nie zastąpi tradycyjnych zabezpieczeń takich jak: jak firewalle, antywirusy czy szyfrowanie danych, natomiast modele AI wytworzone w tym celu mogą jedynie wzmocnić ten poziom bezpieczeństwa i umożliwić odpowiednio wcześniejsze reakcje na wstępne anomalie zachodzące w właśnie tego typu systemach opartych na gromadzeniu i przetwarzaniu dużych zbiorów danych.[3]

Wspomniane już wcześniej pozycje literaturowe łączące zastosowanie AI w bezpieczeństwie systemów IoT to m.in: publikacja z 2021 roku autorstwa Iqbal H. Sarker pod tytułem “*Cyber Learning: Effectiveness Analysis of Machine Learning Security Modeling to Detect Cyber-Anomalies and Multi-Attacks*”. [4] W części badawczej pracy autor skupił się głównie na przeprowadzeniu eksperymentów na dużych zbiorach danych dotyczącym bezpieczeństwa takich jak UNSW-NB15(czyli nazwa zbioru danych stworzony przez Australian Centre for Cyber Security (ACCS) w 2015, który zawiera ponad 100 cech (features) opisujących pakiety sieciowe, oraz etykiety typu: lub jeden z popularnych ataków takich jak shellcode lub worms) i NSL-KDD(podobny zbiór danych do poprzedniego o zbliżonej tematyce struktury) aby ocenić skuteczność trenowanych przez siebie modeli ML oraz ich potencjał do wykrywania anomalii w tego typu danych jak i wskazań, które świadczyłyby o zajściu ataku.

Badania wykazały następujące fakty: głównie to, że ML radzi sobie lepiej niż reguły statyczne w wykrywaniu nowych tzw. zero-day ataków, omawiane zero-day ataki to jedne z najgroźniejszych ataków na systemy IoT, polegają one na tym, że osoba hackuje system jest pierwsza osoba, która znalazła lukę w systemie IoT czyli po prostu jakiś błąd ze strony producenta po stronie uwierzytelniania i wykorzystuje to zanim producent rozpozna to na etapie aktualizowania firmware, często jest tak, że producent dowiaduje się dopiero swoim błędzie po fakcie. [4]

Kolejnym punktem badawczym w publikacji Sarkera była badanie kombinacji technik ML czyli np. SVM, decision tree czy deep learningu. Wynik tego obszaru w badaniu był dość intuicyjny ponieważ dowiódł, że kombinacja technik ML i ich wzajemna kumulacja daje lepszy efekt niż tylko pojedynczy algorytm ML, w tym badaniu skumulowane ze sobą następujące algorytmy :SVM, decision tree oraz deep learning. [4]

Wnioski powyższych badań dowiodły również, że w przyszłości będzie coraz większe zapotrzebowanie na platformy typu edge Impulse, oraz ogólnej technologii podobnych platform skupiających się na edge computingu czyli uczeniu modeli lokalnie na danym mikrokontrolerze o ograniczonych zasobach niż uczeniu ich globalnie wysyłając dane do chmury obliczeniowej, takie zapotrzebowanie bierze się głównie z tzw. "lightweight models" czyli nastawieniu do tworzenia modeli AI żeby sam proces ich tworzenia był dosyć szybki i lekki co sprzyja energooszczędności, jeśli brana byłaby pod uwagę struktura IoT w mikrokontrolerach. [4]

Kolejną pozycją w literaturze jest publikacja, która ukazała się w 2020 roku autorstwa Antoine Delplace, Sheryl Hermoso i Kristofer Anandita pt. "*Cyber Attack Detection thanks to Machine Learning Algorithms*". W tej przytoczonej publikacji badaniu zostały poddane pięć typów algorytmów ML zastosowanych w kontekście wykrywania botnetów, badano je głównie pod względem osiąganego skuteczności w wykrywaniu botnetów jak i klasyfikacji złośliwego ruchu w sieci [28].

Natomiast w artykule dotyczącym roli uczenia maszynowego w cyberbezpieczeństwie z 2022 roku autorstwa Giovannia Apruzzese [29] oraz innych współautorów tej pozycji można przytoczyć najważniejsze z wymienionych zastosowań ML w dziedzinie cybersecurity, głównie skupiających się na automatycznym wykrywaniu anomalii, które realizowane są poprzez modele ML, uczące się ruchu sieciowego, który działa w sposób poprawny to znaczy nie ma w nim wartości wskazujących na anomalie czy redundancje. Modele takie oblicza stałe wychylenia między danymi mierzonymi aktualnie, a historycznymi "dobrymi" danymi i na podstawie zidentyfikowanych zbyt dużych odchyłań od normy identyfikowane są próby włamania. Innym prostym przykładem zastosowania takiego podejścia jest analiza z czujników IoT i identyfikacja tego, że np. jeden z nich zaczyna przysyłać dane z nietypową częstotliwością. W omawianej w tym podrozdziale literaturze zaprezentowano również podejście do klasyfikacji złośliwego oprogramowania (malware), oparte na analizie sekwencji bajtów oraz zachowań systemowych, pozwalające na skuteczne rozpoznanie i przypisanie danego programu do kategorii szkodliwych. Podobne podejście zastosowano także w klasyfikacji aplikacji, gdzie zamiast cech niskopoziomowych wykorzystano dane dotyczące przydzielonych uprawnień oraz wykonywanych działań, co umożliwiło podział aplikacji na bezpieczne i potencjalnie niebezpieczne.[29]

3. Zdalne sposoby przesyłania danych i ich przetwarzania

W poprzednim rozdziale zostały omówione sposoby lokalnego tworzenia modeli, które są wgrywane lokalnie w firmware, natomiast takie podejście niesie ze sobą pewne ograniczenia m.in to, że przy tworzeniu takich modeli zazwyczaj jest ograniczona moc obliczeniowa i pamięć, przez co wyraźnie trudniej jest zaimplementować zaawansowane modele takie jak np. sieci głębokie oraz RNN.

W tym rozdziale został szczegółowo omówiony sposób analizy i przetwarzania danych w chmurze, która została połączona z mikrokontrolerem przez WiFi. Takie rozwiązanie jest rozwiązaniem, które nie jest ograniczane poprzez możliwości lokalne samego urządzenia. Ponadto dane, które zostają przechowywane w tzw. cloud database są dostępne dla użytkowników, co czyni takie podejście bardziej kontrolowanym pod względem procesów jakie w tych danych zachodzą, a także daje możliwość w późniejszym czasie dalszego rozwinięcia aplikacji na innych obszarach [30].

Istnieją dwa główne rozwiązania sterowania, w których wykorzystuje się modele AI. Do tych rozwiązań należą podejścia TinyML, które ma ograniczenia zasobowe ale jest bardziej niezawodne w przypadku braku dostępu do Internetu oraz podejście Cloud AI skupiające się na przechowywaniu modeli w chmurze lub w lokalnej bazie danych, w podejściu Cloud AI dane są przesyłane do chmury obliczeniowej, gdzie są przetwarzane za pomocą dużych modeli uczenia maszynowego. Niektóre systemy wykorzystują oba podejścia jednocześnie, na przykład w mikrokontrolerze odbywa się wszystko lokalnie, a później model lokalny zostaje rozwijany w chmurze. [31]

Poniżej dokładnie opisano przykładowy case-study, który reprezentuje implementację takiego podejścia, składa się on z następujących etapów: pierwszym jest zbieranie danych przez czujniki, dane te są bezpośrednio przesyłane do bazy danych, która najczęściej jest bazą danych czasu rzeczywistego, która gromadzi dane zapisane w sekwencjach czasowych, Oznacza to, że każda mierzona próbka ma znacznik czasu, będący dokładną datą co do godziny, minuty i sekundy oraz milisekundy, w którym dana wartość została zmierzona oraz zapisana. Następnie dane, po trafieniu do bazy danych, są przesyłane poprzez REST API do skryptu, w którym zachodzi proces ich dalszej obróbki, w tym także sam proces trenowania takich danych. Wyniki, które ponownie zostają przesłane do osobnego punktu w bazie danych, są już specjalnie przygotowane i na ich podstawie zostaje podejmowana decyzja oysterowaniu konkretnego sygnału.

Opisane powyżej podejście do sterowania IoT ma większą skalowalność i decentralizację architektury sterowania, w porównaniu z metodami opartymi na TinyML.

3.1.Zastosowanie podejścia typu Cloud AI

Celem niniejszego rozdziału jest przedstawienie nowego podejścia do przetwarzania danych IoT w chmurze, polegającego na adaptacyjnym wyborze miejsca inferencji modelu ML oraz integracji analizy anomalii z systemem sterowania. Proponowane rozwiązanie pozwala na podniesienie efektywności energetycznej i bezpieczeństwa systemu.

Cloud AI (z ang. *Artificial Intelligence in the Cloud*) jest modelem przetwarzania danych, w którym algorytmy sztucznej inteligencji działają w chmurze obliczeniowej, a nie na urządzeniu końcowym. Technologie i platformy jakie znajdują wykorzystywanie w Cloud AI to [m.in.](#): [32]

- bazy danych czasu rzeczywistego takie jak : Firebase RealTime DB, Mongo DB Atlas, InfluxDB Cloud;
- platformy ML/AI Azure ML, Google Cloud AI;
- biblioteki do przeprowadzania modelowania ML TensorFlow, PyTorch, Scikit-learn;
- protokoły do komunikacji takie jak: MQTT, WebSocket, REST API;
- gotowe platformy chmurowe stworzone specjalnie dla projektów IoT takie jak: Blynk, ThingsBoard, Ubidots, AWS IoT Core.

Takie podejście Cloud AI ma wiele zalet w porównaniu z metodami TinyML. Te cechy, które są przeważające nad TinyML to [m.in.](#):

- duża moc obliczeniowa można trenować i uruchamiać duże modele (np. CNN, LSTM);
- łatwiejsze aktualizacje zmiany w modelu nie wymagają ponownego flashowania urządzeń;
- skalowalność łatwo zwiększyć liczbę urządzeń bez zmiany architektury.
- Dostępność narzędzi np. BigQuery, AutoML, Vertex AI (Google), SageMaker (AWS), Azure ML;
- integracja z bazami danych i dashboardami (np. Grafana, Power BI, Streamlit);
- współdzielenie danych ułatwia analizę przez zespoły badawcze / operatorów.

Cloud AI jest szeroko stosowane [m.in.](#) do np. wykrywania anomalii drgań w maszynach, tu dane z akcelerometrów analizowane przez model w chmurze, predykcijnym utrzymaniu ruchu, w nowoczesnych systemach Smart Home do klasyfikacji aktywności na podstawie danych z czujników ruchu, światła, dźwięku. Cloud AI znalazła także zastosowanie do wykrywania ataków poprzez przesyłanie danych sieciowych i ich analizę pod kątem anomalii, w tym wczesne ich wykrywanie, w oparciu o wytrenowane dane.[32]

Podejście Cloud AI ma też wady takie jak: zależność od Internetu, możliwość występowania opóźnień w zależności od szybkości łącza, bezpieczeństwo. Istnieją jednak rozwiązania, które mają na celu ograniczenie tych wad. Do tego typu rozwiązań należą [m.in.](#):

- buffering danych – polega na tym, że w przypadku przerwania połączenia internetowego istnieje możliwość przechowywania danych lokalnie np. w pamięci lub w karcie SD, SALite następnie po odzyskaniu połączenia internetowego dane te są ponownie przesyłane;
- ustawienie Edge AI jako backupu - gdy tymczasowo chmura z powodu braku dostępu do sieci jest niedostępna, jest drugi model ML, który w przypadku awarii będzie działał lokalnie [32]

3.2 Porównanie podejścia typu TinyML z Cloud AI

Zastosowanie TinyML jest prawidłowym rozwiązaniem w przypadku gdy docelowe rozwiązanie ma się skupiać na [44]:

- lokalnym miejscu obliczeń;
- braku dostępu do sieci;
- bardzo szybkim czasie reakcji, niezwiązanym z siecią;
- małej potrzeby skalowalności modelu.

Natomiast zastosowanie podejścia CloudAI jest prawidłowo dobranym rozwiązaniem w przypadku uzyskania wyniku końcowego, którego cechy systemu docelowego skupiają się głównie na:

- wysokiej skalowalności modelu czyli trenowania i przechowywania dużego modelu, który mógłby być w każdej chwili zmieniany przetrenowywany ponownie itp.;
- wysokiej złożoności modelu-jest to niezwykle kluczowe w przypadku modeli typu NLP, Deep Learning, ponieważ proste modele mają ograniczenie stosowania i przechowywania tylko w przypadku TinyML;
- dodatkowego wykorzystania innych serwisów chmurowych, które i tak już przetwarzają i dostarczają niezbędne dane.

Poniższa tabela oparta na bazie kilku publikacji wymienionych w bibliografii opartych na podejściu Cloud-based Machine Learning oraz aplikacji TinyML reprezentuje porównanie zbiorcze modeli TinyML oraz modeli CloudAI.[44]

Tabela 5. Porównanie TinyML oraz CloudAI[44]

Cecha	TinyML	CloudAI
miejsce obliczeń	lokalnie	chmura i lokalnie
dostęp do sieci	brak zależności od sieci	zależny od sieci
czas reakcji	niski	wysoki w przypadku wolniejszego połączenia sieciowego
prywatność danych	minimalne ryzyko utraty wrażliwych danych	wysokie ryzyko wynikające z przesyłania danych między serwerami oraz przechowywania danych w aplikacji w chmurze
modele	tylko proste, ograniczenia co do wielkości	możliwość trenowania złożonych modeli

Reasumując podejście TinyML oraz CloudAI mają zarówno wady i zalety, które trzeba wziąć przy projektowaniu inteligentnego systemu w zależności od wstępnie określonych założeń i wymagań. TinyML wyróżnia się tym, że ma bardzo proste zastosowanie począwszy

od samego wdrożenia, a skończywszy na lokalnym i niezależnym od sieci działaniu, które polega na przetrenowaniu modelu lokalnie, następnie optymalizacji tak aby zmieścił się na mikrokontrolerze i późniejszej konwersji modelu do formatu C oraz wgraniu do mikrokontrolera, co pozwala żeby sztuczna inteligencja działała bezpośrednio na urządzeniu, które ma ograniczone zasoby(kilkadziesiąt lub kilkaset kilobajtów pamięci RAM).

Natomiast CloudAI jest podejściem bardziej popularnym ze względu na swoją większą moc obliczeniową oraz możliwość rozbudowy systemu opartego na AI i IoT w bardziej nowoczesne aplikacje zintegrowane z interfejsami webowymi.[33]

3.3 Przetwarzanie danych oraz budowa modeli ML

Przetwarzanie danych jest bardzo istotnym i nieodłącznym krokiem w przypadku projektowania i modelowania oraz późniejszego wdrażania modeli.

Przesyłanie danych do chmury przez mikrokontroler oraz odbiór odpowiednich decyzji również przez system z mikrokontrolerami, opisane w niniejszej pracy, może wiązać się z ryzykiem potencjalnych przerw oraz zakłóceń stabilności sieci czego wynikiem mogą być: puste dane wysłane z określonym timestampem; przerwa w wysyłaniu danych, która definiuje przerwę w ciągłości czasowej wysłanych danych; ponadto istotny wpływ ma zasilanie mikrokontrolerów ponieważ jeśli zasilanie zostanie odcięte, dane z określonego czasu mogą zostać utracone. Potencjalne zagrożenia wynikające z utraty danych (nawet jeśli jest to zwyczajna strata wynikająca z chwilowych wahań zaprojektowanego systemu), czy też dane te zostaną przechwycone przez niewłaściwe osoby, może okazać się problematyczna dla procesu tworzenia modeli ML, dlatego skuteczne przetwarzanie danych powinno obejmować:

- wykrywanie luk czasowych (w przypadku baz danych czasu rzeczywistego) – proces ten zazwyczaj jest przeprowadzany poprzez algorytmy wykrywające potencjalne “przerwy” w danych;
- imputacja danych – czyli uzupełniania luk, m.in. używając metod: interpolacji liniowej, lub wykorzystując metodę uśredniania ostatnich sąsiednich punktów;
- uczenie modeli z uwzględnieniem luk – polega na dodawaniu specjalnych cech lub zmiennych binarnych, które definiują pustkę, czyli zamiast automatycznego NaN (ang. Not a Number) można zmienić cechą opisową np. “leak”.

Budowa modeli ML czyli modeli uczenia maszynowego składa się z kilku ważnych etapów, na które składa się: wstępne rozpoznanie problemu, który ma opisywać algorytm ML; oraz typ modelu ML, począwszy od modeli mających za zadanie sklasyfikować dane, przewidzieć dane lub grupować dane sensoryczne. Jednakże niezależnie od tego jaki typ modelu będzie konstruowany należy uprzednio podjąć szereg kroków opisanych w poniższych punktach 1 do 4.

1. Zebranie danych i wstępna i ich identyfikacja:

- zebranie i oczyszczenie danych – im więcej danych tym model będzie miał większą przestrzeń do właściwego wytrenowania się, gdzie zbieranie danych jest niczym innym jak pobraniem danych z bazy poprzez API, wcześniej wysłanych do tej bazy, lub poprzez eksport danych w oczekiwanej postaci , np. plików .csv;

- eksploracja danych – niewymagany, ale wstępnie dobrze jest zrozumieć tendencje danych, np. poprzez kreowanie wykresów przedstawiających przebieg danych;
- wstępna identyfikacja braków danych czy szumów – celem ich obsługi lub usunięcia w kolejnym kroku.

2. Proces przetwarzania danych:

- czyszczenie danych z braków oraz duplikatów – tu należy uwzględnić, że są dane, których duplikaty muszą pozostać celem prawidłowego i naturalnego określenia modelu tendencji danych;
- skalowanie oraz normalizacja – jest to istotny etap, który obejmuje proces przeskalowania danych różnego rodzaju np. natężenia światła, prądu, napięcia – posiadających odmienne wartości amplitud i różne jednostki, do wspólnej skali;
- zdefiniowanie zmiennych kategorycznych;
- definiowanie cech;
- utworzenie zbioru treningowego.

3. Trenowanie modelu:

- wyznaczenie algorytmu według, którego ma być wytrenowany model – czyli główne takie opcje jak m.in. Random Forest oraz Xboost, oba sprawdzają się na przeprowadzonych w części badawczej eksperymentów zawierających aktualnie mierzone dane testowe;
- przeprowadzenie treningu modelu według wybranego algorytmu, na wcześniej wybranym zbiorze testowym.

4. Ocena utworzonego modelu:

- wyznaczenie metryk oceniających model takich jak: accuracy, precision, recall, RMSE, MAE, gdzie :
 - accuracy – jest określana jako dokładność wyliczone poprzez iloraz liczby poprawnych predykcji a liczbę wszystkich przypadków, inaczej na podstawie tego parametru można stwierdzić w jakim stopniu model ma rację czyli jego predykcja będzie uznawana za trafną;
 - precision – precyzja, świadczy o liczebności tzw. trafionych wyników, jeśli precyzja wynosiłaby 70% oznaczałoby to, że 70 na 100 wyników jest prawdziwych, a 30 na 100 błędnych;
 - RMSE – wyznacza średnie błędy podniesione do kwadratu, im wyższy RMSE (ang. *Root Mean Squared Error*), tym większe i bardziej zróżnicowane są błędy predykcji modelu;
 - MAE – prosta średnia błędów (ang. *Mean Absolute Error*), wyznacza liczbowo o ile model się myli.

4. Cyberbezpieczeństwo w IoT

Internet of Things (IoT) jest wyjątkową dziedziną, która ma zastosowanie w wielu obszarach, m.in. budowaniu warstwy sensorycznej i sterowania budynków inteligentnych (ang. Smart Homes) jak również znajduje miejsce w infrastrukturze krytycznej jak: w zdalnych odczytach liczników energii (inaczej smart metering), inteligentnych sieci energetycznych (smart grids), a także w przemyśle i produkcji (Przemysł4.0). W wyniku tego rozpowszechnienia się zastosowania Internetu Rzeczy wzrastają potencjalne zagrożenia na ataki tej warstwy komunikacyjnej systemów infrastruktury krytycznej. [16]

Wraz ze wzrostem zagrożeń, rośnie także potrzeba wdrażania nowych form zabezpieczeń oraz norm jakie muszą spełniać nowoczesne systemy infrastruktury krytycznej oparte na koncepcji chmury oraz IoT[6]. Ochrona urządzeń przed zagrożeniami cybernetycznymi sprowadza się nie tylko do ochrony pod względem technicznym, coraz bardziej powszechna staje się ochrona prawna.

Ochrona pod względem prawnym zaczęła przejawiać się już na początku 2018 roku gdy została wprowadzona dyrektywa NIS1 (Network and Information Security no 1), dyrektywa ta zobowiązywała podmioty infrastruktury krytycznej takie jak: energetyka, transport, wodociągi, do obowiązkowego wdrożenia wskazanych w dyrektywie środków bezpieczeństwa warstwy cyfrowej, które głównie polegały na zgłaszaniu incydentów do CERT (Computer Emergency Response Team) oraz współpracowania z krajowymi organami nadzorującymi, takimi jak np. w Polsce jest NASK (Naukowa i Akademicka Sieć Komputerowa) celem niwelowania oraz zarządzania ryzykiem związanym z zagrożeniami cyberbezpieczeństwa. Od roku 2025 weszła w życie dyrektywa NIS2 – która rozszerzyła odpowiedzialność za bezpieczeństwo systemów ICT (ang. *Information and Communications Technology*) również na takie sektory jak [m.in](#) zarządzanie odpadami lub produkcja istotnych produktów takich jak leki, sprzęt medyczny lub chemikaliów, żywności. [34]

4.1 Zagrożenia oraz zabezpieczenia dla systemów IoT

Urządzenia IoT są podatne na potencjalne ryzyko ataków, gdyż z powodu ich technicznych ograniczeń, czyli małej pamięci RAM, niskiej mocy obliczeniowej, a tym samym wpływających na możliwości zastosowania zabezpieczeń takich jak: TLS(z ang.) czy standardowego szyfrowania wysyłanych danych. Dla przykładu tanie i łatwo dostępne urządzenia jak mikrokontrolery serii Espressif [21] nie posiadają oficjalnych i osobnych regulacji pod względem bezpieczeństwa. Same w sobie mikrokontrolery, pozostające w powszechnym użytku, mogą jedynie podlegać dyrektywie NIS1, gdy już wchodzi w podkład urządzeń stosowanych do infrastruktury krytycznej, tam gdzie w danym obszarze te regulacje już są.

Te doprecyzowanie pod względem prawnym oraz widoczny podział regulacji prawnych pomiędzy: działającym systemem/urządzeniem z obszaru infrastruktury krytycznej a pojedynczymi elementami wchodzącymi w skład danego rozwiązania, sprawia, że rośnie ryzyko naruszeń prawnych wynikających z nieścisłości, które powinny być doprecyzowane wraz z rozwojem technologii IoT oraz nowoczesnych urządzeń wykorzystujących te podzespoły.

Główne czynniki, które korelują miarę zagrożenia ze strony bezpieczeństwa to [m.in.](#):

- problematyczny, często niskopoziomowy hardware, którego ograniczenia sprzętowe powodują brak możliwości zastosowania klasycznych zabezpieczeń;
- brak jasnych standardów określonych przez producentów, co przekłada się na jakość zabezpieczeń ich produktów;
- słabe zabezpieczenia sieci, przez co możliwe są awarie, podczas których może dojść do przejęcia kontroli nad częścią infrastruktury.

Wyżej wymienione czynniki determinują potencjalne zagrożenia takie jak:

- dostęp bez autoryzacji, oznacza to, że każdy użytkownik może na skutek złego mechanizmu autoryzacji lub błędów związanych z tokenami dostępu uzyskać dostęp do danych, a w skrajnym przypadku przejąć kontrolę poprzez nałożenie własnych mechanizmów autoryzacji;
- złośliwe oprogramowania typu malware, mające na celu całkowite lub chociażby częściowe zakłócenie działania wybranych obszarów z infrastruktury krytycznej;
- fałszowanie danych, obejmujące wstrzykiwanie fałszywych danych celem zmiany zachowania określonego systemu lub wprowadzenie w błąd jego użytkowników;
- przeciążanie urządzeń IoT, powodując czasowy brak dostępu to jej usług;
- tzw.data-leakage czyli przemycanie przesyłanych danych.

Posiadając wiedzę o czynnikach determinujących wyżej wymienione zagrożenia, należy im przeciwdziałać, zarówno pod względem technicznym, implementując najlepsze dostępne na rynku sposoby i narzędzia walki z cyberatakami – co jest oczywiście kosztowne – ale także należy budować świadomość użytkowników i odpowiedzialność osób decyzyjnych pod względem prawnym. Rozdzielenie zabezpieczeń pod takim kątem dokonano już w normie ISO/IEC 27001 i 27002. Zabezpieczenia można podzielić na:

- zabezpieczenia prawne oraz organizacyjne;
- zabezpieczenia techniczne.

Skupiając się na zabezpieczeniach prawnych można wyróżnić nową dyrektywę wprowadzoną w 2023 tj. NIS2 (UE 2022/2555 z ang. Directive on Security of Network and Information Systems 2 czyli Dyrektywa w sprawie bezpieczeństwa sieci i systemów informatycznych w wersji drugiej. Dyrektywa ta jest następstwem dyrektywy NIS1, jednocześnie jest jej aktualizacją. Najważniejszą aktualizacją w NIS2 to rozszerzenie zakresu odpowiedzialności na sektory takie jak: produkcja, zarządzanie odpadami, zarządzanie wodą. Dyrektywa NIS2 wpływa też na kreowanie stanowisk firm poprzez organizacje szkoleń lub pobieranie sankcji w przypadku naruszenia prawa. W tabeli nr 6 przedstawiono główne aspekty, w których dyrektywa NIS2 rozszerzyła poprzednią dyrektywę NIS1[34].

Tabela 6. Główne założenia dyrektywy NIS2[34]

Dyrektywa NIS2
Sektory główne to: energetyka, transport, zarządzanie odpadami, zarządzanie wodą.
Zwiększenie odpowiedzialności kadry kierowniczej.

Obowiązek zgłaszania incydentów oraz ataków cybernetycznych, obowiązek zgłaszania incydentów oraz w przeciągu 24h od wykrycia.
Zwiększone wymogi bezpieczeństwa w zależności od aspektów środków technicznych i organizacyjnych.
NIS2 wymaga od podmiotów oceny i zarządzania ryzykiem u dostawców, np. sprzętu IoT, oprogramowania, usług zewnętrznych.

Zabezpieczenia używane w infrastrukturze krytycznej to [m.in](#) zabezpieczenia według podziału przedstawionego w punkcie 4.1.. Do bardziej szczegółowych zabezpieczeń prawnych ukierunkowanych na ochronę systemów IoT infrastruktury krytycznej należą:

- wspomniana już wcześniej dyrektywa NIS2;
- europejskie programy certyfikacji;
- normy bezpieczeństwa dla IoT.

Ponadto istotnym bardzo rodzajem zabezpieczeń w infrastrukturze krytycznej są zabezpieczenia techniczne, które obejmują:

- aplikacje monitorujący ruch w sieci takie jak aplikacja Splunk - aplikacja do analizy ruchu, która zbiera logi z bram IoT, systemów operacyjnych urządzeń, usług chmurowych;
- różnego rodzaju firewalle jak IoT aware firewall lub podejście polegające na segmentacji sieci dzięki czemu komunikacja z urządzeniami IoT jest ograniczona do niezbędnych urządzeń, portów.

5. Opracowanie systemu predykcyjnego dla magazynu energii IoT

Omówiony w pracy magazyn energii jest jedynie modelem prototypowym, a nieodłącznym jego suplementem jest aplikacja webowa stworzona do wystawiania konkretnymi jego wyjściami odpowiedzialnymi za m.in proces ładowania akumulatora, proces ładowania urządzeń zewnętrznych podpiętych do tego akumulatora. W trakcie badań, z układu ładowania korzystał: mikrokontroler, telefon komórkowy, elementem wspomagającym był kontroler ładowania, który pośredniczył między panelami fotowoltaicznymi oraz akumulatorem podpiętym do naładowania. Kontroler ładowania służył również do poprawnego obniżania napięcia wyjściowego z paneli fotowoltaicznych, aby było ono odpowiednio dostosowywane do rodzaju podpiętego akumulatora. Na stanowisku badawczym stosowano naprzemiennie akumulatory: żelowy 12V 5Ah 1,5A , 18W lub ogniwa litowo-jonowe 3,6V, 3000mAh (tutaj stosowane wraz z przetwornicą step-down i dodatkowym modulem ładowania TP4056) Układ pomiarowy składał się z następujących sensorów:

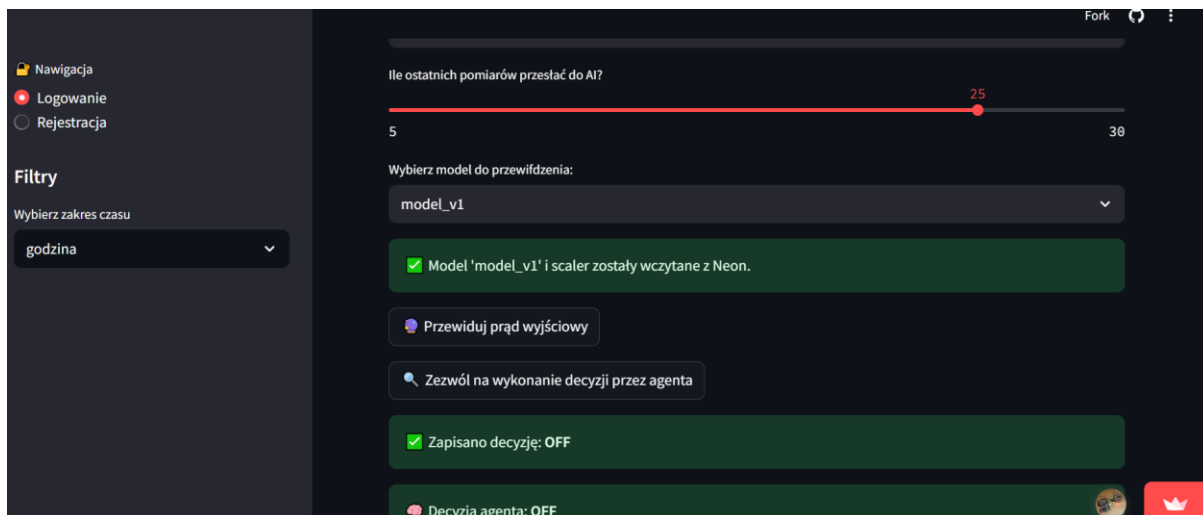
- czujniki prądu, które mierzyły wartość prądu ładowania, prąd wyjściowy INA219 (określany też jako prąd rozładowywania);
- DHT22 – który mierzył temperaturę na zewnątrz (temperaturę pracy), poziom nasłonecznienia, wilgotność (czyli aktualne warunki pogodowe).

Sensory, których użyto do przeprowadzania takich pomiarów to: czujnik prądu i napięcia INA219 (I2C) , DHT22 oraz BH1750 lub TCM6000.

5.1 Cel i funkcjonalność systemu

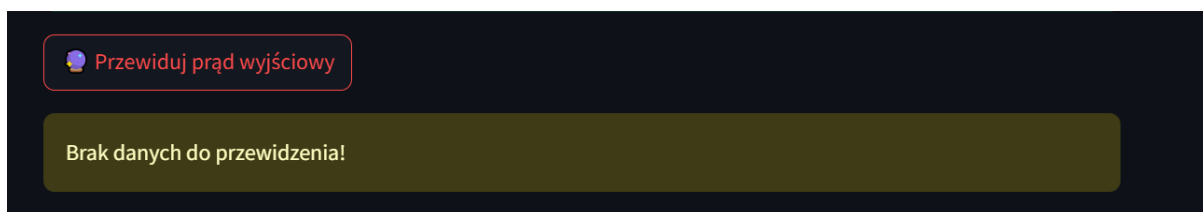
Głównym celem systemu magazynu energii jest sterowanie procesem ładowania akumulatora oraz jego rozładowywania w oparciu o dane przewidziane w procedurze predykcji wymaganych dla magazynu stanów pracy.

System jest oparty na praktycznym wykorzystaniu agentów AI opartych na modelach LLM takich jak: Microsoft MAI, czy GPT 3.5 turbo od OpenAI oraz internVL, a także modeli predykcyjnych, które w pozwalały użytkownikowi na podjęcie decyzji o wystawianiu konkretnego wyjścia mikrokontrolera. Dodatkowo po wybraniu przez użytkownika systemu opcji: „*Zezwól na wykonanie decyzji przez agenta*”, agent LLM będzie podejmował decyzję od razu oraz pisał krótkie wyjaśnienie jej podjęcia (Rys.2). Decyzje podejmowane przez agenta wynikają z wcześniejszego fine-tuningu opartego o dane wyciągnięte z bazy, dla wybranego przez użytkownika przedziału czasowego. Domyślnie, gdy system nie jest włączony i dane nie są zbierane w czasie rzeczywistym, agent AI podejmuje decyzje w oparciu o historyczny przedział czasowy oraz liczbę ostatnich pomiarów jaką należy przesłać agentowi. Im więcej pomiarów zostanie ujętych w zakresie, tym bardziej agent podejmie decyzje interpretowane dla większych odstępów czasu, gdyż skupi się na znacznie większej liczbie pomiarów (Rys.2).

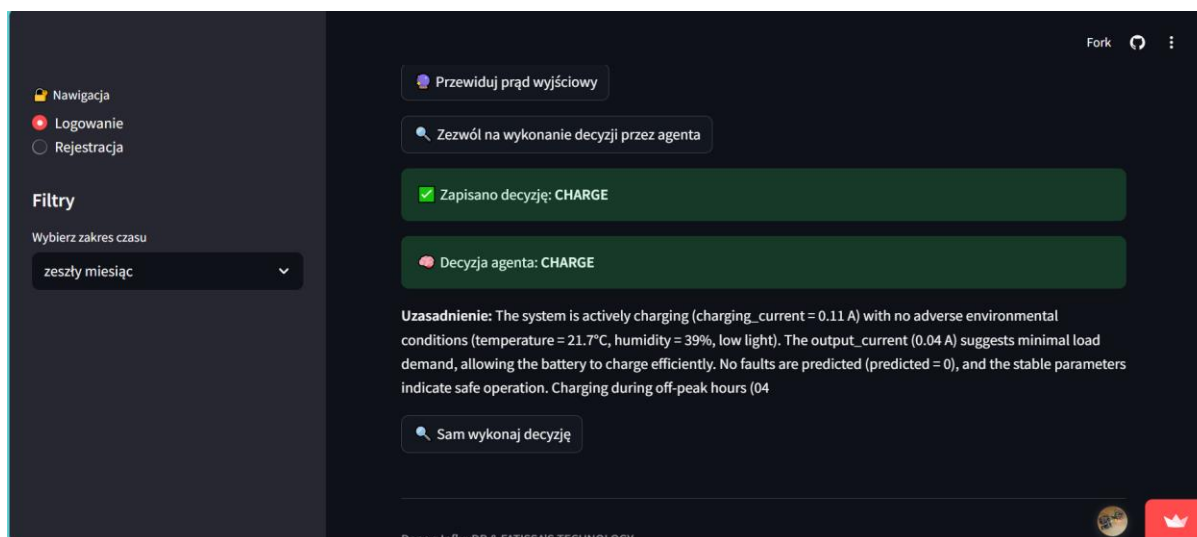


Rysunek 2. Wybór liczby pomiarów przesłanych do agenta.

Sterowanie predykcyjne oparte jest na modelach AI do przewidywania prądów oraz poziomu oświetlenia i pozostałych warunków pogodowych, które opierają się na algorytmie XBoost, które wymagają do prawidłowego procesu trenowania najlepiej danych w liczbie od kilku setek do tysięcy przykładów, dlatego dane przesyłane do algorytmu są wybierane z określonego obszaru czasowego (rys. 2. – filtr po lewej). Jeśli obszar czasowy nie zostaje wybrany przez użytkownika to domyślnie algorytm określa wartości przewidziane na najbardziej świeżych pomiarach z ostatniego czasu jakie są dostępne w bazie. W przypadku gdy w wybranym zakresie filtru nie ma pomiarów, użytkownik dostaje stosowny komunikat o braku danych (Rys.3).

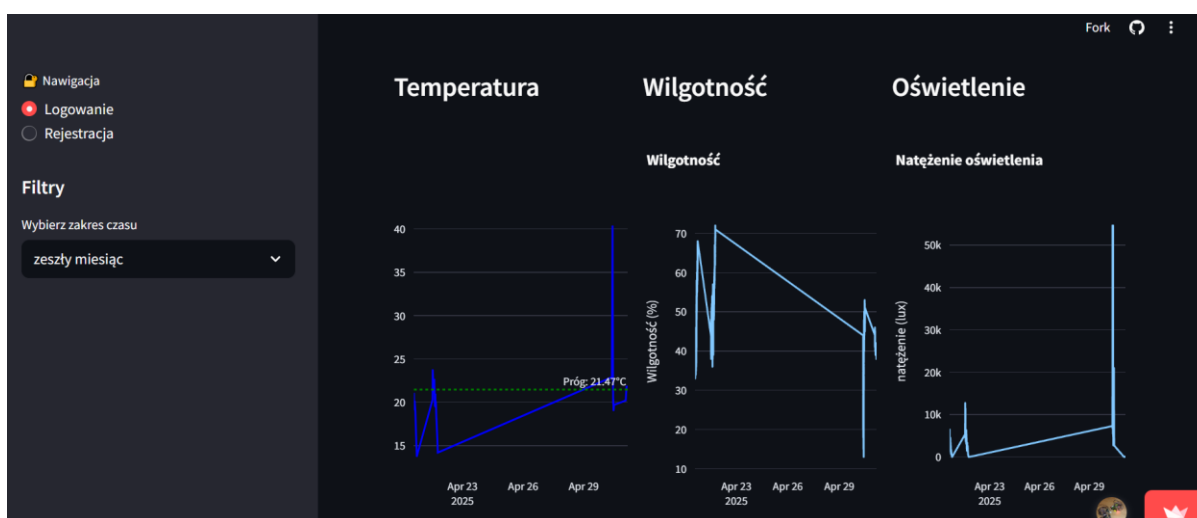


Rysunek 3. Komunikat systemu decyzyjnego



Rysunek 4. Prezentacja działania jednego z zapytanych o decyzję agentów

W przypadku podejmowania decyzji przez agentów AI dużym ułatwieniem jest to, że nie wymagają one przesyłania danych testowych o dużej liczbie oraz trenowania, tak jak to było przedstawione w poprzedniej części dla wykorzystywanych własnych modeli do stworzenia predykcji. Aplikacja zawiera dwie możliwości predykcji, zarówno tej opartej o modele predykcyjne własne, które głównie bazują na algorytmie XBoost jak i możliwości predykcji i sterowania poprzez agenta AI dostosowanego do podejmowania decyzji i równoczesnego sterowania tymi decyzjami. Agenci AI oparte na LLM dostosowywane są na podstawie 5 lub maks. 30 ostatnich pomiarów i w przypadku chwilowego zagrożenia wystąpienia anomalii lub powtarzalnych danych, których podgląd znajduje się w aplikacji na wykresach (Rys.5 i Rys.6), mogą podjąć tak samo skuteczną decyzję jak modele uczenia na dużych zbiorach danych, których to algorytmy czasowo oraz pamięciowo znacznie bardziej zasadochłonne pod względem obliczeniowym.



Rysunek 5. Podgląd danych w aplikacji na wykresach



Rysunek 6. Podgląd danych na wykresach cd

Wszystkie modele skupiają się na przewidywaniu wartości takich jak: predykcja prądu ładowania akumulatora, predykcja prądu pobieranego, predykcja natężenia światła, predykcja temperatury, predykcja wilgotności, predykcja czasu ładowania akumulatora.

5.1 Platformy i narzędzia wykorzystane w implementacji systemu

System magazynu energii składa się z części hardwarowej oraz software'owej, wraz z implementacją nowoczesnych jak i tych klasycznych rozwiązań AI, częściowo bazując na umiejętnym rozszerzaniu gotowych rozwiązań takich jak agenci AI systematyzując je pod rozwiązywanie określonego systemu.

Implementacje systemu można podzielić na :

- **Hardware** – w którego w skład wchodzi: mikrokontrolery firmy Espressif w postaci płytek rozwojowych ESP32 DOIT DEVKIT V1, kontroler ładowania, który pozwala odpowiednio obniżyć napięcie z wyjścia panel dla dobranego typu akumulatora aktualnie podpiętego do systemu oraz sensory służące do pomiaru badanych wielkości, które są zbierane do mikrokontrolera oraz później przesyłane w czasie rzeczywistym do bazy danych w chmurze;
- **Software** - składający się z następujących frameworków, platform i rozwiązań chmurowych oraz środowiska programistyczne:
 - Środowiska programistyczne i platformy chmurowe jak:
 - Visual Studio Code z rozszerzeniem PlatformIO użyte do programowania mikrokontrolerów z rodziny ESP;
 - Influx Database – chmurowa baza danych czasu rzeczywistego;
 - Neon – relacyjna baza danych w chmurze (odpowiednik bazy lokalnej Postgress);
 - Narzędzia wspomagające rozwój i bezpieczeństwo:
 - Insomnia; aplikacja umożliwiająca testowanie poszczególnych endpointów serwera oraz ich publicznej dostępności;

- WireShark do testów; narzędzie do analizy ruchu sieciowego, które umożliwia przechwytywanie przesyłanych pakietów danych;
- ZAP do testów - oprogramowanie do testowania różnych form ataków na aplikacje webowe m.in : wstrzykiwanie złośliwego kodu, testy endpointów, przeprowadzanie ogólnego skanu aplikacji pod względem jej podatności na zagrożenia cybernetyczne i pokazywanie jej słabych punktów w raporcie końcowym;
- Frameworki użyte do budowy aplikacji webowej:
 - Streamlit – do tworzenia interaktywnych dashboardów i wizualizacji danych;
 - Django – do budowy backendu systemu, obsługi API i logiki serwera.

Dodatkowo do stworzenia modeli predykcyjnych użyto następujących algorytmów:

1. RandomForestRegressor,
2. XGBRegressor (XGBoost),
3. LGBMRegressor (LightGBM),
4. Linear Regression.

W następnym rozdziale opisano badania, które pokazują jak powyższe algorytmy będą sprawdzać się w predykcji danych gromadzonych przez czujniki skalibrowane z prototypem MVP aplikacji oraz systemu inteligentnego magazynu energii, a które należałoby z wdrożenia do aplikacji wykluczyć całkowicie. W końcowym etapie, gdy już najlepszy wybór algorytmu do przewidywania będzie znany, ten zostanie wyodrębniony do stworzenia modeli dla pozostałych danych, które mają być przewidywane (czyli pozostałe wartości takie jak: oświetlenie, temperatura, prąd wyjściowy).

5.2 Inteligentne sterowanie IoT oparte na algorytmach AI

Do celów trenowania modeli predykcyjnych do przewidywania wybranych cech pomiarowych, w zależności od pozostałych cech wykorzystano następujące algorytmy:

- RandomForestRegressor,
- GradientBoostingRegressor
- LGBMRegressor (z LightGBM),
- Linear Regression.

Dla każdego modelu zostały wyznaczone parametry: MSE oraz R2 Score, które świadczą o wysokości błędu predykcji oraz o skuteczności uczenia danego modelu.

1. Wyniki skuteczności trenowania modelu dla predykcji prądu ładowania algorytmem Gradient Boosting Regressor z biblioteki stici-learn

```
Dane pobrane: 3842 rekordów
Trenuję model do charging_cur
Model gotowy.
MSE: 0.0029
R2 Score: 0.9266
```

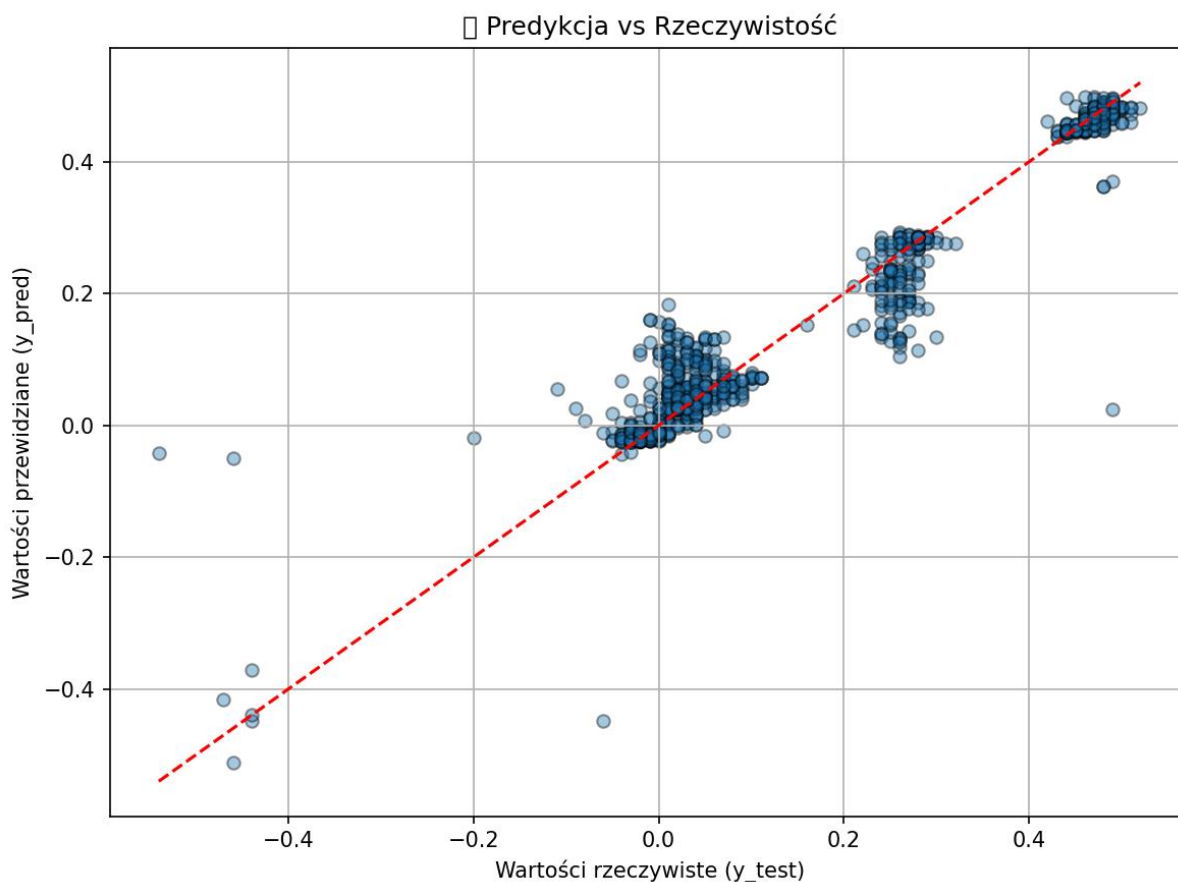
Rysunek 7. Zrzut wyników dla GradientBoostingRegressor

```
Model gotowy. Czas trenowania: 0.47 sekundy
```

Rysunek 8. Wyznaczenie czasu trenowania dla Gradient Boosting Regression

Interpretacja:

- MSE (Mean Squared Error): bardzo niskie — 0.0029 → błąd predykcji jest minimalny,
- R^2 (R2 Score): 0.9266, czyli 92.7% zmienności danych zostało w skuteczny sposób przetworzone przez model, tylko 7 % było sklasyfikowane jako dane błędnego pomiaru.

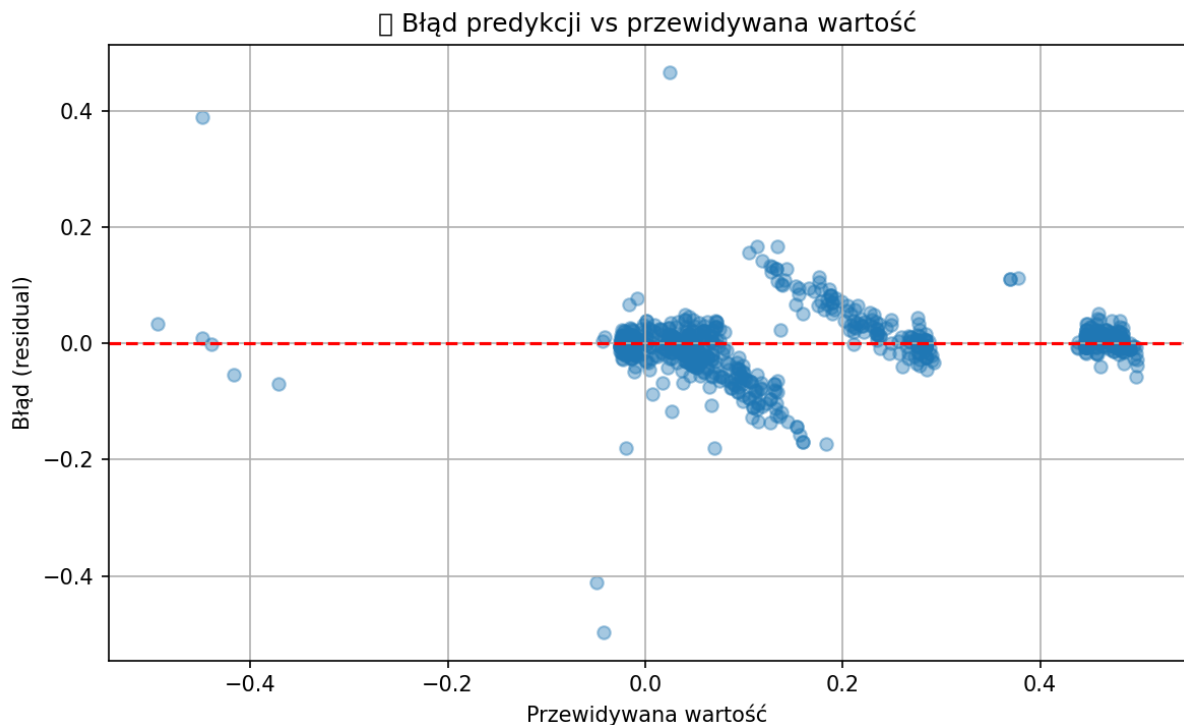


Rysunek 9. Porównanie predykcji prądu ładowania z wartościami mierzonymi dla Gradient Boosting Regression

Interpretacja wykresu:

Widoczne niebieskie punkty reprezentujące rzeczywiste wartości pomiarowe są w większości skupione do przebiegu prostej predykcyjnej (czyli czerwonej linii przerywanej) oznacza to duże

skupienie i zarazem pokrycie wyników rzeczywistych z produkowanymi wynikami co stanowczo wpływa na sformułowanie wniosku o tym, że system predykcji dla przewidywania prądu ładowania będzie działał skutecznie.



Rysunek 10. Błąd predykcji dla Gradient Boosting Regression

Interpretacja wykresu błędów: błąd predykcji jest minimalny, wynosi on zaledwie: 0,003, wartość ta definiowana jest jako MSE i pozostała obliczona w programie konsolowym, fragment wyjścia programu widocznego na zrzucie (Rys.11), wynik ten świadczy o bardzo skutecznej predykcji prądu ładowania.

2. Wyniki skuteczności dla tworzenia modelu do predykcji prądu ładowania algorytmem LGBM REGRESSOR:

```
MSE: 0.0025  
R2 Score: 0.9370
```

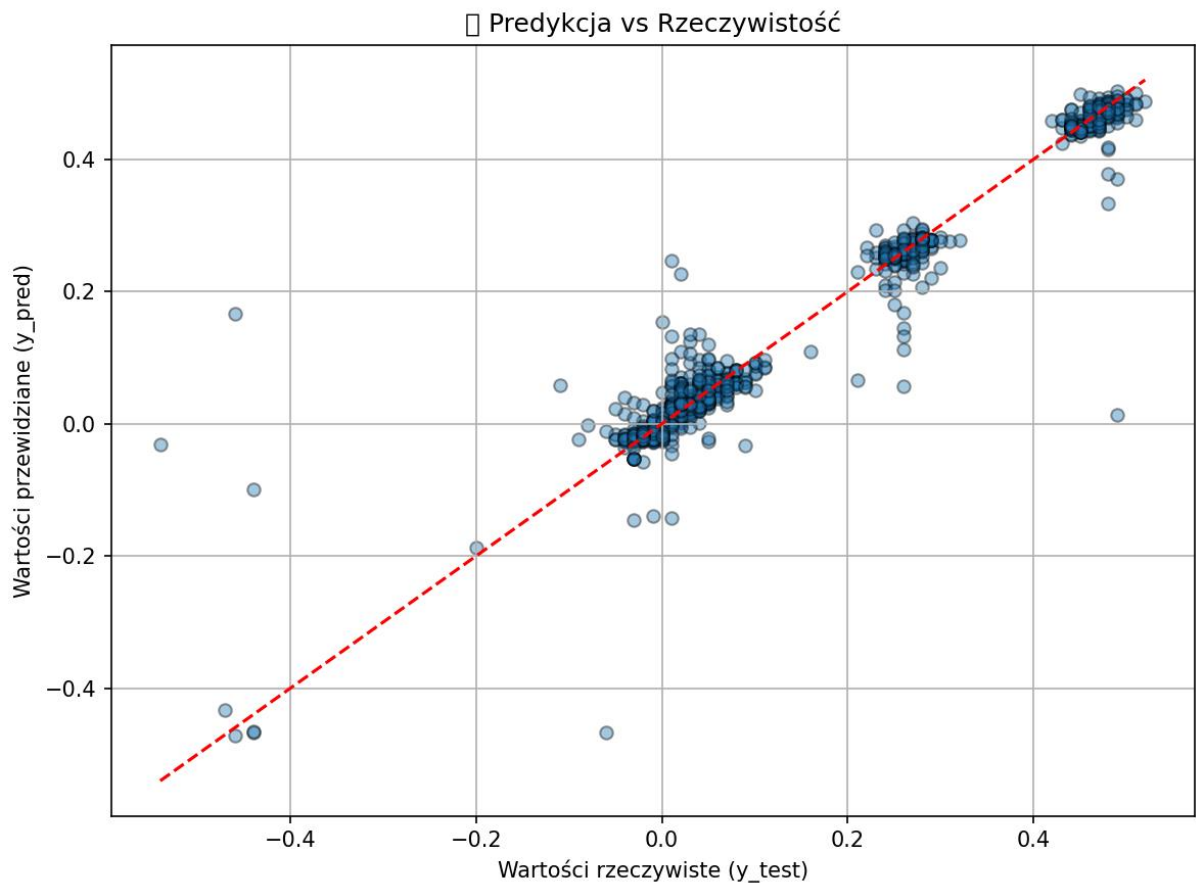
Rysunek 11. Wyniki dla algorytmu LGBM Regression

```
Model gotowy.  
czas trenowania: 2.51 sekundy
```

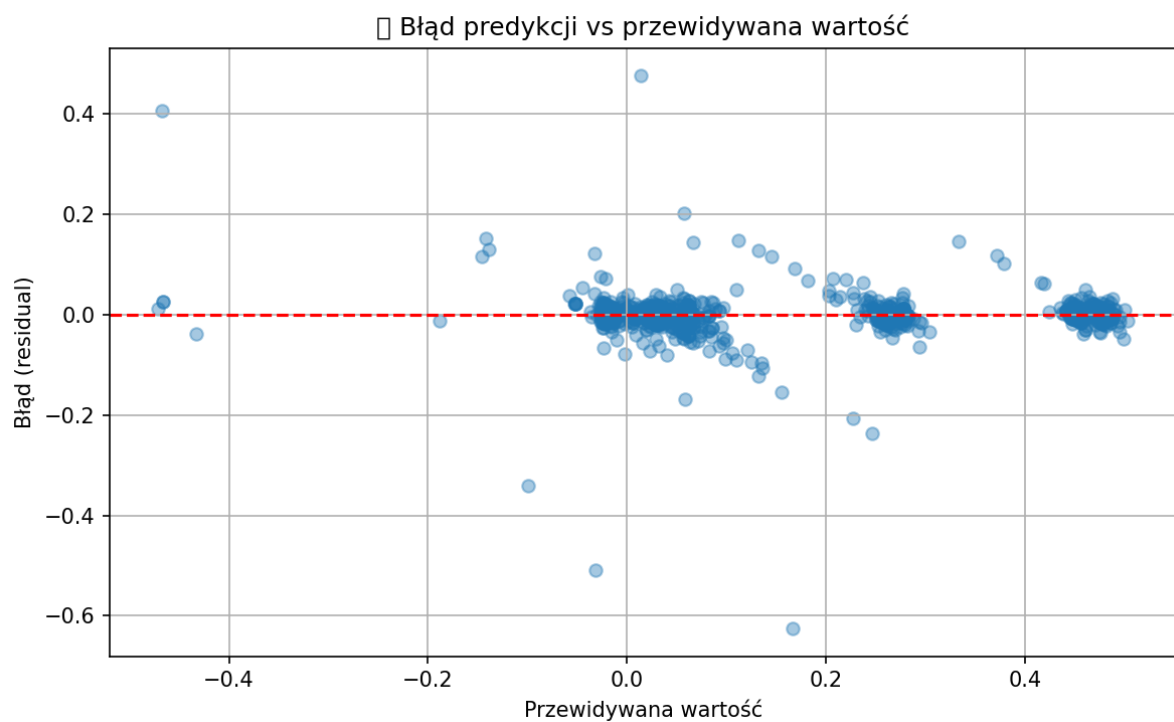
Rysunek 12. Czas trenowania dla LGBM Regression

Interpretacja

Dla algorytmu LGBM Regressor, użytego w tworzeniu modeli do przewidywania prądu ładowania, tak jak poprzednio, wynik błędu predykcji jest mniejszy, natomiast wynik skuteczności predykcji jest nieznacznie większy.



Rysunek 13. Wykres porównawczy predykcji z wynikami rzeczywistymi dla LGBM Regression



Rysunek 14. Wykres błędu predykcji dla LGBM Regression

3. Wyniki skuteczności dla tworzenia modelu do predykcji prądu ładowania

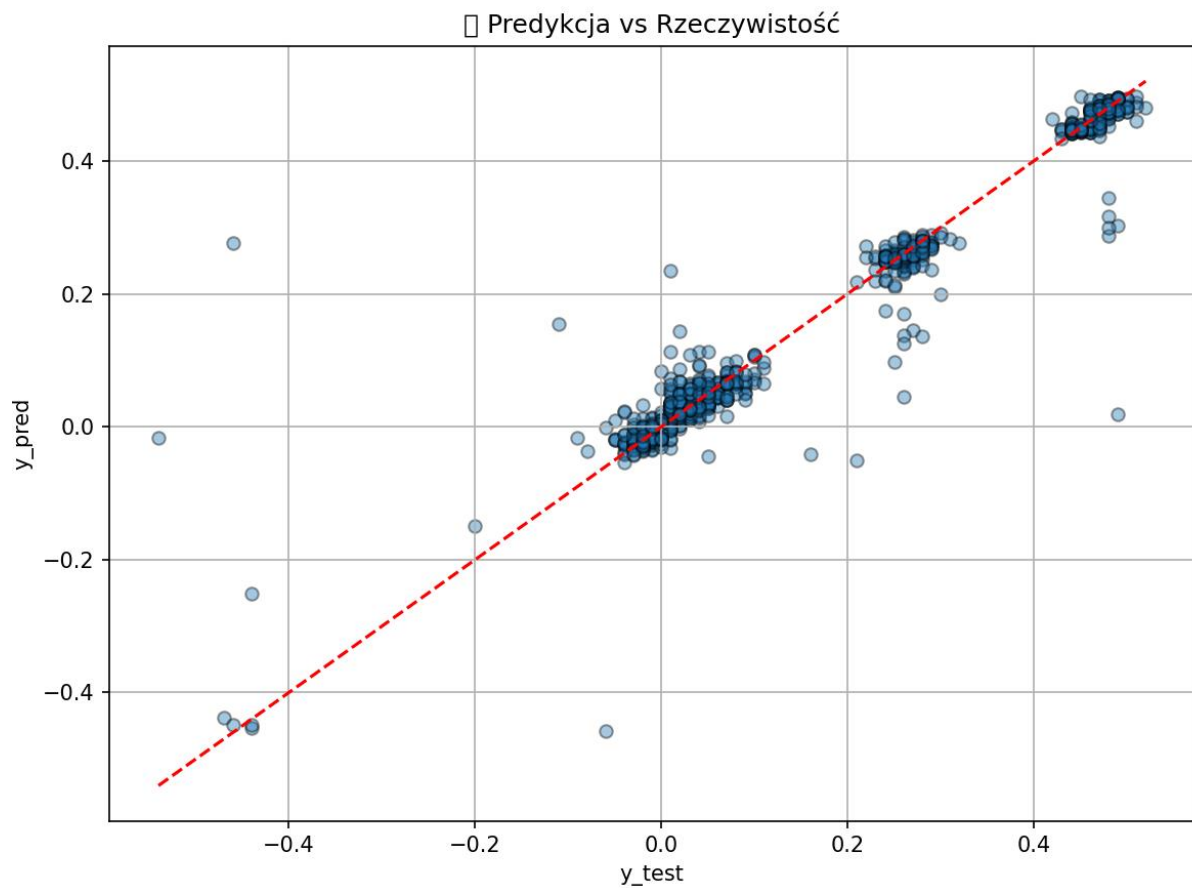
RandomForestRegressor,:

```
MSE: 0.0027  
R2 Score: 0.9324
```

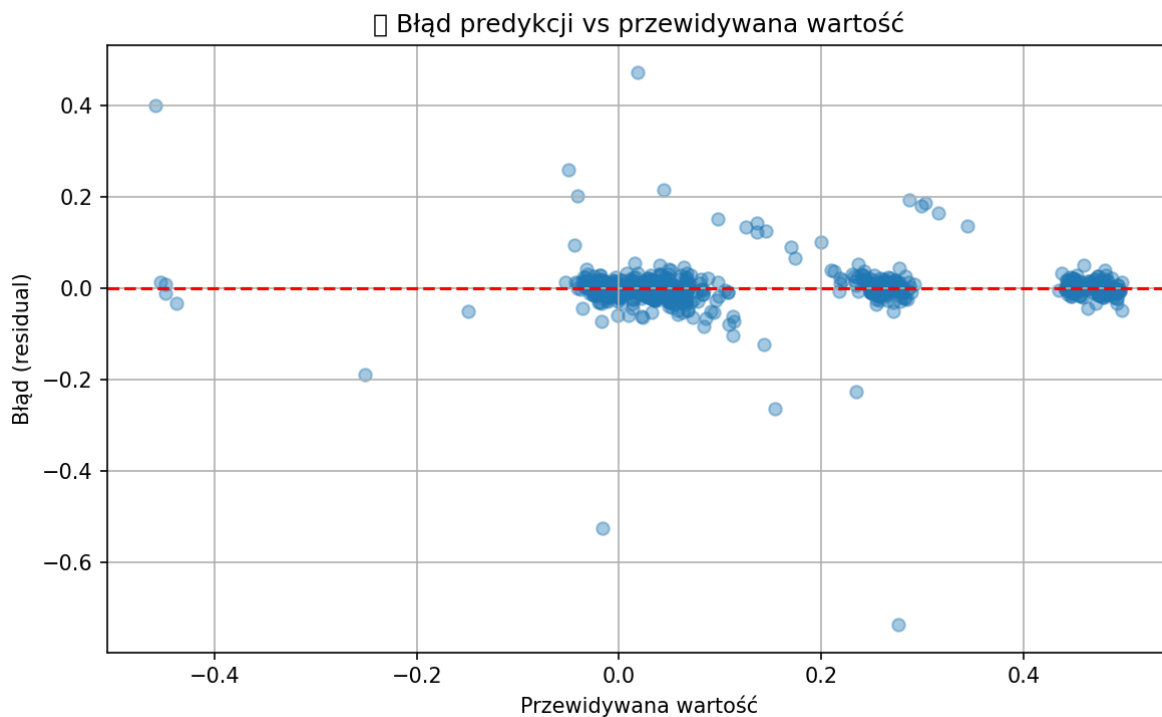
Rysunek 15. Wyniki dla algorytmu RandomForest Regression

```
Model gotowy. Czas trenowania: 1.52 sekundy  
MSE: 0.0027  
R2 Score: 0.9315
```

Rysunek 16. Wynik czasu trenowania dla RandomForest Regression



Rysunek 17. Wykres predykcji prądu ładowania w porównaniu z mierzonymi wartościami rzeczywistymi dla RandomForest Regression,:



Rysunek 18. Wykres błędu predykcji dla algorytmu RandomForest Regression

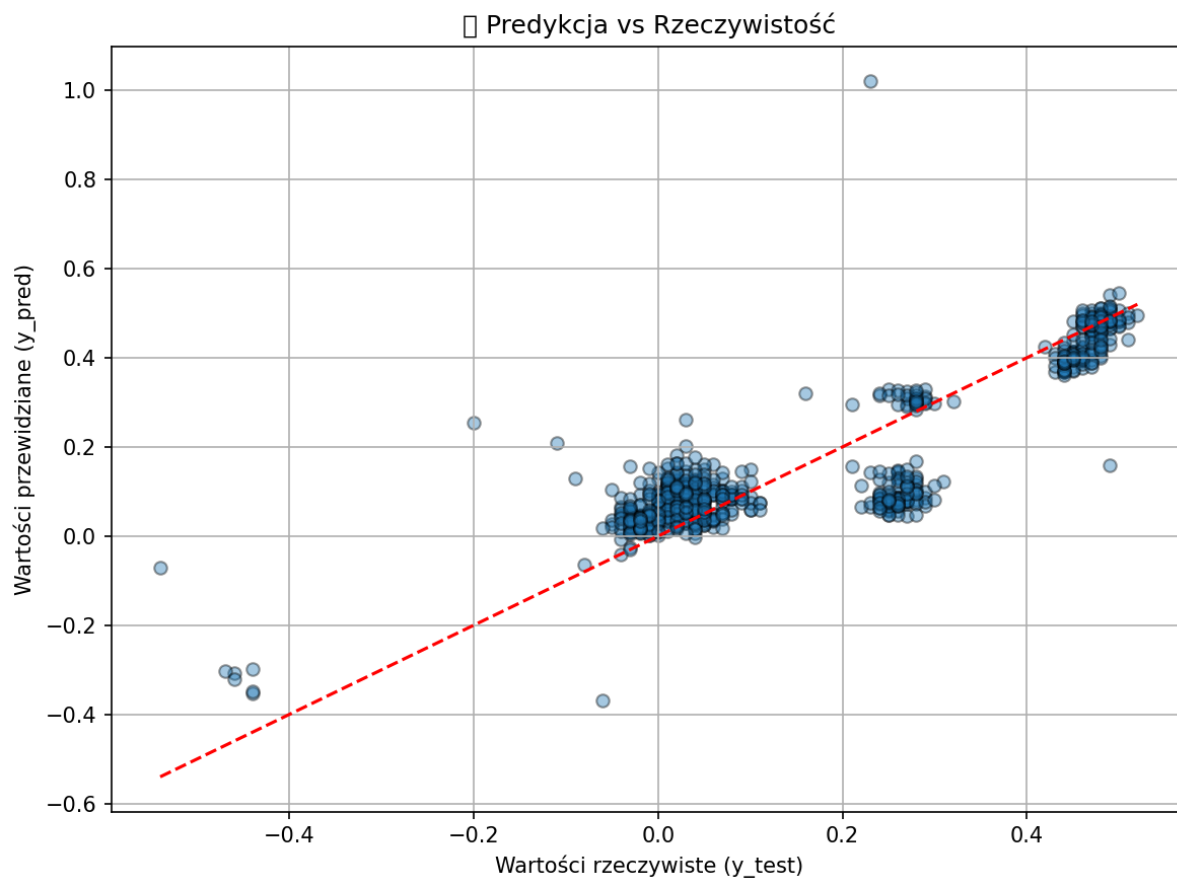
Interpretacja wyników:

Model RandomForest okazuje się mniej skuteczny z uwagi na wartość błędu MSE (większa niż w przypadku LGBM Regression) skuteczny jak model LGBM Regression, jednakże ze względu na czas trenowania zyskuje przewagę nad LGBM, gdyż czas trenowania w przypadku LGBM wynosi 2,51 sekundy co stanowi najdłuższy czas w porównaniu do pozostałych algorytmów odznaczających się znacznie krótszym czasem trenowania.

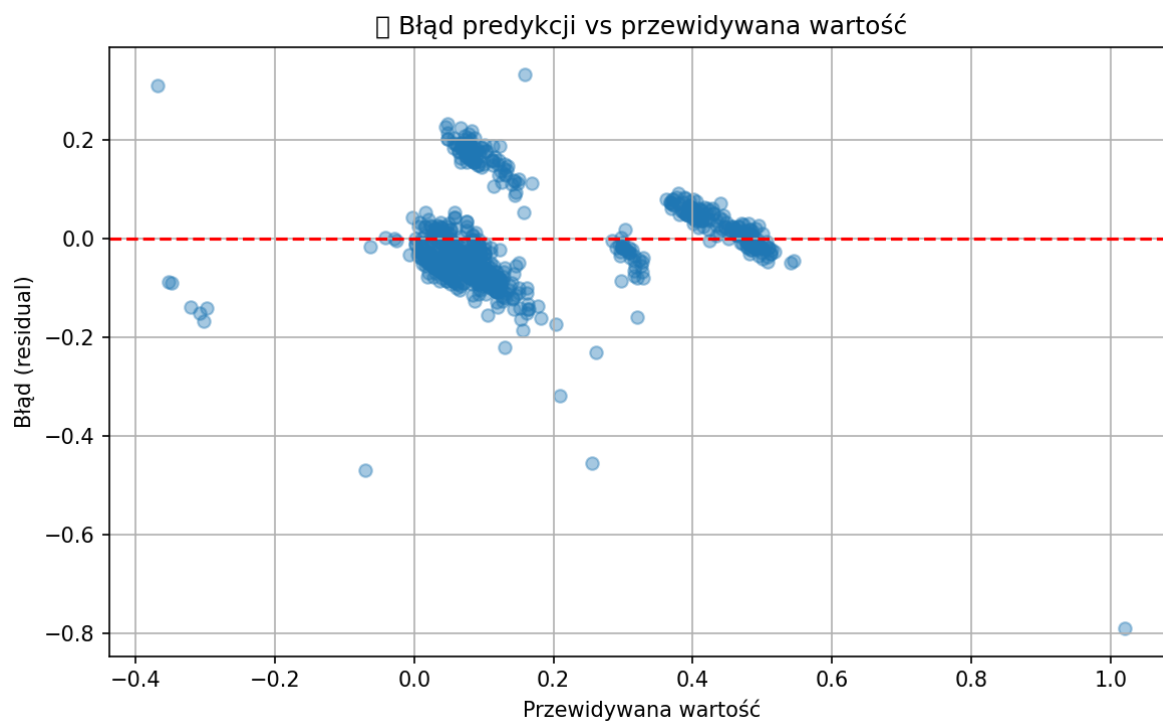
4. Wyniki skuteczności do tworzenia modelu Linear Regression

```
Model gotowy. Czas trenowania: 0.03 sekundy
MSE: 0.0089
R2 Score: 0.7747
Zapisuję model i scaler do Neon...
```

Rysunek 19. Wyniki dla modelu utworzonego za pomocą Linear Regression



Rysunek 20. Predykcja na tle rzeczywistych wyników dla Linear Regression



Rysunek 21. Wykres błędu predykcji dla algorytmu Linear Regression

Interpretacja wyników:

W porównaniu z wykresami dla pozostałych modeli (rys 14 – 18) model wytworzony na bazie algorytmu Linear Regression, ma najbardziej odstający błąd oraz wartości przewidywane mają liczne punkty, które nie przecinają się z osią czerwoną symbolizującą wartość przewidywaną. Dodatkowo wartości wyników na poziomie MSE 0,0089 oraz R2 równy 0,77 wskazują na wyniki zdecydowanie mniej skuteczne niż algorytmy RandomForest czy LGBM Regression. Jedyną niepodważalną zaletą algorytmu Linear Regression jest bardzo krótki czas trenowania, który wyniósł 0,03 sekundy.

Najlepszym algorytmem jaki można zastosować do przewidywania pomiarów fizycznych, które zarazem mogą być nieidealne, gdyż zależne są od wielu czynników – w tym: co chwila podatnych na zmianę warunków atmosferycznych, eksploatacji sprzętu pomiarowego, przerw w usługach sieciowych lub znacznym spowolnieniu działania mobilnego transferu danych (hotspot) – jest algorytm LGBM Regression lub GradientBoostingRegressor. Dla danych gromadzonych w czasie rzeczywistym, które mają być trenowane przez aplikację internetową został wybrany algorytm Gradient Boosting Regressor. Wybór ten wynika z uwagi na bardzo szybki czas trenowania 0,47 sekundy oraz optymalne parametry, w tym: niski błąd i dobry wskaźnik R2 widocznym w tabeli 7. Pozostałe modele również zostały nauczone wykorzystując algorytm Gradient Boosting Regression. Wyniki dla modeli przewidujących przyszłe wskazania prądu ładowania, prądu wyjściowego oraz temperatury, natężenia oświetlenia i wilgotności zostały zamieszczone odpowiednio na rys.22,23,24,25,26,27,28 i 29 oraz tabeli podsumowującej wyniki pt. "Podsumowanie wyników badań wyboru algorytmu "

Tabela 7. Podsumowanie wyników badań wyboru algorytmu

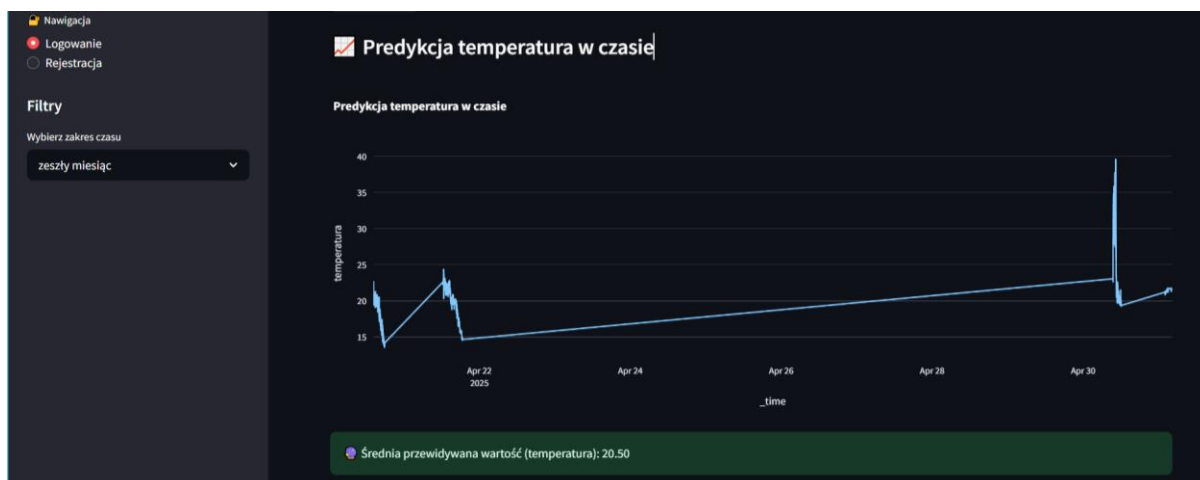
Algorytm	MSE	R2	czas trenowania
Gradient Boost Regressor	0,0029	0,9266	0,47 sekundy
LGBM Regressor	0,0025	0,9370	2,51 sekundy
Random Forest Regression	0,0027	0,9324	1,52 sekundy
Linear Regression	0,0089	0,7747	0,03 sekundy



Rysunek 22. Widok z poziomu aplikacji wykresu predykcji dla poziomu oświetlenia



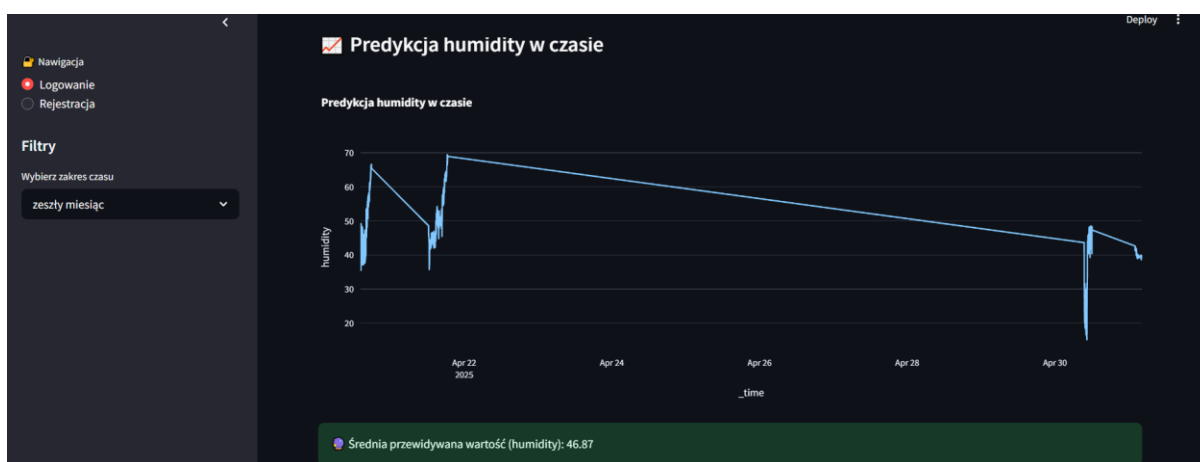
Rysunek 23. Statystyki predykcji dla oświetlenia



Rysunek 24. Wykres predykcji z poziomu aplikacji dla predykcji temperatury



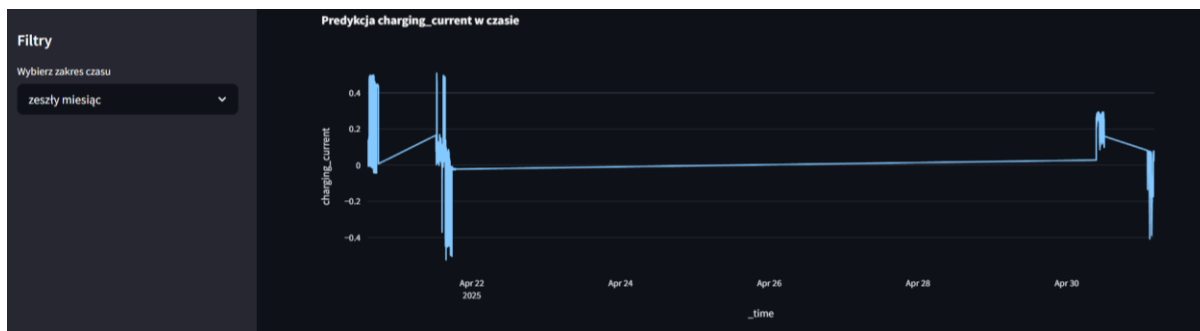
Rysunek 25. Statystyki predykcji



Rysunek 26. Predykcja wilgotności w czasie



Rysunek 27. Statystyki predykcji dla wilgotności



Rysunek 28. Predykcja prądu ładowania



Rysunek 29. Statystyki predykcji dla prądu ładowania

5.3 Wykorzystanie agentów AI do inteligentnego sterowania

Modele LLM takie jak ChatGPT służą nie tylko do czatowania z ludźmi, wyjaśniając im konkretne tematy lub wspomagając ich pisanie kontentu, czy to marketingowego, czy formalnego, są przede wszystkim doskonałym przykładem w nowoczesnych rozwiązaniach oraz rozwojem wizji interakcji człowiek – komputer. W niniejszej pracy modele LLM zostały przedstawione jako pośrednicy w podejmowaniu decyzji skupionych i dostosowanych do konkretnego problemu, a konkretnie do autonomicznego podejmowania decyzji w sterowaniu magazynem energii. W tym celu, w realizowanym projekcie, zostały wdrożone następujące modele LLM:

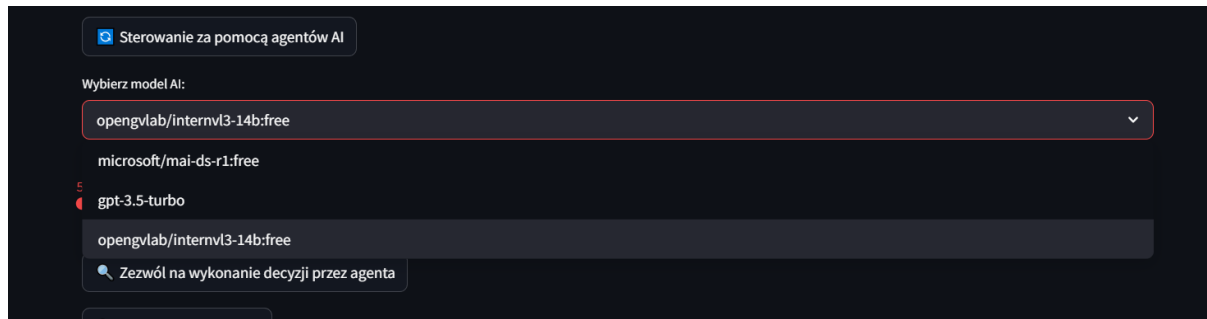
- microsoft/mai-ds-r1:free;
 - gpt-3.5-turbo;
 - opengvlab/internvl3-14b:free;
- oraz zastosowano API od openRouter (rys.30).

Powyższe modele LLM zostały wybrane ponieważ cechują się niezawodnością oraz niskim wskaźnikiem „temperature” (jest to wskaźnik, który cechuje tzw. twórczość modelu LLM im jest on wyższy tym bardziej „poetyckie” odpowiedzi model zaczyna pisać, jako parametr ustawiony. Parametr niskiego temperature sprawia, że modele te są bardzo konkretne w podejmowaniu decyzji oraz ściśle i zwięźle formułują wypowiedzi uzasadniające podjętą decyzję. Optymalna wartość parametru temperatury – celem dostrojenia takiego modelu winna wynosić 0,2.

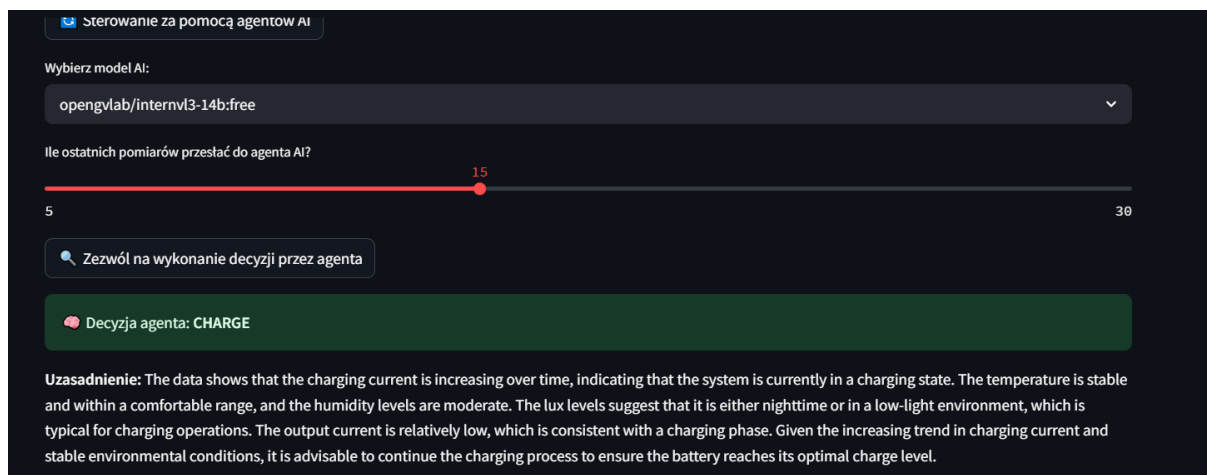
Powyższe modele zostały nastrojone przez fine-tuning aby ich podejmowana decyzja miała jedną z 3 możliwych odpowiedzi: CHARGE, DISCHARGE, OFF, co przedstawiono na rys.11.

Modele LLM zostały uczone za pomocą szybkiej metody few-shot-learningu, różni się ona tym od klasycznego fine-tuningu, tym że do LLM przekazywany jest wyspecjalizowany prompt jako polecenie, a nie jak w przypadku fine-tuningu zbiór testowy zawierający konkretne cechy i decyzje podjęte dla tych cech. Dla powyższych przypadków dostarczono do prompta

ilość ostatnich wartości zebranych z bazy, którą użytkownik może wybrać sobie z poziomu aplikacji z zakresu od 5 do 30 ostatnich sczytanych pomiarów. Przekazywana wartość ostatnich pomiarów sprawia, że agent AI operuje na najświeższych wynikach pomiarów, minimalizując przypadkowy błąd użytkownika spowodowany jego nieznaną działaniem całej aplikacji lub nieznaną trenowania agentów AI. Agenci AI, tuż po naciśnięciu przez użytkownika przycisku “Zezwól na wykonanie decyzji przez agenta”, uruchamiają proces wysterylizowania sygnału z poziomu aplikacji do mikrokontrolera, patrz rys. 30 i 31.

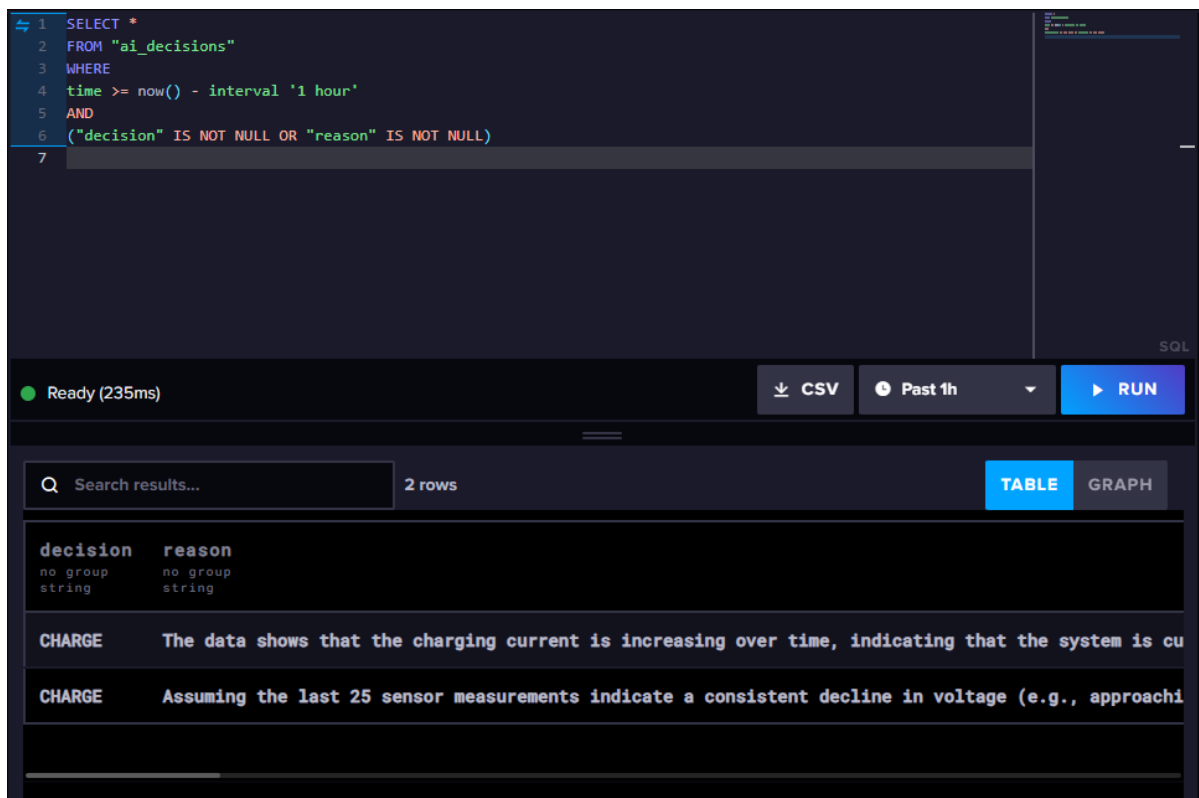


Rysunek 30. Zastosowane modele LLM do budowy agentów



Rysunek 31. Odpowiedź agenta LLM oraz podjęcie decyzji

Podgląd wyników decyzji podjętych przez agentów AI, które zostały wysterylizowane do bazy danych czasu rzeczywistego InfluxDB widnieje na rys.32.



Rysunek 32. Podgląd wyników gromadzonych w bazie chmurowej InfluxDB Cloud

Ponadto w aplikacji jest możliwość samodzielnego sterowania przez użytkownika wybierając opcję "Sam wykonaj decyzję". Wybierając tę opcję użytkownik, niezależnie od decyzji podjętej przez agenta AI, może zmienić tę decyzję w czasie rzeczywistym iysterować odpowiednie wyjścia mikrokontrolera na "Charge", "Discharge" lub "Off", jak przedstawiono na rys.33.



Rysunek 33. Panel wyboru decyzji przez użytkownika

5.4 Proces sterowania

Proces sterowania opiera się na dwóch mikrokontrolerach z rodziny Esspressif: ESP32 DOIT DEVKIT V1. Przebieg sterowania polega na nadawaniu danych z sensorów do bazy danych oraz późniejszym ich odbieraniu z bazy. Pobranie danych z bazy następuje przez prywatne API, dzięki czemu z poziomu mikrokontrolera można wysłać zbierane dane przez czujniki. Autoryzacja wykorzystuje parametry wygenerowane przy rejestracji w bazie chmurowej takie jak: prywatny token dostępu, endpoint do bazy czyli inaczej DATABASE URL, ORG oraz BUCKET.

Drugi mikrokontroler służy do odbierania danych z bazy danych, również poprzez API wykorzystując te same dane z tego samego endpointu, poprzez wykonanie odpowiednich zapytań do bazy w języku zbliżonym do SQL (patrz przykład poniżej, w pkt 4), jednak nie kwalifikowanym formalnie jako język SQL[22].

Proces sterowania jaki zachodzi między agentem sztucznej inteligencji, a mikrokontrolerem składa się z następujących etapów:

1. Agent AI podejmuje decyzje czy włączyć panel, celem naładowania akumulatora tzn. czy w danej chwili będzie to korzystne, czy też okaże się, że z punktu widzenia aktualnie panujących warunków atmosferycznych nie jest to efektywne pod względem energetycznym. Agent ten podejmuje decyzje na podstawie danych przesłanych do LLM z ostatnich 5 do 30 pomiarów zależnie od wyboru (patrz rys.31).
2. Decyzja jest wysyłana do bazy danych w chmurze InfluxDB Cloud (rys.32).
3. W zależności od decyzji, którą podjęło AI – system magazynu energii ma się dostosować. Gdy zostanie zdecydowana wartość 'CHARGE' w magazynie energii powinno dojść do wysterowania stanu wysokiego dla styków załączających przez mikrokontroler przekaźnik, który został zamieszczonym na przewodzie dostarczającym prąd z panelu do kontrolera paneli. Natomiast wartość decyzji 'DISCHARGE' powoduje wyłączenie tego przekaźnika. Opcja 'OFF' powoduje wyłączenie obu przekaźników, zarówno łączącego zasilanie z panelu do kontrolera panelu fotowoltaicznego, jak i przekaźnika łączącego obwód akumulatora z kontrolerem ładowania.
4. Sterowanie zachodzi poprzez komunikacje REST API drugiego mikrokontrolera z tą samą bazą danych, lecz odmiennie zdefiniowanych opcjach field oraz measurements. Przykład zapytania do danych InfluxDB, w którym zdefiniowano bazę z pola bucket o nazwie SENSORS, z measurement=ai_decision z field=decision, jak to zamieszczono na rys.35 i rys 36.
5. Następnie zapytanie: zostało wplecione do programu wgrywanego do mikrokontrolera, zapytanie, które zwróci decyzje ma następującą postać:

```
"from(bucket: \"\" + String(INFLUXDB_BUCKET) + "\") "  
" |> range(start: -1h) "  
" |> filter(fn: (r) => r._measurement == \"ai_decisions\") "  
" |> filter(fn: (r) => r._field == \"decision\") "  
" |> last()";
```

jest to typowy język używany w InfluxDB pod nazwą flux, wykazuje on cechy znacznego podobieństwa do zwykłego SQL.[22]

Następnie jeśli dana decyzja zostaje odczytana jako "CHARGE" i znaki zostają porównane, mikrokontroler wysteruje odpowiednie wyjście na stan wysoki. Przykładowy kod, który po odczycie z bazy danych 'CHARGE' zapala diodę został zamieszczony w poniższym listingu:

```
#include <Arduino.h>  
#include <WiFi.h>  
#include <HTTPClient.h>
```

```

#define WIFI_SSID "Livebox-49E8"
#define WIFI_PASSWORD "dane wrzliwe nie ujawnione "
#define INFLUXDB_URL "dane wrazliwie nie ujawnione"
#define INFLUXDB_TOKEN "dane wrazliwie nie ujawnione "
#define INFLUXDB_ORG "dane wrazliwie nie ujawnione "
#define INFLUXDB_BUCKET "SENSOR"

String influxQueryURL = String(INFLUXDB_URL) +
"/api/v2/query?org=" + INFLUXDB_ORG;
String influxQuery =
  "from(bucket: \"" + String(INFLUXDB_BUCKET) + "\")"
  " |> range(start: -1h)"
  " |> filter(fn: (r) => r._measurement == \"ai_decisions\")"
  " |> filter(fn: (r) => r._field == \"decision\")"
  " |> last()";

// Parsowanie wartości tekstowej decyzji
String parseDecision(const String& lastLine) {
  int commaCount = 0;
  int startIndex = 0;

  for (int i = 0; i < lastLine.length(); i++) {
    if (lastLine.charAt(i) == ',') {
      commaCount++;
      if (commaCount == 6) {
        startIndex = i + 1;
        int endIndex = lastLine.indexOf(',', startIndex);
        if (endIndex != -1) {
          return lastLine.substring(startIndex, endIndex);
        }
      }
    }
  }
  return "";
}

void setup() {
  Serial.begin(9600);
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, LOW);

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nConnected to WiFi");
}

void loop() {
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
    http.begin(influxQueryURL);
    http.addHeader("Authorization", "Token " +
String(INFLUXDB_TOKEN));
    http.addHeader("Content-Type", "application/vnd.flux");

    int httpResponseCode = http.POST(influxQuery);

```

```

if (httpResponseCode > 0) {
    String payload = http.getString();
    Serial.println("Odpowiedź InfluxDB:");
    Serial.println(payload);

    int lastIndex = payload.lastIndexOf("\n");
    while (lastIndex > 0) {
        String lastLine = payload.substring(lastIndex + 1);
        lastLine.trim();

        if (lastLine.length() > 0) {
            Serial.println("Ostatnia linia: " + lastLine);
            String decision = parseDecision(lastLine);
            Serial.print("Odczytana decyzja: ");
            Serial.println(decision);

            if (decision == "CHARGE") {
                digitalWrite(LED_BUILTIN, HIGH);
                Serial.println("Dioda włączona (decyzja: CHARGE).");
            } else {
                digitalWrite(LED_BUILTIN, LOW);
                Serial.println("Dioda wyłączona (decyzja inna niż
CHARGE).");
            }
            break;
        }

        payload = payload.substring(0, lastIndex);
        lastIndex = payload.lastIndexOf("\n");
    }

    if (lastIndex <= 0) {
        Serial.println("Nie znaleziono danych w odpowiedzi.");
    }
    else {
        Serial.print("Błąd HTTP: ");
        Serial.println(httpResponseCode);
    }

    http.end();
}

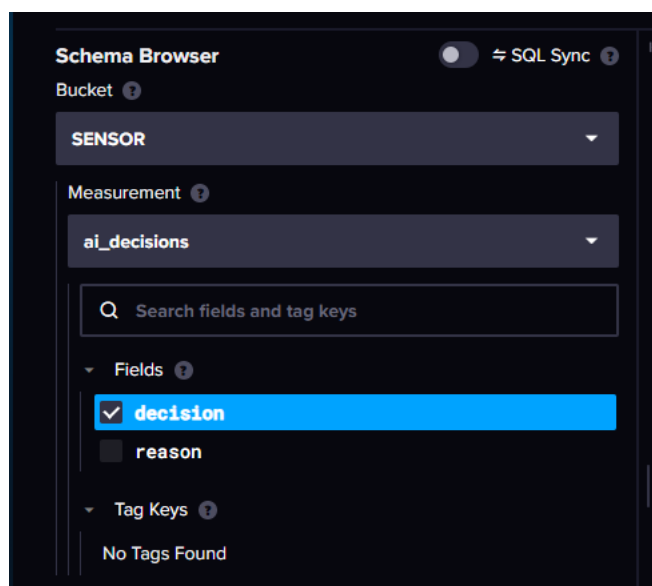
delay(10000); // odczyt co 10 sekund
}

```

Kod powyżej jest tylko przykładem sterowania jednym z wyjść mikrokontrolera, na podstawie decyzji przesłanych przez agenta AI do bazy w chmurze. oprogramowanie aplikacji zarówno od strony hardwarowej jak i software'owej – zostało zamieszczone w linkach do Github dostarczonych w załącznikach do pracy.

decision	reason
no group	no group
string	string
CHARGE	The charging current is negative, indicating that the device is currently discharging. However, th
CHARGE	The charging current is negative, indicating that the device is currently discharging. However, th
CHARGE	The charging current is negative, indicating that the device is currently discharging. However, th

Rysunek 34. Decyzje wysyłane przez agenta AI do bazy



Rysunek 35. Szczegółowe parametry bazy danych

Schema Browser

Bucket: SENSOR

Measurement: ai_decisions

Fields: ☒ decision, ☐ reason

Tag Keys: No Tags Found

```

1 SELECT *
2 FROM "ai_decisions"
3 WHERE
4   time >= now() - interval '30 days'
5

```

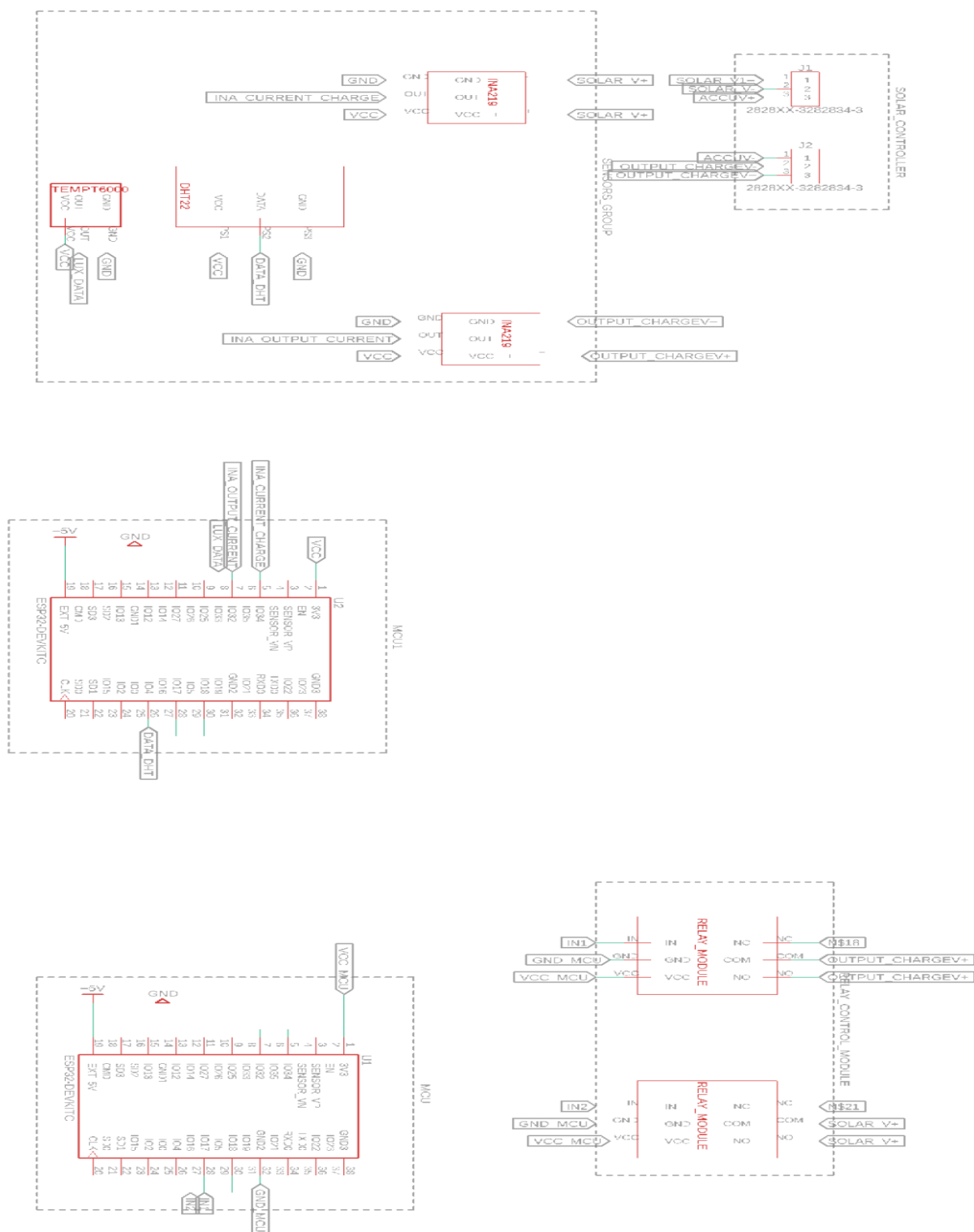
Ready (410ms) CSV Past 30d RUN

Search results... 60 rows

decision	reason
no group	no group
string	string
OFF	The light intensity is consistently high, suggesting that natural light is sufficient, and there i
OFF	The charging current is consistently low (0.06-0.08) and the device is not being charged, and ther
OFF	The light intensity is consistently low and the charging current is not significant, suggesting th

Rysunek 36. Widok zapytania do bazy i wyników

Schemat elektryczny podłączenia mikrokontrolerów i całego systemu pomiarowego do magazynu energii przedstawiono na rys. 37.



Rysunek 37. Schemat części pomiarowej i sterującej do prototypu magazynu energii

Spis komponentów elektronicznych uwzględnionych w schemacie znajduje się w tabeli poniżej (tabela 8)

Tabela 8. Komponenty użyte w projekcie

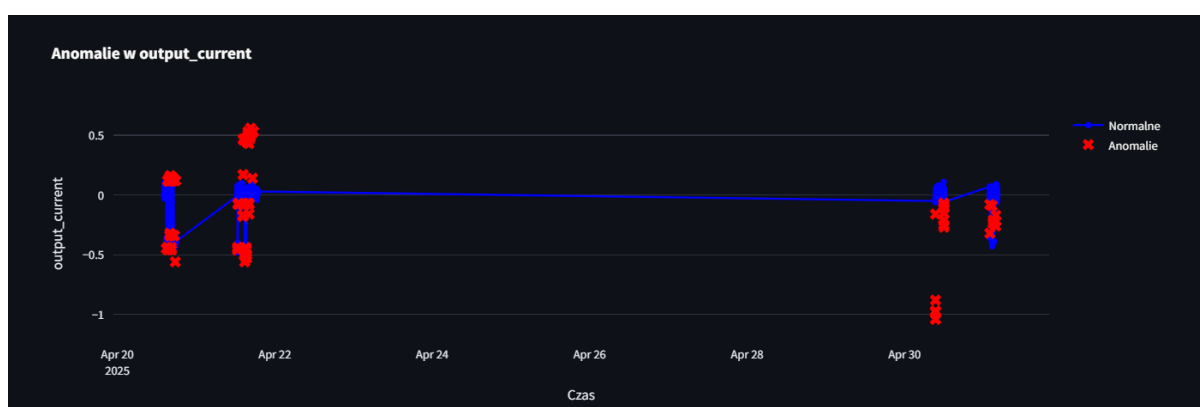
Ref	Part Value	Device	Footprint	Description	Category	Manufacturer	MPN	Operating Temp
J1	2828XX-3282834-3	2828XX-3282834-3	TERMBLK_254-3N	3 Position Wire to Board Terminal Block Horizontal with Board	Fixed Terminal Blocks 3P TERMINAL BLOCK	TE Connectivity AMP Connectors	282834-3	-40°C ~ 105°C
J2	2828XX-3282834-3	2828XX-3282834-3	TERMBLK_254-3N	3 Position Wire to Board Terminal Block Horizontal with Board	Fixed Terminal Blocks 3P TERMINAL BLOCK	TE Connectivity AMP Connectors	282834-3	-40°C ~ 105°C
US1	DHT22	DHT22	DHT22-FOOTPRINT	nie dotyczy	nie dotyczy	nie dotyczy	nie dotyczy	nie dotyczy
US2	TEMPT6000	TEMPT6000	TEMPT6000-FOOTPRINT	nie dotyczy	nie dotyczy	nie dotyczy	nie dotyczy	nie dotyczy
US3	INA219-DEVICE	INA219-DEVICE	INA219-FOOTPRINT	nie dotyczy	nie dotyczy	nie dotyczy	nie dotyczy	nie dotyczy
US4	INA219-DEVICE	INA219-DEVICE	INA219-FOOTPRINT	nie dotyczy	nie dotyczy	nie dotyczy	nie dotyczy	nie dotyczy
US6	RELAY_MODULE	RELAY_MODULE	RELAY_MODULE-FOOTPRINT	nie dotyczy	nie dotyczy	nie dotyczy	nie dotyczy	nie dotyczy
US7	RELAY_MODULE	RELAY_MODULE	RELAY_MODULE-FOOTPRINT	nie dotyczy	nie dotyczy	nie dotyczy	nie dotyczy	nie dotyczy
U1	ESP32-DEVKITC	ESP32-DEVKITC	MODULE_ESP32-DEVKITC	nie dotyczy	nie dotyczy	ESPRESSIF	N/A	nie dotyczy
U2	ESP32-DEVKITC	ESP32-DEVKITC	MODULE_ESP32-DEVKITC	nie dotyczy	nie dotyczy	ESPRESSIF	N/A	nie dotyczy

6. Analiza anomalii w zbieranych danych z sensorów

Anomalie w danych zbieranych z sensorów w czasie rzeczywistym mogą wynikać z różnych czynników, niekoniecznie takich, które wskazywały by na celowe przejęcie systemu przez osoby trzecie. Często anomalie wynikają z:

- z jakości podłączania i rozłączania się obwodu zasilającego panele lub akumulator,
- zakłócenia w transferze danych przez udostępniany hotspot, wynikające ze słabego połączenia internetowego,
- zbyt luźne podłączenie sensorów do mikrokontrolera (prototyp w trakcie badań był podłączany do płytek stykowych, nie zaś przylutowany na stałe do drukowanych),
- innych zakłóceń w linii zasilającej układ – urządzenia funkcjonujące w domu, gdzie uruchomiono stanowisko badawcze,
- wystąpienia tzw. zjawisk eksploatacyjnych (np. automatyczne dołączanie/odłączanie innych odbiorów w układzie, w celu doraźnych testów).

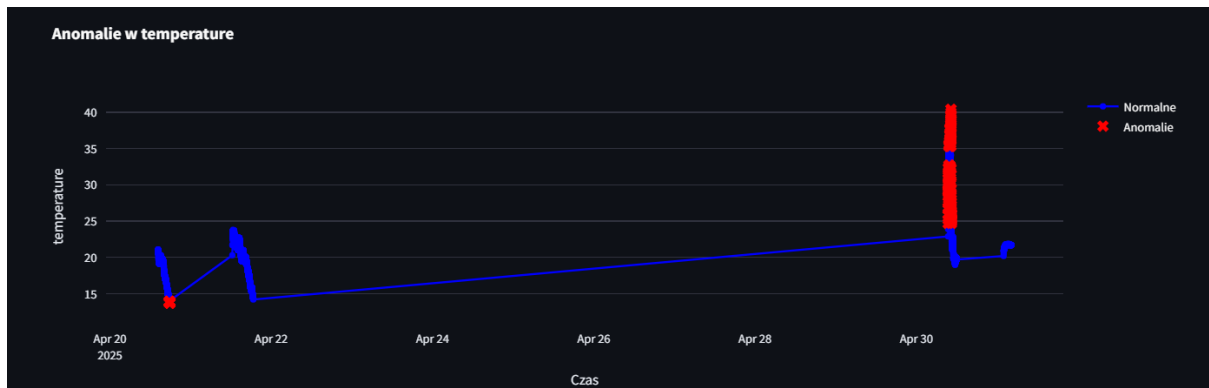
Anomalie w mierzonych wartościach były widoczne w każdej z mierzonych cech, zaznaczono je na wykresach kolorem czerwonym, odpowiednio na rys.40, 41, 42.



Rysunek 39. Anomalie dla prądu wyjściowego



Rysunek 40. Anomalie dla mierzonego poziomu oświetlenia



Rysunek 41. Anomalie w temperaturze



Rysunek 42. Anomalie w wilgotności

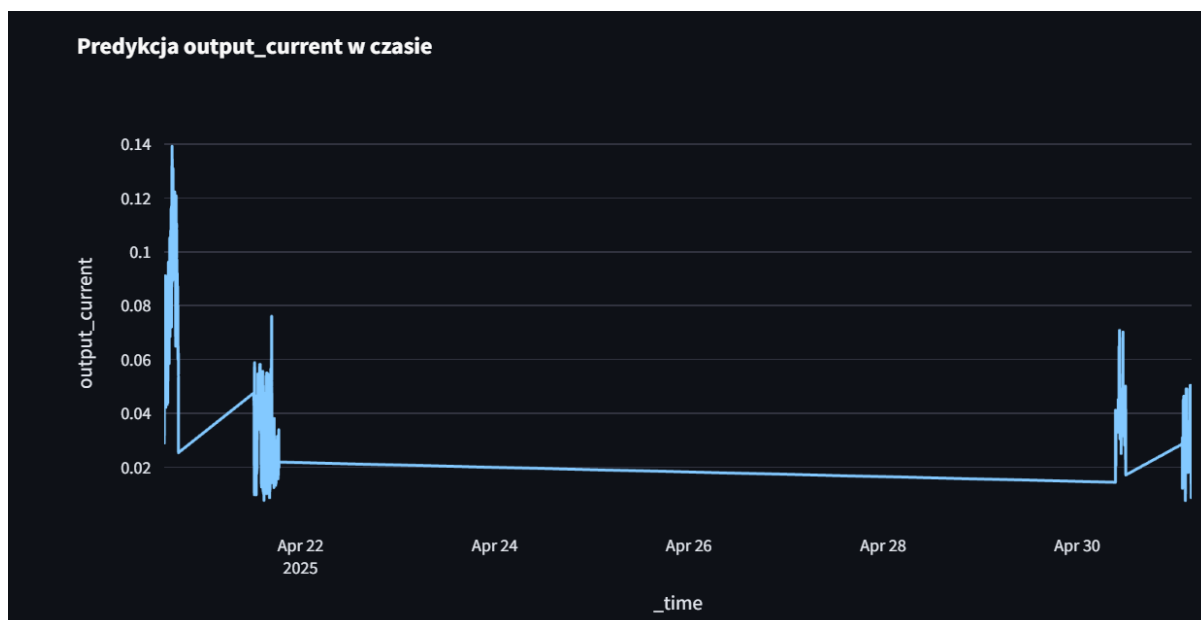


Rysunek 43. Anomalie w prądzie ładowania

Najwięcej anomalii jest widocznych w prądzie ładowania (rys 43) i wyjściowym (rys 39)

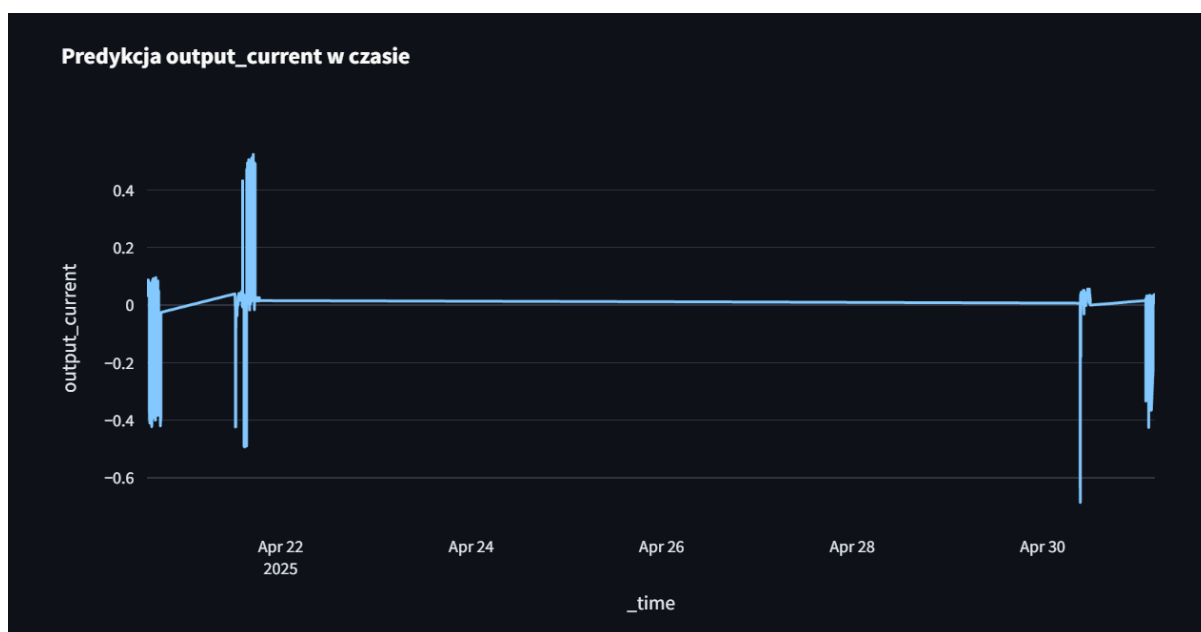
Rozwiązania jakie zastosowano żeby wykluczyć anomalie przekazywane do zbioru treningowego to zastosowanie algorytmów, które wykluczają w trenowaniu anomalie, nawet jeśli występują one w większym stopniu jak w przypadku rejestrowanego prądu wyjściowego i prądu ładowania.

Model oczyszczający anomalie w predykcji zawarty jest jako wynik przewidywania przedstawiony na wykresie na rysunkach poniżej (tj. od rys.44 do rys.46)

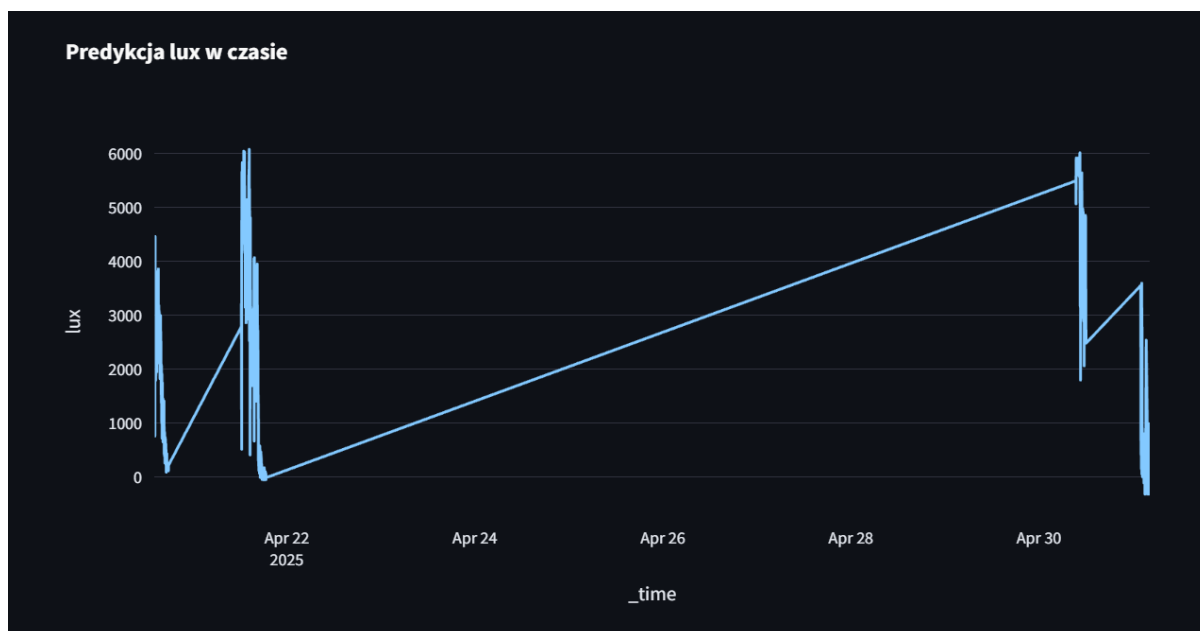


Rysunek 44. Predykcja prądu wyjściowego w czasie

Ten sam model bez usuwania anomalii w predykcji:



Rysunek 45. Predykcja z anomaliami



Rysunek 46. Predykcja oświetlenia oczyszczona z anomalii

6.1. Metody detekcji i klasyfikacji anomalii

Metody jakie zostały użyte aby oczyścić dane z anomalii to:

- usuwanie wartości ujemnych (w przypadku rejestracji sygnału prądu),
- odcięcie wartości odstających (statystyka 3σ – trzy odchylenia standardowe),
- izolacja anomalii algorytmem Isolation Forest,
- usuwanie wartości powyżej progów w sposób ręczny.

6.2. Badania anomalii braku wartości i ich redukcja

W wyniku porównania modeli uczących się na danych z anomaliami i modeli uczonych na danych oczyszczonych widoczne są istotne różnice w predykcji wybranych charakterystyk dla obu tych przypadków. Główne zarejestrowane różnice to [m.in.](#): brak ujemnych wartości przy predykcji prądu oraz bardziej realne wartości predykcyjne, bez zawyżonych wartości powyżej wskaźnika odchylenia standardowego od pozostałych wyników normy. Reasumując modele, które uczyły się na danych bez anomalii przetworzone przez ponowne dane dawały wynik predykcji bez anomalii, natomiast modele uczone na anomaliami dawały wyniki nie precyzyjne i z widocznymi anomaliami w predykcji “odziedziczonymi” po modelu.

Pozostałą kwestią, oprócz detekcji anomalii i ich oczyszczanie przed trenowaniem, była redukcja wartości zerowych czyli redukcja przedziałów, w których wybrana wartość nie była mierzona. Problem ten odnosi się wyłącznie do monitorowania prądu, gdyż w trakcie monitorowania pomiaru zarejestrowano sytuacje braku podłączenia urządzeń na wyjściu. Wskutek czego, rejestrowany przez czujniki prąd, podawał wartości 0 przez określony czas. W przypadku pozostałych czujników takie problemy nie występowały, gdyż ich pomiar był

przewodzony w sposób ciągły. Zerowe wartości prądu w pomiarach mogą powodować zniekształcenie modelowanej predykcji wartości oczekiwanych tej wielkości.

Zastosowane rozwiązania:

- usuwanie wartości zerowych, świadczących o braku zasilania, poprzez filtrowanie i przyjmowanie wartości większych od zera,
- dodanie nowej cechy o nazwie `is_power_on`, a następnie uwzględnianie jej w predykcji – ten sposób model sam nauczy się, że gdy `is_power_on == 0`, inne wartości mogą być niezerowe,
- wprowadzenie modelu składającego się z dwóch faz: np. fazy pierwszej czyli klasyfikatora o aktywnym zasilaniu oraz fazy drugiej czyli regresora dla przypadków aktywnych.

6.3 Wnioski

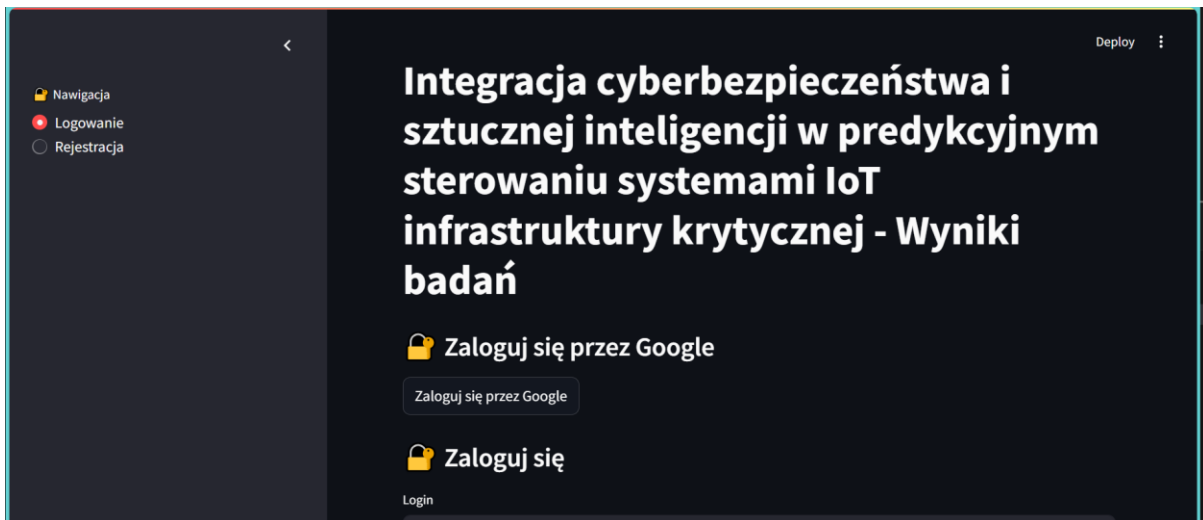
Po usunięciu zidentyfikowanych anomalii w danych – modele predycyjne działały w sposób, który charakteryzował się mniejszymi wartościami błędów predykcji oraz mniejszą ilością występujących w sygnale predykowanym anomalii lub całkowitym ich braku. Dodatkowym problemem, oprócz powstających anomalii w trakcie zbierania danych z czujników, były wartości zerowe rejestrowane w wyniku odłączenia obciążenia z akumulatora lub samym odłączeniu akumulatora, wartości zerowe prądu mogły wpływać na dalszą jakość predykcji. By zniwelować ten wpływ wykonano redukcje wartości zerowych w pomiarach. Najbardziej poprawnym sposobem okazało się zastosowanie podziału modelu na stany, w których osobno model rozróżnia proces ładowania tam gdzie są wartości większe od zera, jak i stan rozłączenia zasilania gdzie wartości są zerowe.

7. Analiza i testowanie aplikacji pod względem bezpieczeństwa

Równie ważnym aspektem jak poprawność pomiaru i wnioskowania z danych dokonywanego przez wyuczony model AI – jest kwestia bezpieczeństwa cyfrowego i jego zapewnienie w opracowywanym rozwiązaniu. W tym rozdziale przeprowadzone zostały badania w celu wykrycia ewentualnych zagrożeń i niedopracowań, które mogłyby przyczynić się do przejęcia opracowanej aplikacji przez niepożądane osoby. Zagrożenia takie jak: wyłudzenia danych użytkowników, wpływu na działanie systemu sterowania i innych potencjalnych zagrożeń przedstawiono w kolejnych podrozdziałach.

7.1. Testowanie ataków lokalnie

Aplikacja podzielona została na backend wykonany we frameworku Django oraz frontend, który został natomiast wykonany w frameworku StreamLite. Widok aplikacji uruchomionej lokalnie przedstawia zrzut na rys.47. Aplikacja umożliwia proces rejestracji nowych użytkowników, jak i logowania się przez konto Google oraz zwykłe logowanie, nie połączone z profilem Google.



Rysunek 47. Widok powitalny aplikacji

Na testy na poziomie lokalnym składały się:

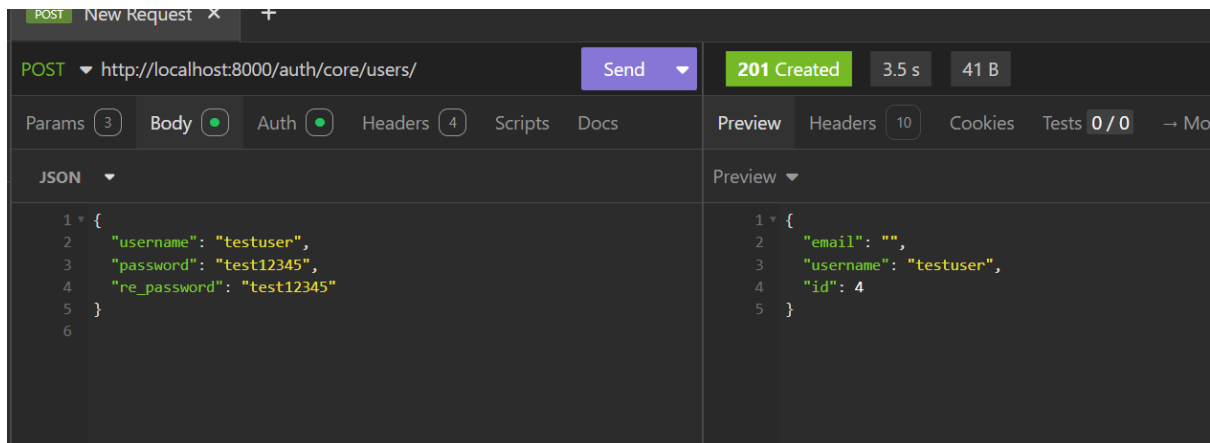
1. testy JWT i autoryzacji,
 2. testy rejestracji i logowania,
 3. testy logowania Google OAuth2,
 4. testy Cross-Site-Scripting,
 5. testy CSRF,
 6. staki brutal-force,
 7. sprawdzenie endpointów i routingu,
- poniżej przedstawiono ich przebieg i wyniki.

7.1.1. Przebieg testów JWT i autoryzacji

Celowość testu: Sprawdzenie czy token JWT działa poprawnie, a nieautoryzowane żądania są blokowane, czy podrobione tokeny są odrzucane oraz czy aplikacja daje dostęp do danych tylko i wyłącznie użytkownikom zautoryzowanym?

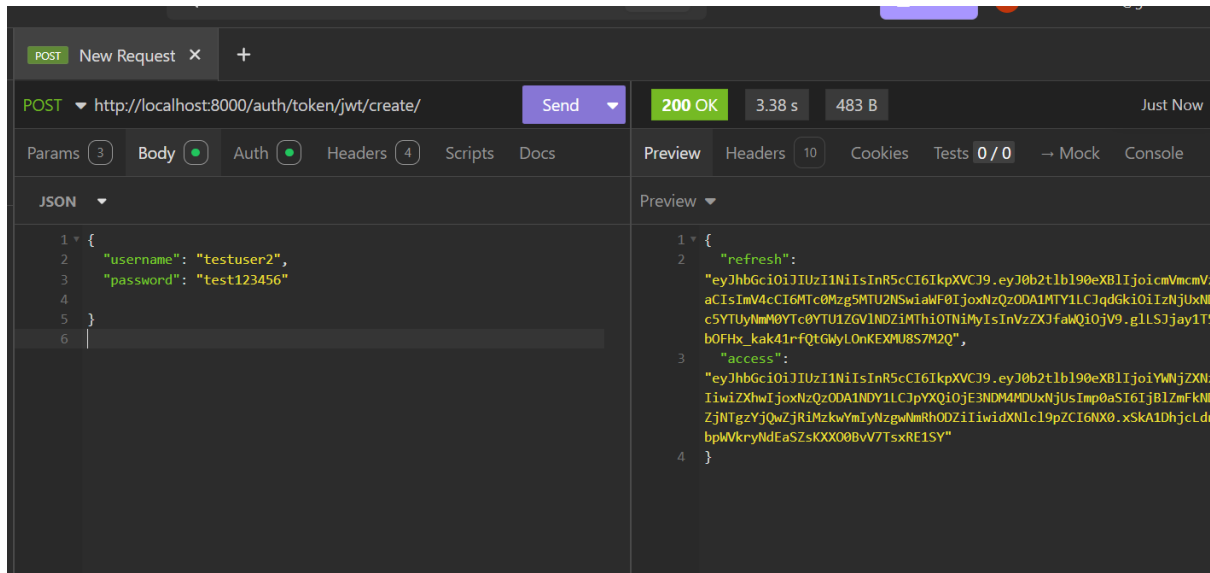
Proces przebiegu testów

1. Sprawdzenie endpointów umożliwiających rejestrację przykładowego użytkownika:



Rysunek 48. Testowanie endpointu rejestracji

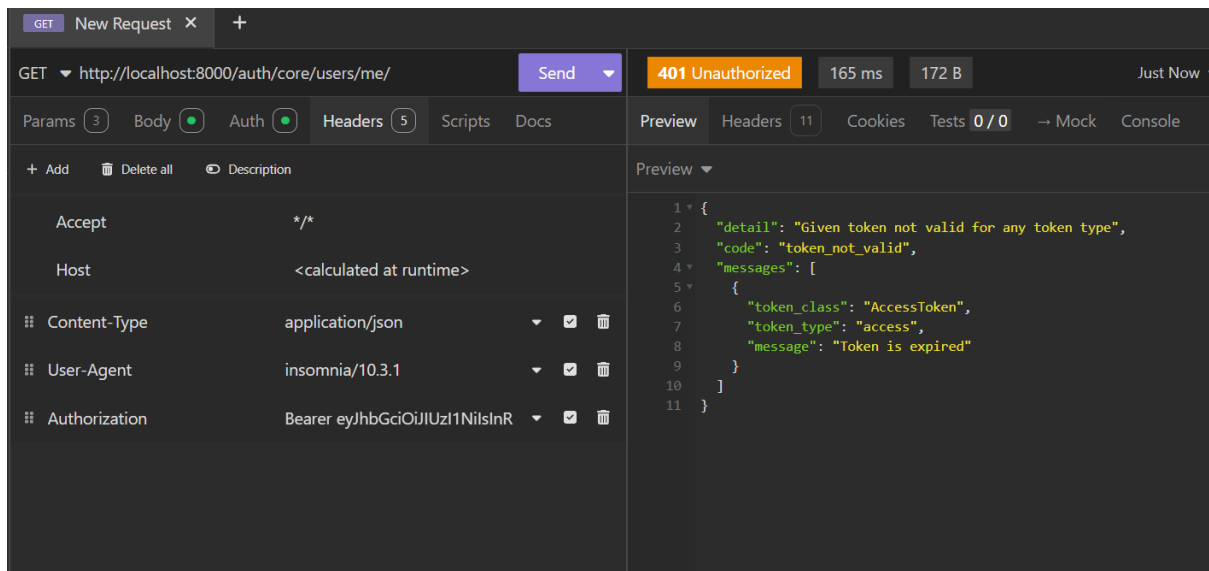
Sprawdzenie poprawności logowania:



Rysunek 49. Zwracanie tokenu

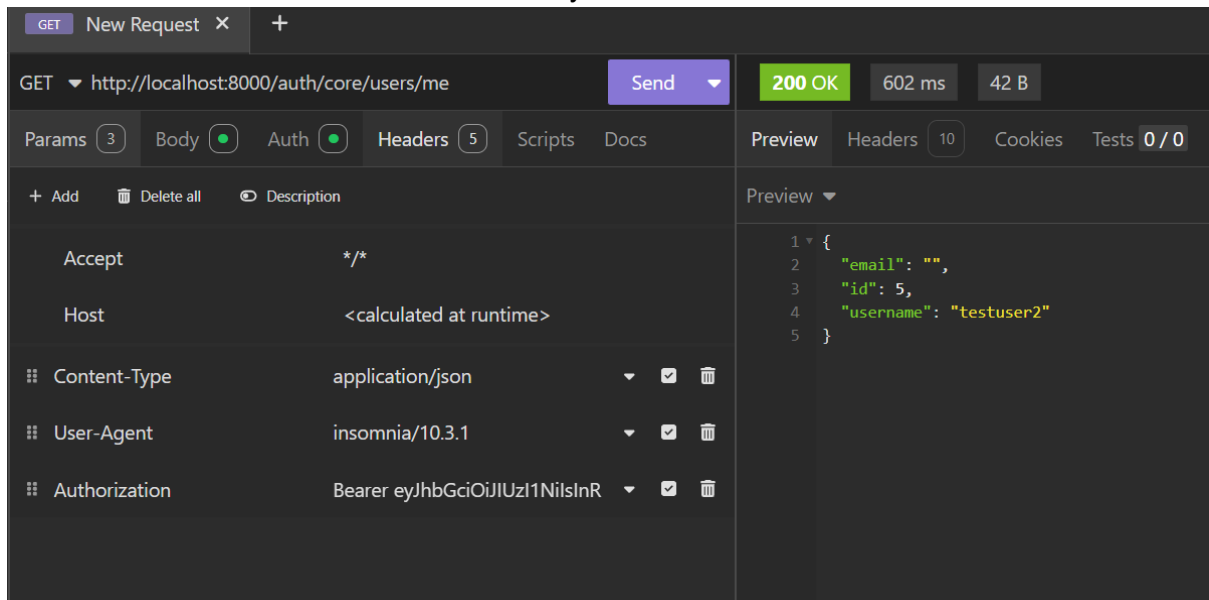
endpoint POST <http://localhost:8000/auth/token/jwt/create/> działa poprawnie.

Dostęp do chronionego tokenu (dla endpointu zwracającego dane aktualnie zalogowanego użytkownika):



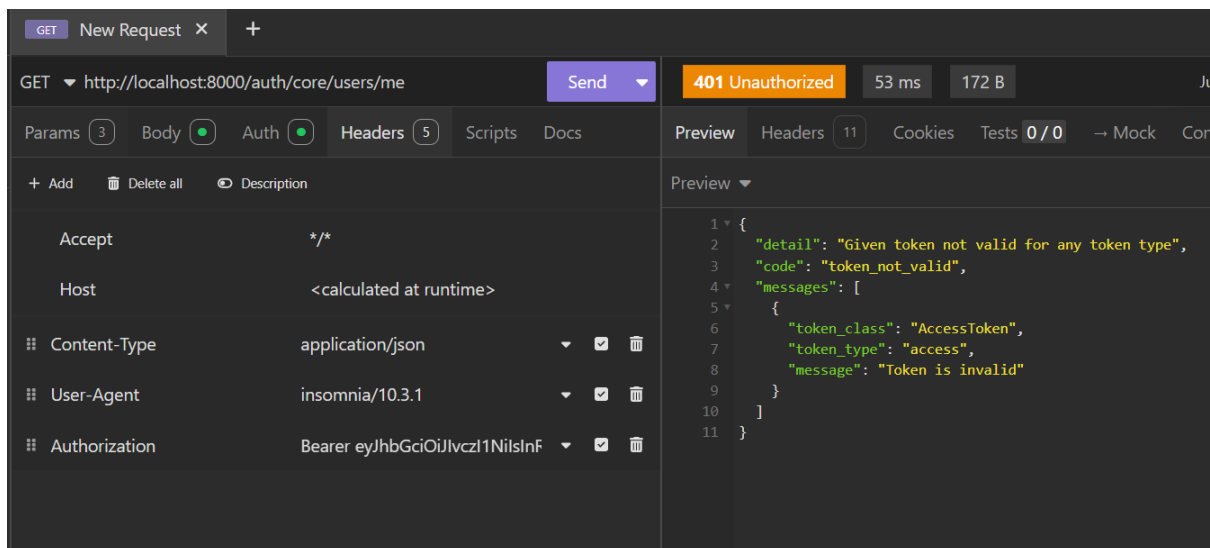
Rysunek 50. Widok dostępu do chronionego tokenu

Po ponownym zalogowaniu się celem uzyskania świeżego nowego tokena autoryzacja do chronionego endpointu zwracającego dane o użytkowniku przebiegła pomyślnie i dane zostały zwrócone:



Rysunek 51. Przebieg testów autoryzacji

Dalsza część testów opierała się na podawaniu fałszywego/podrabianego tokenu:



Rysunek 52. Dalsza część testów

7.1.2. Testy bezpieczeństwa przed atakami Cross-Site-Scripting

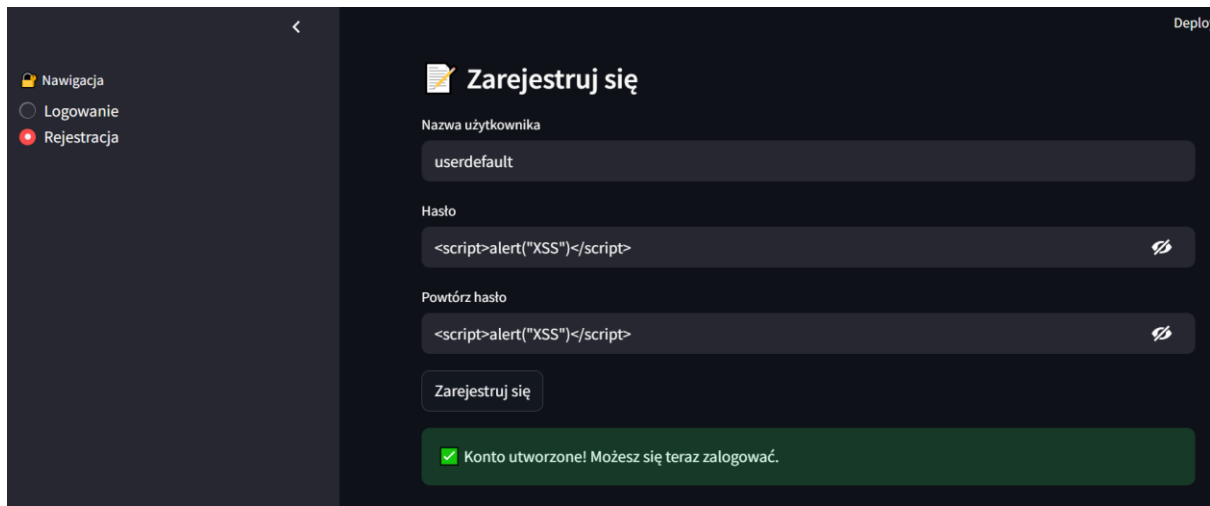
Celowość wykonania tego testu: Sprawdzenie, czy aplikacja zabezpiecza się przed złośliwym kodem JS (np. `<script>`)?

Przebieg testu: Wpisywanie złożonych złośliwych segmentów kodu przez formularz aplikacji, widoczny jak na zrzucie:



Rysunek 53. Przebieg testów wstrzykiwania złośliwego kodu

Wynik testu:

The screenshot shows a web application interface with a dark theme. On the left is a sidebar with navigation links: 'Nawigacja' (active), 'Logowanie', and 'Rejestracja'. The main content area is titled 'Zarejestruj się'. It contains three input fields: 'Nazwa użytkownika' with the value 'userdefault', 'Hasło' with the value '<script>alert("XSS")</script>', and 'Powtórz hasło' with the same value. Below these fields is a 'Zarejestruj się' button. At the bottom, a green message box with a checkmark icon states: 'Konto utworzone! Możesz się teraz zalogować.'

Rysunek 54. Wynik testu

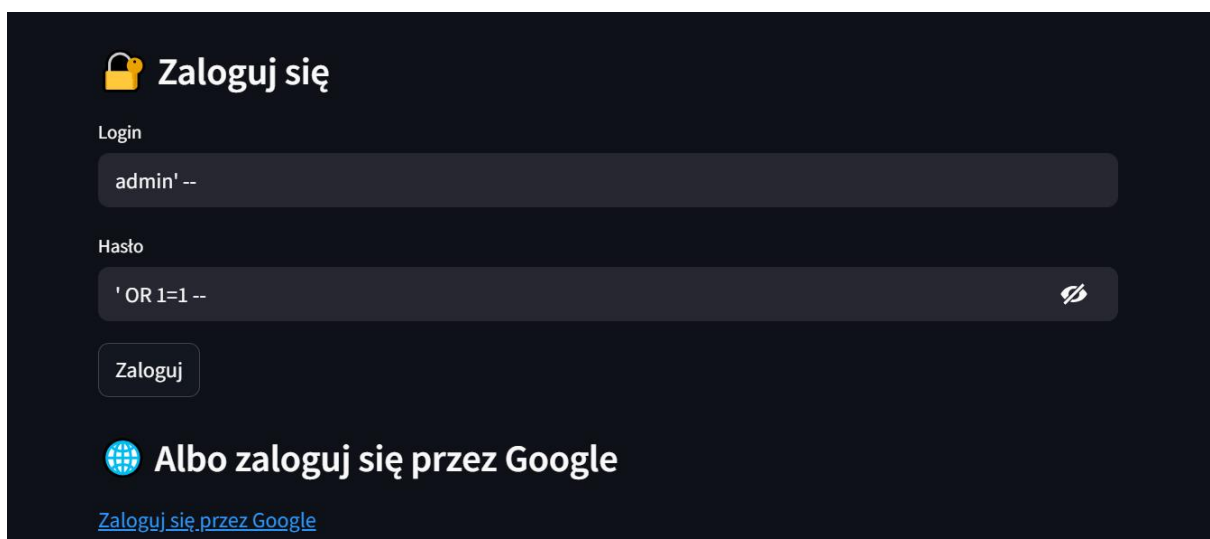
Wyniki testu potwierdzają odporność aplikacji

7.1.3. Testowanie na SQL Injection

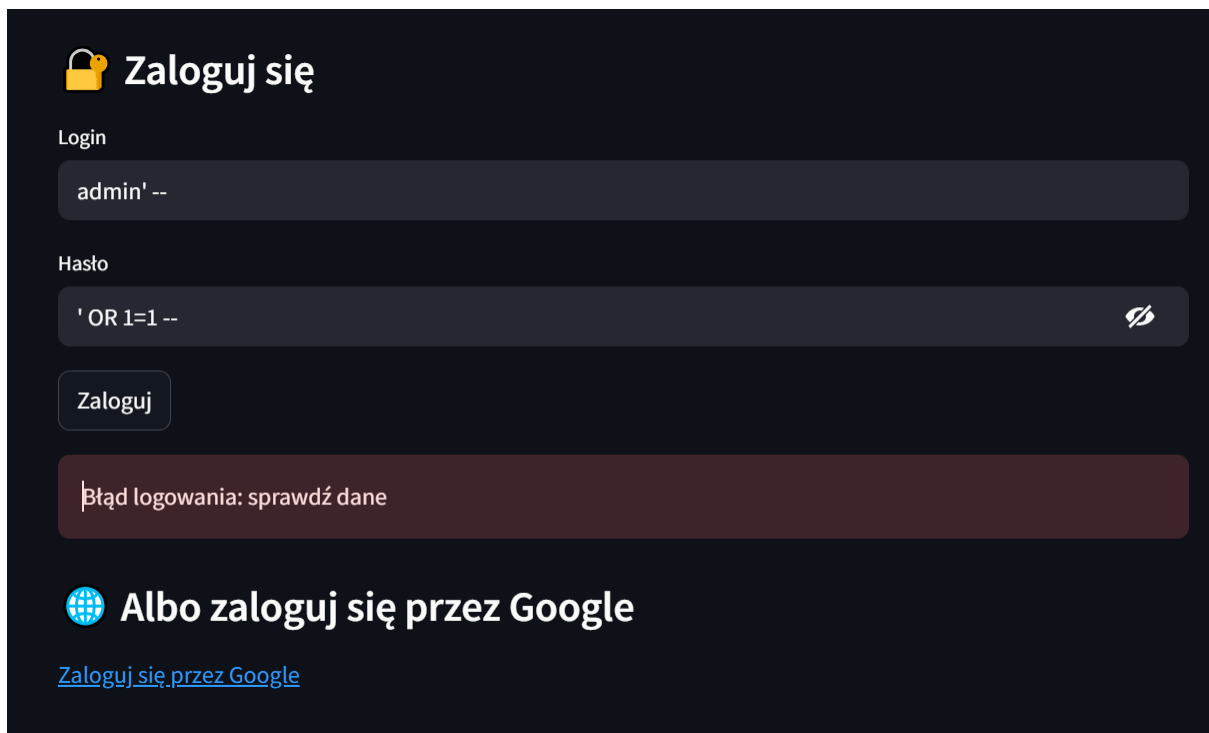
SQL Injection to rodzaj taktowania aplikacji webowych, który przejawia się poprzez wpisanie kwerend SQLowych poprzez formularze dostępne na stronie internetowej. W wyniku sukcesywnych prób tego rodzaju ataku użytkownik atakujący może uzyskać dostęp do danych i je manipulować. W przypadku aplikacji, której backend łączy się z zewnętrzną bazą chmurową.

Celowość wykonania testu: Sprawdzenie jak aplikacja radzi sobie z przykładowymi SQL Injection.

Przebieg testu: Widoczny na rys. 55

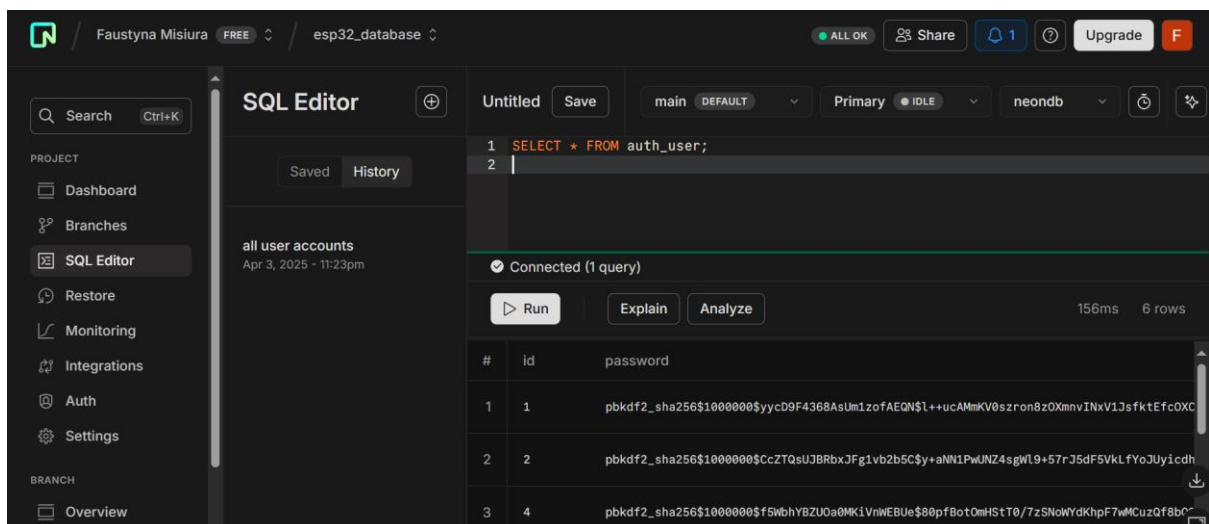
The screenshot shows a web application interface with a dark theme. The main content area is titled 'Zaloguj się'. It contains two input fields: 'Login' with the value 'admin' --' and 'Hasło' with the value '' OR 1=1 --'. Below these fields is a 'Zaloguj' button. At the bottom, there is a link to 'Albo zaloguj się przez Google' and a text link 'Zaloguj się przez Google'.

Rysunek 55. Przebieg testów wstrzykiwania złośliwych zapytań SQL



Rysunek 56. Przebieg testów

Przeprowadzono test bezpieczeństwa pod kątem ataków typu SQL Injection. Użyto klasycznego ciągu ' OR 1=1 --, który w przypadku braku zabezpieczeń pozwala na nieautoryzowany dostęp. Aplikacja prawidłowo odrzuciła dane logowania i nie dopuściła do nieautoryzowanego dostępu. Framework Django z zastosowaniem ORM skutecznie zabezpiecza przed SQL Injection.



Rysunek 57. Sprawdzenie haseł w bazie

Wynik sprawdzenia haseł w bazie (czy s,a zahashowane).

7.1.4. Testy rejestracji i logowania

Przebieg testów: Testy polegały na zalogowaniu się do aplikacji zarówno w wersji lokalnej jak i w wersji produkcyjnej, czas logowania/ rejestracji był dłuższy dla wersji lokalnej

wobec wersji działającej na zewnętrznym serwerze, wykluczono w testach [m.in](#) powolne działania transferu mobilnego.

Cel testów: Sprawdzenie procesu logowania i rejestracji oraz czasu przejścia do aplikacji po zalogowaniu/przejścia do panelu logowania po rejestracji

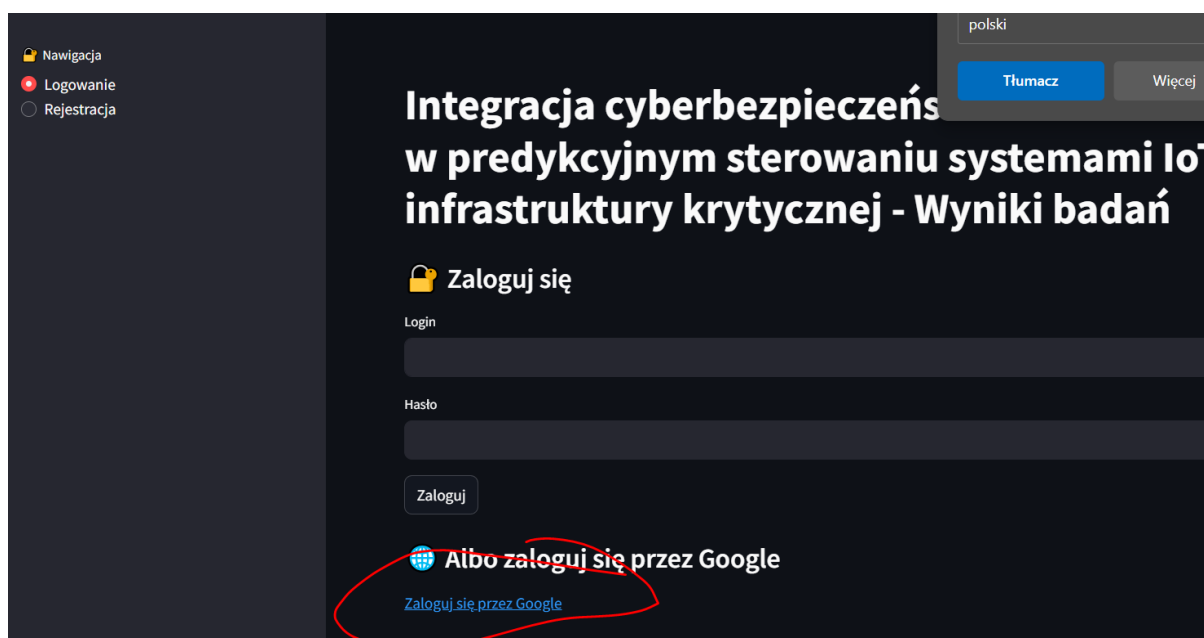
7.1.5. Testy logowania Google OAuth2

Cel testów: Celem testu było sprawdzenie poprawności podpiętej usługi od Google czyli Google Authorization, która umożliwia poprawne zalogowanie się nowym uczestnikom używając swojego konta Google, bez konieczności tworzenia konta w serwisie aplikacji.

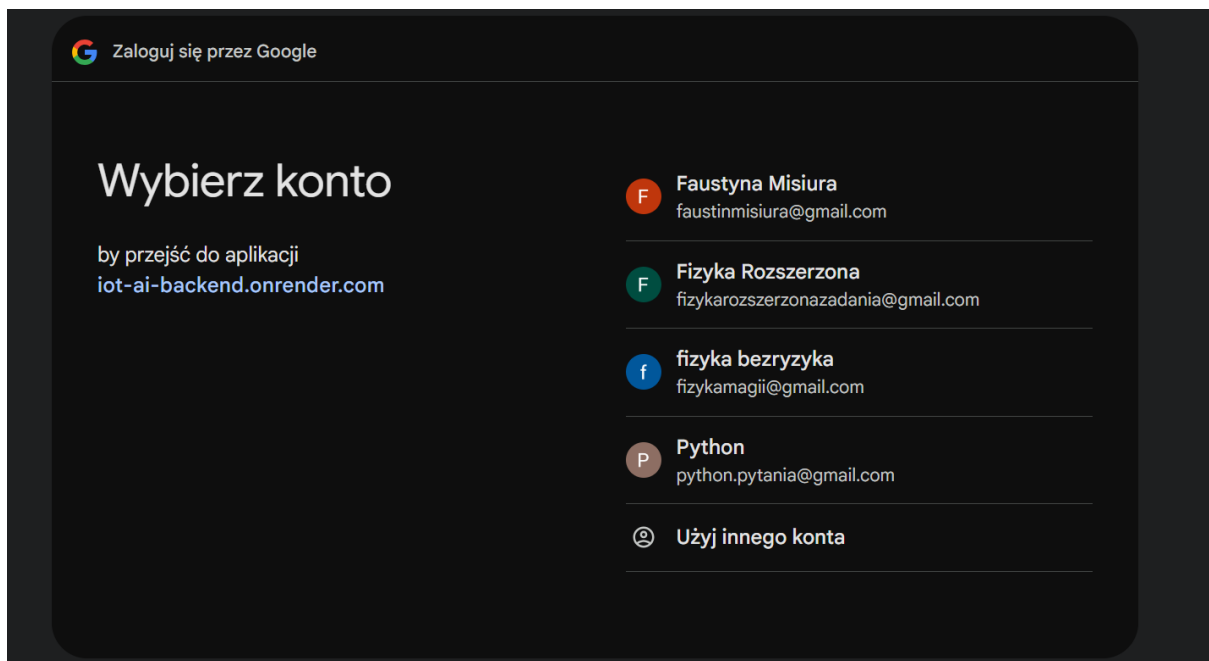
Przebieg testów: Testy polegały na monitorowaniu procesu logowania się. Proces ten powinien przebiegać w 3 krokach:

1. Wybór opcji "Logowanie przez google".
2. prawidłowe przekierowanie do kont google i wybór konta docelowego.
3. Ponowne przekierowanie na stronę już z uprawnieniami dostępu.

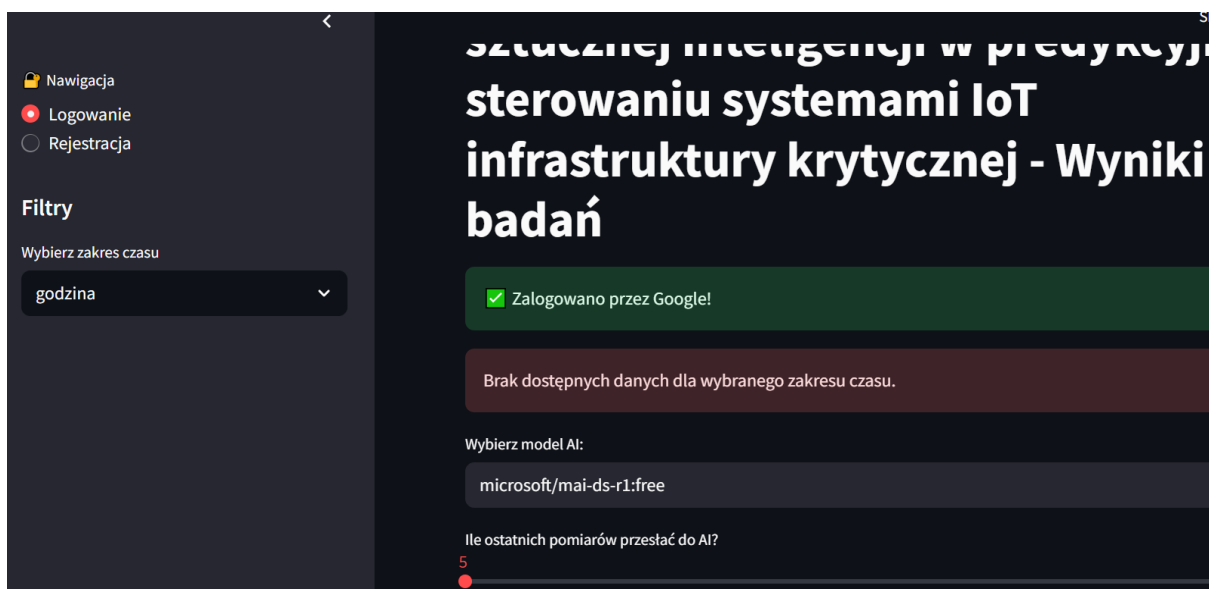
Powyższe kroki zostały przedstawione na rysunkach poniżej:



Rysunek 58. Sprawdzenie działania logowania przez Google



Rysunek 59. Widok panelu logowania przez Google



Rysunek 60. Widok po poprawnym zalogowaniu

Proces logowania oraz rejestracji przebiegł bezproblemowo.

7.2. Testowanie aplikacji w wersji produkcyjnej

Wstępne informacje o wersji produkcyjnej:

- adres URL frontendu: <https://fastinatechnology.streamlit.app>,
- adres serwera backendu: <https://iot-ai-backend.onrender.com>,
- baza danych : Neon (PostgreSQL),
- baza danych czasu rzeczywistego: InfluxDB Cloud.

Na testy aplikacji w wersji produkcyjnej składały się:

1. testy OWASP ZAP i testy API (inspekcja, fuzzing, brute-force tokenów,
2. testy bezpiecznego połączenia z backendem w renderze,
3. DoS – Denial of Service (atak na dostępność) sprawdzić, jak backend lub frontend radzi sobie z przeciążeniem lub wieloma żadaniami.
4. próba przejęcia danych z czujników (spoofing lub manipulacja), będzie polegać na stymulowaniu ataku polegający na przesłaniu fałszywych danych pomiarowych do InfluxDB – czyli przejęcie „czujnika”.

Wykorzystane narzędzie do przeprowadzenia testów to ZAP (Zed Attack Proxy) - jest to darmowe narzędzie do testowania aplikacji, którego twórca jest OWASP czyli Open Web Security Project). Oprogramowanie to wykonuje głównie testy penetracyjne, do których należą m.in.:

- SQL Injection,
- Cross-Site Scripting (XSS),
- Brute Force,
- luki w konfiguracji.

ZAP umożliwia nie tylko testowanie aplikacji lecz także tworzy raporty po każdym teście. W raportach tych znajduje się spis podatności jakie oprogramowanie ZAP było w stanie zarejestrować.

7.2.1. Testy przeprowadzone w programie OWASP ZAP

Do testów został użyty program OWA SP ZAP(v2.16.1), ZAP Version: 2.16.1. Test został przeprowadzony dla endpointu : <https://iot-ai-backend.onrender.com/auth/login/google-oauth2/?next=https://fastinatechnology.streamlit.app> – umożliwiające zalogowanie się do aplikacji za pośrednictwem konta Google.

Na przeprowadzone testy dla tego endpointu składały się: ataki XSS, CSRF, analiza nagłówków bezpieczeństwa.

Po przeprowadzonym skanowaniu, a późniejszych testach sporządzony został raport, którego fragment przedstawiono na rys.61

Summaries

Alert counts by risk and confidence

This table shows the number of alerts for each level of risk and confidence included in the report.

(The percentages in brackets represent the count as a percentage of the total number of alerts included in the report, rounded to one decimal place.)

		Confidence				
		User				
		Confirmed	Wysoki	Średni	Niski	Total
Risk	Wysoki	0 (0,0%)	0 (0,0%)	0 (0,0%)	0 (0,0%)	0 (0,0%)
	Średni	0 (0,0%)	6 (37,5%)	0 (0,0%)	1 (6,2%)	7 (43,8%)
	Niski	0 (0,0%)	2 (12,5%)	2 (12,5%)	1 (6,2%)	5 (31,2%)
	Informacyjny	0 (0,0%)	1 (6,2%)	2 (12,5%)	1 (6,2%)	4 (25,0%)
	Total	0 (0,0%)	9 (56,2%)	4 (25,0%)	3 (18,8%)	16 (100%)

Rysunek 61. Wyniki raportu (podane w procentach zagrożenia określonego typu)

W powyższym zestawieniu na rys.61, na całkowity raport składał się podział według następujących ogólnych klasyfikatorów zagrożeń: czyli zagrożenia o stopniu niskim, średnim informacyjne zagrożenia. Przy każdym zestawieniu została podana liczba wykrytych problemów. Problemy te podawane w procentach zostały wyszczególnione na rys.62

Alert counts by alert type

This table shows the number of alerts of each alert type, together with the alert type's risk level.

(The percentages in brackets represent each count as a percentage, rounded to one decimal place, of the total number of alerts included in this report.)

Alert type	Risk	Count
Absence of Anti-CSRF Tokens	Średni	1 (6,2%)
CSP: Failure to Define Directive with No Fallback	Średni	3 (18,8%)
CSP: Wildcard Directive	Średni	3 (18,8%)
CSP: script-src unsafe-eval	Średni	1 (6,2%)
CSP: script-src unsafe-inline	Średni	2 (12,5%)
CSP: style-src unsafe-inline	Średni	3 (18,8%)
Content Security Policy (CSP) Header Not Set	Średni	2 (12,5%)
CSP: Notices	Niski	3 (18,8%)

Rysunek 62. Widok wyników badań

Najistotniejsze Wyniki z raportu zostały przedstawione według powyższego podziału:

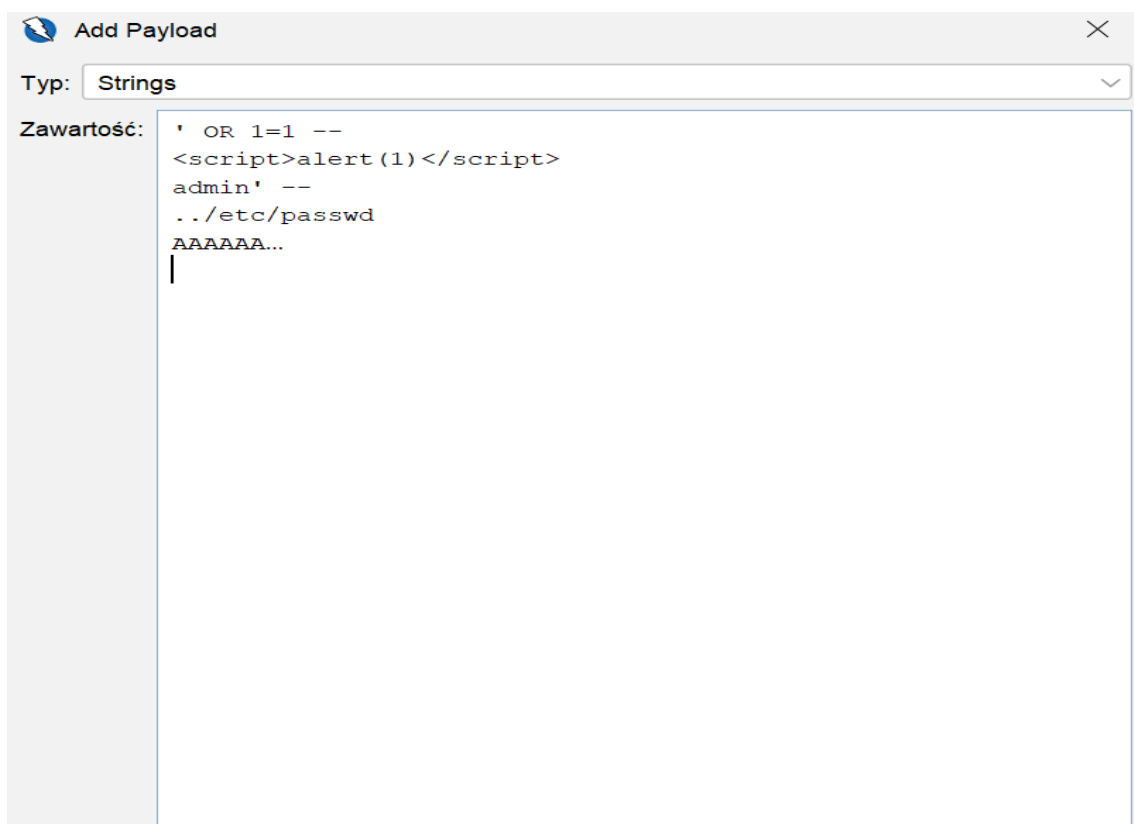
- Zagrożenia sklasyfikowane jako średni poziom ryzyka to:
 1. **Absence of Anti-CSRF Tokens** – oznacza to, że testy nie wykryły w aplikacji tokenów CSRF, w skutek czego aplikacja może wykazywać podatność na ataki typu Cross-Site-Request Forgery, które polegają m.in na tym, że użytkownik, który jest zalogowany i ma aktywną sesję na danej stronie, to aplikacja może wymusić na użytkowniku wykonanie nieświadomego żądania np. zmiana hasła poprzez formularz wyglądający na stworzony do innego celu.
 2. **CSP (Content Security Policy) errors** – występowanie błędów związanych z Content Security Policy takich jak: unsafe-inline czy brak CSP.
- Zagrożenia o niskim stopniu ryzyka:
 1. **Cookie Without Secure Flag**: oznacza, że istnieje ryzyko, że cookie mogą zostać przesłane w sposób zaszyfrowany.
 2. **Strict-Transport-Security Header Not Set** – brak wymuszenia HSTS, w wyniku tego może być zwiększone ryzyko ataków na aplikacje typu downgrade/mixcontent

Rysunek 64. Widok z oprogramowania do przeprowadzania testów aplikacji.

7.2.2. Testy fuzz endpointu create

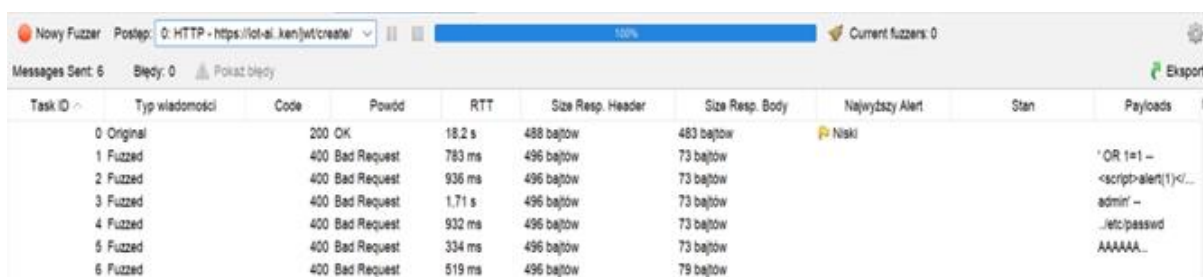
Testy fuzz na endpointzie będą polegały na “fuzzowaniu” username z password. Fuzzowanie polega na testach endpointa backendowego próbując przesłać do niego fałszywe dane i sprawdzić jak na nie reaguje. Do fuzzowania można zaliczyć ataki brutal force na hasła, XSS oraz SQL Injection. Testom fuzzowania poddany zostanie endpoint:

<https://iot-ai-backend.onrender.com/auth/token/jwt/create/>.



Rysunek 65. Widok wprowadzonych fałszywych payloadów do testów

lista payloadów zadeklarowanych w Zap’ie znajduje się na rys.65 i rys.68.



Task ID	Typ wiadomości	Code	Powód	RTT	Size Resp. Header	Size Resp. Body	Najwyższy Alert	Stan	Payloads
0	Original	200 OK		18.2 s	488 bajtów	483 bajtów	Niski		
1	Fuzzed	400 Bad Request		783 ms	496 bajtów	73 bajtów			' OR 1=1 --
2	Fuzzed	400 Bad Request		936 ms	496 bajtów	73 bajtów			<script>alert(1)</script>
3	Fuzzed	400 Bad Request		1.71 s	496 bajtów	73 bajtów			admin' --
4	Fuzzed	400 Bad Request		932 ms	496 bajtów	73 bajtów			../etc/passwd
5	Fuzzed	400 Bad Request		334 ms	496 bajtów	73 bajtów			AAAAAA...
6	Fuzzed	400 Bad Request		519 ms	496 bajtów	79 bajtów			

Rysunek 66. Wyniki badań fuzzowania

wyniki badań fuzzowania loginu i hasła payloadami ze zdjęcia powyżej.

Messages Sent: 5482		Błędy: 0		Pokaż błędy		Eksport			
Task ID ^	Typ wiadomości	Code	Powód	RTT	Size Resp. Header	Size Resp. Body	Najwyższy Alert	Stan	Payloads
4 827 Fuzzed		400	Bad Request	649 ms	496 bajtów	73 bajtów			teacher_lesson_...
4 828 Fuzzed		400	Bad Request	651 ms	496 bajtów	73 bajtów			GCSEs_revision_...
4 829 Fuzzed		400	Bad Request	444 ms	496 bajtów	73 bajtów			TrainingAndWork...
4 830 Fuzzed		400	Bad Request	121 ms	496 bajtów	73 bajtów			TrainingAndWork...
4 831 Fuzzed		400	Bad Request	180 ms	496 bajtów	73 bajtów			computing
4 832 Fuzzed		400	Bad Request	319 ms	496 bajtów	73 bajtów			ie
4 833 Fuzzed		400	Bad Request	367 ms	496 bajtów	73 bajtów			FinancialSupport...
4 834 Fuzzed		400	Bad Request	741 ms	496 bajtów	73 bajtów			FinancialSupport...
4 835 Fuzzed		400	Bad Request	301 ms	496 bajtów	74 bajtów			40556000
4 836 Fuzzed		400	Bad Request	601 ms	496 bajtów	73 bajtów			DisabledPeople
4 837 Fuzzed		400	Bad Request	614 ms	496 bajtów	73 bajtów			EducationAndTra...
4 838 Fuzzed		400	Bad Request	302 ms	496 bajtów	73 bajtów			FurtherEducation

Rysunek 67. Wyniki badań dalsza część

Wszystkie fuzzed requests dostały 400 Bad Request. To oznacza, że backend dobrze odrzuca niepoprawne / podejrzane payloady.

7.2.3. Testy fuzz endpointu users

Kolejnym etapem było wykonanie fuzzingu z fałszywymi payloadami dla enpointu:

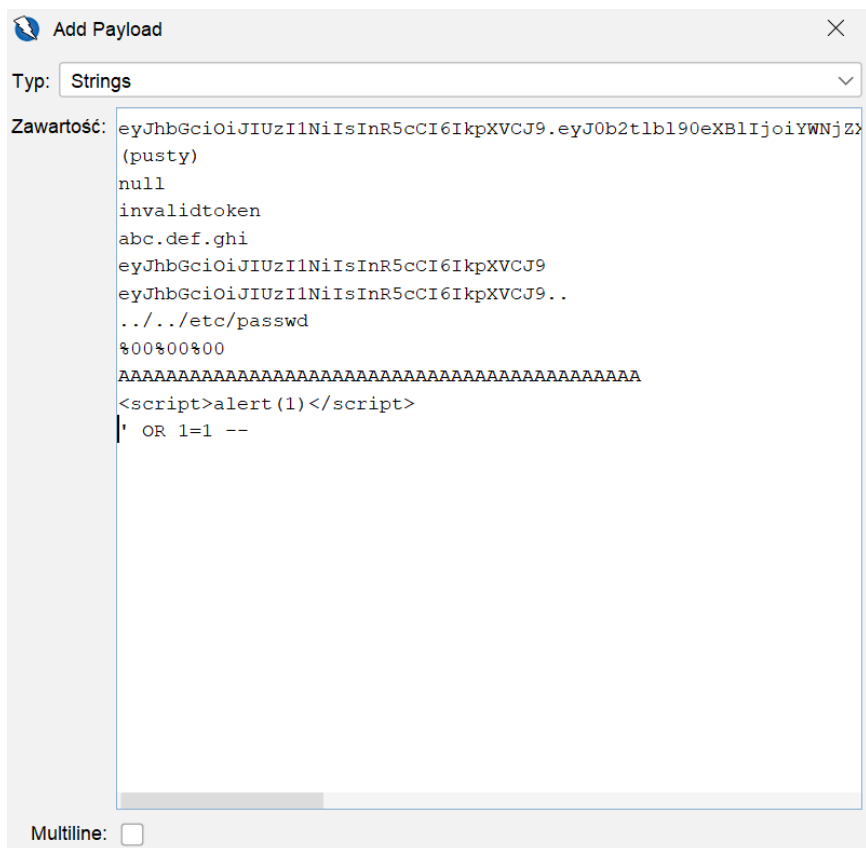
<https://iot-ai-backend.onrender.com/auth/core/users/>

Test polegał na uzyskaniu dostępu do zasobów, do których ma tylko dostęp zalogowany użytkownik z określoną rolą, poprzez podawanie fałszywych access tokenów. Test sprawdzał jak serwer reaguje na fałszywe access tokeny. Prawidłowo działający serwer powinien odrzucać złe tokeny. W przypadku podanego złego tokena, bądź tokena w którym zostaje zmieniona tylko jedna litera, powinien pojawiać się wynik- 401 lub 403 lub 200. W tym wypadku kod 200 oznacza błąd. Lista payloadów, które były podane jest opisana niżej.

Kod do każdego z payloadu wygląda następująco:

```
GET https://iot-ai-backend.onrender.com/auth/core/users/ HTTP/1.1
host: iot-ai-backend.onrender.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0
  Safari/537.36
content-length: 228
Authorization: Bearer <fałszywe payloady>
```

w polu <fałszywe payloady> zostają wstrzyknięte następujące payloady dla aktualnego testu:



Rysunek 68. Widok kolejnych serii payloadów

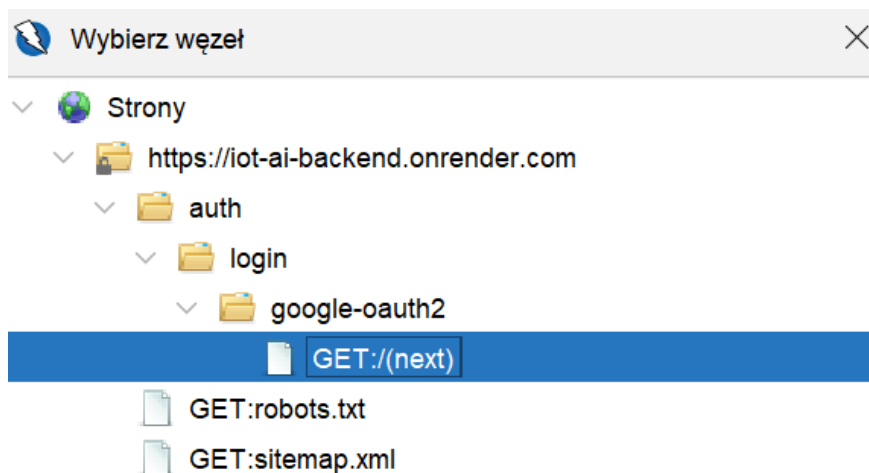
Powyższe okno zawiera [m.in](#) następujące payloady:

- ../../etc/passwd – testy na path traversal
- \x00\x00\x00 – testowanie obsługi znaków null
- AAAAAAAAAA... – fuzzing z dużą ilością danych
- <script>alert(1)</script> – XSS
- ' OR 1=1 -- – SQL Injection

wyniki testu:

Task ID ^	Typ wiadomości	Code	Powód	RTT	Size Resp. Header	Size Resp. Body	Najwyższy Alert	Stan	Payloads
6 Fuzzed		401 Unauthorized	259 ms	547 bajtów	172 bajtów				../etc/passwd
7 Fuzzed		401 Unauthorized	259 ms	547 bajtów	172 bajtów				%00%00%00
8 Fuzzed		401 Unauthorized	259 ms	547 bajtów	172 bajtów				AAAAAAAAAAAA...
9 Fuzzed		401 Unauthorized	277 ms	547 bajtów	172 bajtów				<script>alert(1)</script>
10 Fuzzed		401 Unauthorized	256 ms	547 bajtów	172 bajtów				' OR 1=1 --

Rysunek 69. Widok wyników testów



Rysunek 70. Widok wybierania węzła do badań w oprogramowaniu ZAP

ID	Źródło	Req. Timestamp	Metody	URL	Code	Powód	RTT	Size Resp. Body	Najwyższy Alert	Note	Tagi
1	Proxy	1.05.2025, 05:51:36	GET	https://iot-ai-backend.onrender.com/auth/login...	200	OK	311 ms	1 077 885 bajtów	Średni		Form, Password, Hid...

Rysunek 71. Widok wyników badań o największym priorytecie zagrożenia

7.2.4. Wnioski z testowania serwera backednowego oraz jego endpointów

Testom poddano: <https://iot-ai-backend.onrender.com> oraz jego endpointy. Wyniki testów zawarte są na zrzutach (rysunki 49 do 70), dla poszczególnych endpointów zawarte są odpowiedzi na wysłane payloady.

Tabela z wynikami testów składa się z kolumn: Tak ID, Typ wiadomości, Powód, RTT czyli czas odpowiedzi serwera oraz rozmiar odpowiedzi podany jako Size Resp.Header /Body i informacja o najwyższym alercie, stanie i aktualnie wykorzystywanym payloadzie.

Wyniki testów są jednoznacznie, gdyż wszystkie payloady uzyskały status 401 Unauthorized, oprócz payload z prawidłowym tokenem dostępu. To pokazuje, że pozostałe odpowiedzi serwera są definiowane jako odrzucone z powodu braku uprawnień, payload z właściwym tokenem przeszedł dalej, ale w ostateczności nie spowodowało to w żadnym wypadku dalszego nieautoryzowanego przyznania dostępu do aplikacji. Dodatkowo oprogramowanie ZAP nie sklasyfikowało w żadnej badanej próbie krytycznego błędu lub alarmu.

7.2.5. Badanie frontendu aplikacji pod względem bezpieczeństwa

Oprogramowanie ZAP jest przeznaczony do znajdowania potencjalnych podatności na obszarze backendowym związanych np. z JWT, CSRF. Natomiast w związku z tym, że frontend aplikacji komunikuje się z serwerem backendowym – może nadal istnieć ryzyko ataków takich jak: CSP, XSS, a oprogramowanie ZAP również takie zagrożenia wykrywa. Badanie frontendu aplikacji pod względem bezpieczeństwa zrealizowano z wykorzystaniem OWASP ZAP v2.16.1.

Badania frontendu: `fastinatechnology.streamli.app`:

Badania aplikacji od strony frontendu zostały przeprowadzone za pomocą OWASP ZAP v2.16.1 i zawierały kilka serii testów. Testy przeprowadzone ujawniły w raporcie następujące potencjalne zagrożenia, na które to system mógłby być podatny, oraz przyczyny z jakich te podatności wynikają. W raporcie podano łącznie 8 wykrytych alertów, podzielonych według następujących kategorii zagrożeń: wysokie, średnie, niskie i informacyjne. Do zagrożeń sklasyfikowanych jako średnie należały:

- Content Security Policy Header Not Set - wynika z tego że aplikacja nie ma ustawionego nagłówka CSP, brak CSP może wiązać się z większym ryzykiem na ataki XSS;
- Missing Anti-clickjacking Header - testy wykryły brak nagłówka X-Frame-Options, czyli takiego, który chroni przed atakiem Clickjacking polegającym na możliwym osadzeniu strony w innym serwisie.

Natomiast do zagrożeń o niskim ryzyku należy:

- Server Leaks Version Information via "Server" Header - ten komunikat wygenerowany z raportu oznacza, że serwer zdradza informację o wersji aplikacji.

Do kryteriów informacyjnych (niekoniecznie zagrożeń) należą:

- Modern Web Application- informacja, że aplikacja została sklasyfikowana w tym raporcie jak nowoczesna;
- Session Management Response Identified - wykryty został mechanizm, który zarządza sesją, co jest potwierdzeniem działania logowania do aplikacji.
- Re-examine Cache - informacja oznaczająca o ponownym przeanalizowanie ustawień nagłówków HTTP Cache-Control

7.2.6. Wnioski ze skanu front-endowej części aplikacji

Powyższe alerty są wynikami skanu bezpieczeństwa front-endowej części aplikacji. Ogólnie wyniki nie są poważnymi zagrożeniami i można je naprawić dodając nagłówki w aplikacji tj. nagłówek CSP oraz nagłówek anti-clickjacking oraz blokadę w wysyłaniu informacji o wersji serwera. Natomiast aplikacja zaprezentowana w pracy jest to prototypowa aplikacja testowa i jest hostowana na streamlite. Hosting aplikacji na streamlite Cloud ogranicza dodawanie bezpośrednio własnych nagłówków CSP i X-Frame Options, można natomiast to zrobić na innym serwerze np. Render lub VPS lub użyć CloudFare.

7.2.7. Testy bezpiecznego połączenia backendu Rendera

Dane potrzebne do wykonania testu to IP backendu hostowanego na renderze oraz adres hosta przedstawiono na rysunku 72.

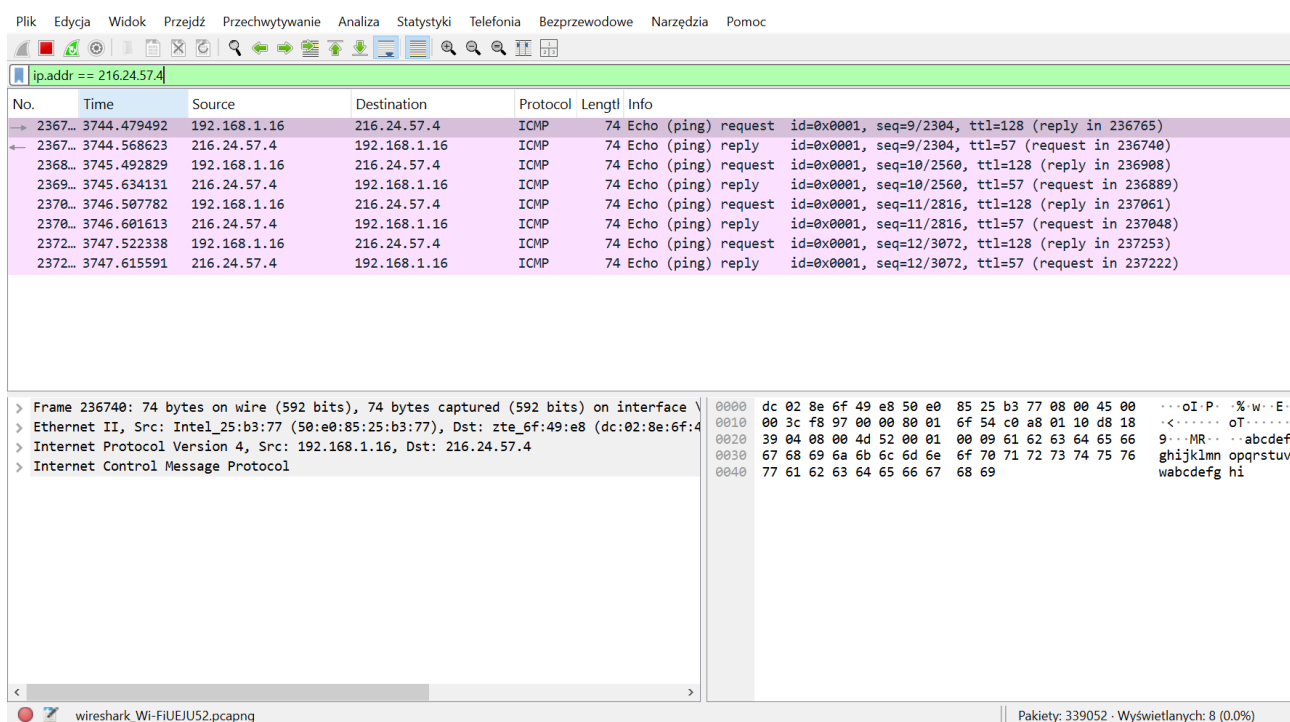
```
C:\Windows\system32>ping iot-ai-backend.onrender.com

Pinging gcp-us-west1-1.origin.onrender.com.cdn.cloudflare.net [216.24.57.4] with 32 bytes of data:
Reply from 216.24.57.4: bytes=32 time=89ms TTL=57
Reply from 216.24.57.4: bytes=32 time=141ms TTL=57
Reply from 216.24.57.4: bytes=32 time=93ms TTL=57
```

Rysunek 72. Wynik testu połączenia z backendem Rendera przy użyciu polecenia ping do hosta iot-ai-backend.onrender.com

Wnioski:

- Backend iot-ai-backend.onrender.com jest osiągalny z testowanej lokalizacji.
- Tłumaczenie nazwy hosta na adres IP działa poprawnie.
- Czas odpowiedzi jest stabilny, co świadczy o prawidłowym działaniu sieci.
- Brak utraconych pakietów wskazuje na stabilne połączenie TCP/IP.



Rysunek 73. widok z panelu Wireshark

Na rys.73 widnieją tylko podstawowe komunikację ICMP i jest tak dopóki nie zostanie złamany SSL lub testy nie będą wykonywane lokalnie, zrzut ekranu na rys.73 jest dowodem na to, że transmisja w aplikacji jest chroniona.

7.3 Ocena wyników badań i propozycje rozwiązań

Analiza z rys.73 z poprzedniego rozdziału przeprowadzona została narzędziem Wireshark. Celem sprawdzenia dostępności serwera oraz jego zabezpieczeń DNS i ICMP (z ang.Internet Control Message Protocol). ICMP służy głównie do informowania w błędach transmisji oraz wykrywaniu ewentualnych przeciążeń sieciowych lub do testowania połączeń między hostami. Szczegółowe wnioski dla tego testu zostały przedstawione w tabeli 8 poniżej:

Tabela 9. Wyniki badań z oprogramowania Wireshark

Badany aspekt	Wniosek
ICMP	opis
Dostępność odpowiedzi z serwera <code>iot-ai-backend.onrender.com</code>	po wykonaniu komendy <code>ping</code> serwer odpowiada ICMP Echo Reply, oznacza to, że <ul style="list-style-type: none"> – serwer jest aktywny – pakiety docierają do celu i wracają – nie ma pełnej blokady (np. firewalla dla ICMP).
DNS	DNS “przetransformuje” nazwę serwera na adres IP, w wypadku tego testu adres IP zostaje zwrócony co oznacza, że istnieje on w sieci i działa w pełni poprawnie
RTT (Echo Request)	czas odpowiedzi serwera wynosi 89–141 ms), z czego wynika że serwer nie ma niestabilnego połączenia lub wolnego połączenia i co za tym idzie nie ma też utraty pakietów związanym z tym procesem
Wireshark	Informacje na temat prawidłowości komunikacji sieciowej jak ICMP, Echo Request oraz Echo Reply oraz zgodności identyfikatorów i numerów sekwencyjnych wykazują poprawne zależności oraz nie ma żadnych innych nieprawidłowości w sekcjach omawianych powyżej

8. Dyskusja uzyskanych wyników

Głównym założeniem aplikacji była możliwość predykcji na podstawie stale nauczanego modelu AI, w oparciu o nowe dane gromadzone w czasie rzeczywistym. W wyniku przeprowadzonych badań został wybrany najlepszy algorytm do trenowania modelu tj. Gradient Boosting Regressor spośród pozostałych badanych algorytmów, które były przedstawione w głównym rozdziale badawczym pracy i obejmowały porównanie algorytmów pod względem czasu trenowania, błędu oraz trafności predykcji. Porównując wartości odnotowywane w danych chwilowych z zapisanymi wartościami predykcyjnymi, należy stwierdzić, że proces przewidywania krótkofalowego działa w sposób prawidłowy, a system jest chroniony dzięki nieustannemu działającemu modelowi predykcji oraz doradztwu wyuczonych agentów. Celem najbezpieczniejszego i najbardziej efektywnego korzystania z magazynu energii – zaimplementowano w aplikacji sterowanie niezależnie (przy pomocy agentów AI) oraz zależnie od użytkownika (w przypadku awarii lub innego rodzaju potrzeby ustawienia magazynu w stanie charge , discharge).

8.1. Ocena wdrożonych modeli i agentów

Zastosowanie agentów AI do podejmowania decyzji w systemach magazynów energii zostało podjęte w implementacji aplikacji jak i również szczegółowo opisane w pracy. Umiejętne wytrenowanie agentów ma szereg zalet i potencjalnych obszarów rozwojowych do zastosowania, nie tylko w zarządzaniu małymi magazynami energii ale również do zarządzania systemami, w których człowiek jest narażony na popełnianie wielokrotnych błędów, powodowanych zmęczeniem, natłokiem innych obowiązków , lub nierzadko brakiem kompetencji wynikającego z krótkiego stażu, czy zbyt skomplikowanymi procesami uruchomieniowymi kontrolowanego systemu. Ponadto modele AI są dostępne przez cały czas, nie biorą urlopu, nie „idą na L4”, tylko systematycznie, co zadany interwał czasowy nieustannie mogą kontrolować podległy im system. Obecnie dzięki zastosowaniu modeli LLM AI informuje również w sposób zrozumiały dla użytkownika, poprzez zwyczajne ludzkie objaśnienia podejmowanych decyzji.

Modele LLM w przeprowadzonych badaniach podejmowania decyzji wyróżniały się [m.in.](#):

- dużą elastycznością językową i odpowiedni sposób radzeniem sobie z niepełnymi danymi wejściowymi, czyli umiejętnym rozpoznawaniu kontekstu głównego z danych;
- elastyczna adaptacja procesu decyzyjnego - modele takie mogą być dostrojone fine-tuningiem lub one-shot tuningiem lub wykorzystywać wyniki z predykcji klasycznych modeli opartych na algorytmach z biblioteki scikit-learn. Dostosowanie promptów tych modeli można dowolnie zmieniać co w przypadku aplikacji stworzonej w celu zarządzania magazynem energii jest bardzo dobrym rozwiązaniem i pozwala na jej późniejsze rozwijanie w dziedzinie nowych agentów AI.

8.2. Wpływ zakłóceń jakości predykcji i sterowanie systemem

Anomalie oraz wszelkie wyżej przedstawione zagrożenia przejęcia kontroli nad systemem są nieodłącznym źródłem zagrożeń na jakie narażone jest każde urządzenie IoT w szczególności mające kilka połączeń z chmurą zewnętrzną. Detekcja anomalii i pokazywanie anomalii w aplikacji, stanowi już pewne zabezpieczenie oraz wzbudza świadomość użytkownika oraz zewnętrznych agentów LLM na sytuacje w aktualnie mierzonych wartościach. W modelowaniu oraz porównywaniu modeli szkolonych na danych zawierających anomalie jak i tych, które tych anomalii nie miały pokazały różnice w krzywej predykcyjnej. W rozdziale 6 wykazano, jak nawet niewielka liczba anomalii może skutkować różnicami w uzyskiwanych wynikach.

Podatność systemu na zagrożenia od strony cybernetycznej ma wpływ na bezpieczeństwo przechowywanych i gromadzonych danych. Natomiast w związku z wykonanymi badaniami opierającymi się na serii testów wykazano, że aplikacja w sposób poprawny radzi sobie ze znaczną większością zagrożeń. Nie zidentyfikowano żadnych zagrożeń o wysokim stopniu, na które trzeba byłoby reagować natychmiast, a rozwiązania dla pozostałych zagrożeń zostały szczegółowo przedstawione i ocenione w rozdziale 7.

8.3 Dalszy rozwój badań

Inteligentne sterowanie poprzez wykorzystanie do wspomagania decyzji przez agentów AI czy całkowicie i niezależne sterowanie przez nie wybranym systemem jest obecnie bardzo wrażliwym zagadnieniem. Temat ten jest obecnie często podejmowany w mediach branżowych i mierzy się z nim praktycznie każdy kto korzysta zawodowo z technik ICT. W obiegu nieformalnym często pojawia się bardzo popularny slang pt. „Czy sztuczna inteligencja zastąpi człowieka?”. Okazuje się, że procesy sterowania sprzętem fizycznym przez sztuczną inteligencję są bardzo możliwe, w niniejszej pracy zaprezentowano możliwość wyzerowywania przełączaniem magazynu energii jako elementu infrastruktury krytycznej. Wniosek jest taki, że chociaż nadal to człowiek podejmuje decyzje o tym czy pozwolić AI na podjęcie decyzji, to jak wykazano, nic nie stoi na przeszkodzie aby tak zmodyfikować implementację aplikacji, aby agent AI nie musiał czekać na pozwolenie do podjęcia decyzji wydane przez człowieka. W tym celu wystarczy jedynie zmienić implementację w następujący sposób: aby funkcją przesyłającą prompt do agenta AI wykonywała się co zadany czas np. co 20 sekund lub 5 minut pytała AI o decyzje w ten sposób agent AI samodzielnie działa w tle co chwila aktualizując stan mikrokontrolera w zależności od jego personalnego wyboru opartego głównie na danych statystycznych zgromadzonych z sensorów. Takie podejście będzie przyszłością w nowoczesnych systemach IoT, które – w co nie należy wątpić – będą opierały się na rozwiązaniach jakie dostarcza sztuczna inteligencja wraz z gotowymi modelami językowymi LLM

9. Podsumowanie

Przedstawiona praca skupiała się na opracowaniu systemu opierającego na wykorzystaniu sztucznej inteligencji poprzez zaimplementowanie w tym celu własnych agentów AI. W pracy zaimplementowano modele predykcji mierzonych wartości z czujników niezbędnych do sterowania prototypem magazynu energii. Stanowisko badawcze zostało również przez Autorkę zaprojektowane i wykonane samodzielnie, na potrzeby realizacji zaplanowanych scenariuszy badań. Problemy badawcze, jakie zostały przedstawione w pracy to:

- Czy można zautomatyzować reakcję systemu bez potrzeby interwencji człowieka?
- Jakie ryzyko niosą błędne predykcje w kontekście systemów sterowania oraz jak oczyszczanie danych z anomalii i wartości zerowych wpływa na jakość predykcji?
- Które algorytmy w przewidywaniu danych sensorycznych dla mierzonych wielkości fizycznych były najskuteczniejsze?
- Czy aplikacja opracowana na potrzeby testów spełnia wymagania bezpieczeństwa, niezbędne w zastosowaniach infrastruktury krytycznej?

Przeprowadzone badania pozwoliły na potwierdzenie wstępnych hipotez założonych w pracy ujętych jako założenia systemu przy implementacji aplikacji do magazynu energii. Odpowiedź na główne pytanie odnośnie automatyzacji systemu została zarówno zaimplementowana w aplikacji, jak i przebadana w wymiarze skuteczności podejmowania decyzji, celem pokazania, że sterowanie urządzeniami fizycznymi przez AI jest jak najbardziej realne i może być niezależne od człowieka.

W aspekcie podejmowanego w pracy problemu zostało pokazane od czego zależy jakość podjętej decyzji przez AI i jak się ma tak uzyskany wynik do decyzji wyznaczonej poprzez klasyczne modele regresji, które też w celu wykonania badań zostały zaimplementowane.

W odpowiedzi na pytanie badawcze odnośnie ryzyka błędnej predykcji wykazano na przykładzie porównania zbudowanych modeli na danych z anomaliami i oczyszczonych, dla których opracowano dwa rodzaje modeli, dla modelowania predykcji dwóch badanych wielkości tego samego typu zarówno dla modelu uczonego na danych z anomaliami jak i po oczyszczeniu. Ponadto wyszczególniono czynniki, które powodują wystąpienia anomalii.

W ramach pracy badawczej dokonano również zestawienia algorytmów do predykcji, porównano je i wybrano jeden najskuteczniejszy - Gradient Boosting Regressor, sprawdzony na zbieranych danych, fizycznie gromadzonych ze zbudowanego w tym celu prototypowego stanowiska laboratoryjnego. Dane zbierano w czasie rzeczywistym i uzyskiwano od razu efekt sterowania przez AI. Uzyskane z porównania wybranych modelu zestawienie było odpowiedzią na pytanie badawcze dotyczące najskuteczniejszych algorytmów z dziedziny regresji czyli uczenia maszynowego, a dokładnie regresji nadzorowanej do predykcji.

W wymiarze cyberbezpieczeństwa opracowanego rozwiązania, wykonano testy odporności na wszystkie zidentyfikowane w State of Arts formy ataku, zarówno na aplikację jak i na urządzenia IoT – wykorzystano w tym celu oprogramowanie ZAP oraz Wireshark z Insomnia. Wyniki testów wykazały zachowanie wymaganego poziomu bezpieczeństwa.

Bibliografia

- [1]. [What Is Teachable Machine? How to Use It to Teach AI | Tech & Learning \[data dostępu 28.05.2025\]](#)
- [2]. Warden, P., & Situnayake, D. (2020). *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*. O'Reilly Media
- [3]. Alblehai, F. (2025). Artificial intelligence-driven cybersecurity system for internet of things using self-attention deep learning and metaheuristic algorithms. *Scientific Reports*, 15, Article 13215. <https://doi.org/10.1038/s41598-025-98056-2>
- [4]. Sarker, I. H. (2021). *CyberLearning: Effectiveness Analysis of Machine Learning Security Modeling to Detect Cyber-Anomalies and Multi-Attacks*. *Internet of Things*, 14, 100393
- [5]. Lipka, J. B., & Hans, C. A. (2024). *Data-driven model predictive control of battery storage units*. arXiv preprint arXiv:2407.05157. <https://arxiv.org/abs/2407.05157>
- [6]. Aung, Y. L., Christian, I., Dong, Y., Ye, X., Chattopadhyay, S., & Zhou, J. (2025). Generative AI for Internet of Things Security: Challenges and Opportunities. arXiv preprint arXiv:2502.08886. <https://arxiv.org/abs/2502.08886>
- [7]. Aghajani, G. R., & Kalantar, M. "Optimal energy management of a smart microgrid with hybrid renewable energy systems using new fuzzy adaptive modified particle swarm optimization."(.2016).
- [8]. Matthew, B., Utman, I., Gordan, M., & Ejaz, U. (2025). *Optimizing Microgrid Energy Management with Hybrid Machine Learning Models*. ResearchGate. <https://www.researchgate.net/publication/391482950>
- [9]. Munoz, O., Ruelas, A., Rosales, P., Acuña, A., Suastegui, A., & Lara, F. (2022). *Design and Development of an IoT Smart Meter with Load Control for Home Energy Management Systems*. *Sensors*, 22(19), 7536. <https://doi.org/10.3390/s22197536>
- [10] Skibko, Z., & Pisarek, Ł. (2023). *Nowoczesne metody magazynowania energii*. Politechnika Białostocka.
- [11] Pizzolli, D., Cossu, G., Santoro, D., Capra, L., Dupont, C., Charalampous, D., De Pellegrini, F., Antonelli, F., & Cretti, S. (2018). *Cloud4IoT: A heterogeneous, distributed and*

autonomic cloud platform for the IoT. *arXiv preprint arXiv:1810.01839*.
<https://arxiv.org/abs/1810.01839>

[12]. Pan, Q., & Wu, J. (2024). AI Technology for Cybersecurity and IoT Applications. *Sensors*, Special Issue. https://www.mdpi.com/journal/sensors/special_issues/1H1N3QZMJ2

[13]. Plett, G. L. (2004). Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 3. State and parameter estimation. *Journal of Power Sources*, 134(2), 277–292

[14]. Roman, R., Najera, P., & Lopez, J. (2011). Securing the internet of things. *Computer*, 44(9), 51–58. <https://doi.org/10.1109/MC.2011.291>

[15]. Stojmenovic, I., & Wen, S. (2014). The fog computing paradigm: Scenarios and security issues. 2014 Federated Conference on Computer Science and Information Systems, 1–8. <https://doi.org/10.15439/2014F083>

[16]. Mosenia, A., & Jha, N. K. (2017). A comprehensive study of security of Internet-of-Things. *IEEE Transactions on Emerging Topics in Computing*, 5(4), 586–602. <https://doi.org/10.1109/TETC.2016.2606384>

[17]. Yang, Y., Wu, L., Yin, G., Li, L., & Zhao, H. (2017). A survey on security and privacy issues in Internet-of-Things. *IEEE Internet of Things Journal*, 4(5), 1250–1258. <https://doi.org/10.1109/JIOT.2017.2694844>

[18]. Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2), 1153–1176. <https://doi.org/10.1109/COMST.2015.2494502>

[19]. Tahsien, S. M., Karimipour, H., & Spachos, P. (2020). Machine Learning Based Solutions for Security of Internet of Things (IoT): A Survey. *Journal of Network and Computer Applications*, 161, 102630

[20]. Rot, A., & Blaić, B. (2016). Zagrożenia wynikające z implementacji koncepcji Internetu rzeczy. Rekomendacje dla organizacji i dostawców rozwiązań. *Informatyka Ekonomiczna = Business Informatics*, (3), 76–91

[21]. [Wireless SoCs, Software, Cloud and AIoT Solutions | Espressif Systems](#) [data dostępu 27.05.2025]

[22]. [Influx Query Language \(InfluxQL\) | InfluxDB OSS v1 Documentation](#) [data dostępu 27.05.2025]

[23] Hernández, M., Mayoral, L. A., & Alonso, G. (2023). Real-time data acquisition with ESP32 for IoT applications using open-source MQTT brokers. *Proceedings of the 2023 International Conference on IoT Systems*, 1–7. <https://www.researchgate.net/publication/388464048>

[24] Sabo, A., Suleiman, H. O., Dahiru, Y., Jatau, N. D., Yusuf, A., & Chikodi, A. T. (2024). *Development and Implementation of an ESP32 IoT-Based Smart Grid for Enhanced Energy Efficiency and Management*.

[25] Narendran, S., Pradeep, P., & Ramesh, M. V. (2017). An Internet of Things (IoT) based Sustainable Water Management. *International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 1–6. <https://doi.org/10.1109/WiSPNET.2017.8299821>

[26] Simon, T. (2017). Critical Infrastructure and the Internet of Things. *Global Commission on Internet Governance Paper Series*, 46. <https://www.cigionline.org/publications/critical-infrastructure-and-internet-things-0/>

[27] Silva, N. S. e., Castro, R., & Ferrão, P. (2025). Smart Grids in the Context of Smart Cities: A Literature Review and Gap Analysis. *Energies*, 18(5), 1186. <https://doi.org/10.3390/en18051186>

[28] Delplace, A., Hermoso, S., & Anandita, K. (2020). Cyber Attack Detection thanks to Machine Learning Algorithms

[29] Apruzzese, G., Laskov, P., Montes de Oca, E., Mallouli, W., Rapa, L. B., Grammatopoulos, A. V., & Di Franco, F. (2022). The Role of Machine Learning in Cybersecurity.

[30] Alhalabi, W., Manoharan, H., & Alhalabi, A. (2023). Efficient data transmission on wireless communication through a hybrid microwave transmission method. *Sensors*, 23(1), 1234. <https://doi.org/10.3390/s230101234>

[31] David, R., Duke, J., Jain, A., Reddi, V. J., Jeffries, N., Li, J., Kreeger, N., Nappier, I., Natraj, M., Regev, S., Rhodes, R., Wang, T., & Warden, P. (2020). TensorFlow Lite Micro: Embedded Machine Learning on TinyML Systems

[32] Wang, J., Antwi-Afari, M. F., Tezel, A., & Kasim, T. (2024). Artificial Intelligence in Cloud Computing Technology in the Construction Industry: A Bibliometric and Systematic Review. *Journal of Information Technology in Construction*, 29, 480–502.

[33] Tyagi, A. (2024). Edge AI and TinyML: Real-Time Intelligence at the Edge.

[34] Ruohonen, J. (2024). A Systematic Literature Review on the NIS2 Directive.[2412.08084] A Systematic Literature Review on the NIS2 Directive

[35] Such-Pyrgiel, M., & Rosińska-Wielec, E. (2024). Internet of Things platform as support for the digitization of enterprises on the example of the Lingaro IoT Cloud Platform project.(<https://www.researchgate.net/publication/379386517>)

[36] Orange Pi 2G-IoT - Orangepi [data dostępu 1.06.2025]

[37] Barghi, A., Kosari, A., Shokri, M., & Sheikhaei, S. (2015). Intelligent lighting control with LEDs for smart home. *International Conference on Smart Grid and Clean Energy Technologies (ICSGCE)*.

[38] Lu, J., Sookoor, T., Srinivasan, V., Gao, G., Holben, B., Stankovic, J., Field, E., & Whitehouse, K. (2010). The smart thermostat: Using occupancy sensors to save energy in homes. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems* (pp. 211–224). ACM. <https://doi.org/10.1145/1869983.1870005>

[39] Sharma, A., & Goen, A. (2018). Smart Home Security System. ResearchGate. https://www.researchgate.net/publication/338071682_Smart_Home_Security_System

[40] Smart Metering. Jak to działa? - e-magazyny.pl [data dostępu 2.06.2025]

[41] Jawurek, M., Johns, M., & Kerschbaum, F. (2011). Plug-in privacy for smart metering billing. In G. Danezis (Ed.), *Privacy Enhancing Technologies* (pp. 192–210). Springer. https://doi.org/10.1007/978-3-642-22263-4_11

[42] Rot, A., & Blaike, B. (2017). Bezpieczeństwo Internetu rzeczy. Wybrane zagrożenia i sposoby zabezpieczeń na przykładzie systemów produkcyjnych. *Zeszyty Naukowe Politechniki Częstochowskiej. Zarządzanie*, 26(1), 188–198

[43] Edge Impulse - The Leading Edge AI Platform [data dostępu 02.06.2025]

[44] Lin, J., Zhu, L., Chen, W.-M., Wang, W.-C., & Han, S. (2023). Tiny Machine Learning: Progress and Futures [Feature]. IEEE Circuits and Systems Magazine, 23(3), 8–34. <https://doi.org/10.1109/MCAS.2023.3302182>

[45] Xu, R., Nikouei, S. Y., Chen, Y., Blasch, E., & Aved, A. (2019). BlendMAS: A BLockchain-ENabled Decentralized Microservices Architecture for Smart Public Safety. arXiv preprint arXiv:1902.10567. <https://arxiv.org/abs/1902.10567>

Spis rysunków, spis tabel

Spis rysunków

Rysunek 1. Zestaw wspieranych mikrokontrolerów przez Edge Impulse	20
Rysunek 2. Wybór ilości przesłanych danych do agenta.	33
Rysunek 3. Komunikat systemu decyzyjnego	33
Rysunek 4. Prezentacja działania jednego z zapytanych o decyzję agentów	33
Rysunek 5. Podgląd danych w aplikacji na wykresach	34
Rysunek 6. Podgląd danych na wykresach cd	34
Rysunek 7. Zrzut wyników dla GradientBoostingRegressor	36
Rysunek 8. Wyznaczenie czasu trenowania dla Gradient Boosting Regression	36
Rysunek 9. Porównanie predykcji z wartościami mierzonymi dla Gradient Boosting Regression	37
Rysunek 10. Błąd predykcji dla Gradient Boosting Regression	38
Rysunek 11. Wyniki dla algorytmu LGBM Regression	38
Rysunek 12. Czas trenowania dla LGBM Regression	38
Rysunek 13. Wykres porównawczy predykcji z wynikami rzeczywistymi dla LGBM Regression	39
Rysunek 14. Wykres błędu predykcji dla LGBM Regression	39
Rysunek 15. Wyniki dla algorytmu RandomForest Regression	40
Rysunek 16. Wynik czasu trenowania dla RandomForest Regression	40
Rysunek 17. Wykres predykcji w porównaniu z mierzonymi wartościami rzeczywistymi dla RandomForest Regression,:	40

Rysunek 18. Wykres błędu predykcji dla algorytmu RandomForest Regression	41
Rysunek 19. Wyniki dla modelu utworzonego za pomocą Linear Regression	41
Rysunek 20. Predykcja na tle rzeczywistych wyników dla Linear Regression	42
Rysunek 21. Wykres błędu predykcji dla algorytmu Linear Regression	42
Rysunek 22. 22 Widok z poziomu aplikacji wykresu predykcji dla poziomu oświetlenia	43
Rysunek 23. Statystyki predykcji dla oświetlenia	43
Rysunek 24. Wykres predykcji z poziomu aplikacji dla predykcji temperatury	44
Rysunek 25. Statystyki predykcji	44
Rysunek 26. Predykcja wilgotności w czasie	44
Rysunek 27. Statystyki predykcji dla wilgotności	44
Rysunek 28. Predykcja prądu ładowania	45
Rysunek 29. Statystyki predykcji dla prądu ładowania	45
Rysunek 30. Zastosowane modele LLM do budowy agentów	46
Rysunek 31. Odpowiedź agenta LLM oraz podjęcie decyzji	46
Rysunek 32. Podgląd wyników dla bazy	47
Rysunek 33. Panel wyboru decyzji przez użytkownika	47
Rysunek 34. Decyzje wysyłane przez agenta AI do bazy	51
Rysunek 35. Szczegółowe parametry bazy danych	51
Rysunek 36. Widok zapytania do bazy i wyników	52
Rysunek 37. Schemat części pomiarowej i sterującej do prototypu magazynu energii	52
Rysunek 38. Spis komponentów wygenerowany z oprogramowania Fusion module electronics	360 53
Rysunek 39. Anomalie dla prądu wyjściowego	54
Rysunek 40. Anomalie dla mierzonego poziomu oświetlenia	54
Rysunek 41. Anomalie w temperaturze	55
Rysunek 42. Anomalie w wilgotności	55

Rysunek 43. Anomalie w prądzie ładowania	55
Rysunek 44. Predykcja prądu wyjściowego w czasie	56
Rysunek 45. Predykcja z anomaliami	56
Rysunek 46. Predykcja oświetlenia oczyszczona z anomalii	57
Rysunek 47. Widok powitalny aplikacji	59
Rysunek 48. Testowanie endpointu rejestracji	60
Rysunek 49. Zwracanie tokenu	60
Rysunek 50. Widok dostępu do chronionego tokenu	61
Rysunek 51. Przebieg testów autoryzacji	61
Rysunek 52. Dalsza część testów	62
Rysunek 53. Przebieg testów wstrzykiwania złośliwego kodu	62
Rysunek 54. Wynik testu	63
Rysunek 55. Przebieg testów wstrzykiwania złośliwych zapytań SQL	64
Rysunek 56. Przebieg testów	64
Rysunek 57. Sprawdzenie haseł w bazie	65
Rysunek 58. Sprawdzenie działania logowania przez Google	66
Rysunek 59. Widok panelu logowania przez Google	66
Rysunek 60. Widok po poprawnym zalogowaniu	67
Rysunek 61. Wyniki raportu (podane w procentach zagrożenia określonego typu)	68
Rysunek 62. Widok wyników badań	69
Rysunek 63. Widok testu na endpointzie powyżej	70
Rysunek 64. Widok z oprogramowania do przeprowadzania testów aplikacji.	70
Rysunek 65. Widok wprowadzonych fałszywych payloadów do testów	71
Rysunek 66. Wyniki badań fuzzowania	71
Rysunek 67. Wyniki badań dalsza część	72
Rysunek 68. Widok kolejnych serii payloadów	73

Rysunek 69. Widok wyników testów	73
Rysunek 70. Widok wybierania węzła do badań w oprogramowaniu ZAP	74
Rysunek 71. Widok wyników badań o największym priorytecie zagrożenia	74
Rysunek 72. Wynik testu połączenia z backendem Rendera przy użyciu polecenia ping do hosta <code>iot-ai-backend.onrender.com</code>	76
Rysunek 73. widok z panelu Wireshark	76
Spis tabel	
Tabela 1. Komponenty w IoT	10
Tabela 2. Komponenty w SmartHome	11
Tabela 3. Aktualne liczniki w przemysłowych wariantach dla gospodarstw domowych	13
Tabela 4. Algorytmy dla urządzeń o ograniczonym zasobie	18
Tabela 5. Porównanie TinyML oraz CloudAI	25
Tabela 6. Główne założenia dyrektywy NIS2	30
Tabela 7. Podsumowanie wyników badań wyboru algorytmu	48
Tabela 8. Komponenty użyte w projekcie	59
Tabela 9. Wyniki badań z oprogramowania Wireshark	77

Załączniki do pracy:

[1] [faustyna77/iot_AI_backend](#)

[2] [faustyna77/iot_AI_at_newer](#)