**WOJSKOWA AKADEMIA TECHNICZNA**

**im. Jarosława Dąbrowskiego**

**WYDZIAŁ CYBERNETYKI**

A logo of a bird

Description automatically generated

**PRACA DYPLOMOWA**

STACJONARNE STUDIA II°

Temat:

**Implementacja pulpitu do monitorowania adekwatności modelu symulacyjnego**

Autor:

**Inż. Rafał Łazicki**

Promotor pracy:

**Lorem ipsum**

Wstęp

- Cel pracy

- Zakres pracy

- Struktura pracy

1. Podstawy teoretyczne

- 1.1 Technologia Digital Twin

- 1.1.1 Definicja i zastosowania

- 1.1.2 Istniejące rozwiązania

- 1.1.3 Monitorowanie modeli

- 1.2 Metody analizy i walidacji modeli symulacyjnych

- 1.2.1 Metody statystyczne

- 1.2.2 Metody maszynowego uczenia

- 1.3 Metody granulacji danych

- 1.4 Architektura systemów wielokomponentowych

2. Architektura

- 2.1 Diagram komponentów

- 2.1 Warstwa serwerowa

- 2.1.1 REST API

- 2.1.2 Integracja modeli symulacyjnych

- 2.1.3 Zastosowanie Celery w zarządzaniu zadaniami

- 2.1.4 Wykorzystanie Django, djangorestframework

- 2.1.5 Poetry jako zarządcze narzędzie wirtualizacji, obsługi zależności oraz budowy paczki

- 2.2 Warstwa interfejsu użytkownika

- 2.2.1 Wykorzystanie ReactJS

- wykorzystanie vite

- 2.2.2 Wykorzystanie Recharts i Material-UI

- 2.3 Konteneryzacja i orkiestracja

- 2.3.1 Zastosowanie Dockera

- uprodukcyjnianie

- skalowalność

- deployment

- 2.3.2 Komunikacja między komponentami

- sieć wewnętrzna dockera

- http, websocket, redis

- 2.4 Wybrane aspekty procesu CI/CD

- 2.4.1 Github pipelines

- 2.4.2 Github workflow

- 2.4.3 Budowa oraz publikacja paczki

3. Implementacja

- 3.1 Moduł symulacyjny (Mock)

- 3.1.1 Generowanie losowych danych

- 3.1.2 Imitacja modeli symulacyjnych

- 3.1.3 Komunikacja z backendem przez Redis

- 3.2 Moduł analizy danych historycznych

- 3.2.1 Przechowywanie i odczyt danych

- 3.2.2 Granulacja danych

- 3.2.3 Prezentacja historycznych predykcji

- 3.3 Moduł prognozowania w czasie rzeczywistym

- 3.3.1 Odbiór danych

- 3.3.2 Generowanie predykcji

- 3.4 Moduł porównawczy metryk modeli

- 3.4.1 Prezentacja metryk

- 3.4.2 Wykrywanie i prezentacja błędów

- 3.5 Rejestracja modeli oraz sensorów

- 3.5.1 Wykorzystanie Django Admin Panel

- 3.5.2 Wykorzystanie REST API

- 3.6 Moduł monitorowania w czasie rzeczywistym

- 3.6.1 Odbiór i przetwarzanie danych

- 3.6.2 Komunikacja WebSocket

4. Testy

- 4.1 Środowisko testowe

- 4.1.1 Selekcja i charakteryzacja narzędzi testujących

- 4.1.2 Konfiguracja i parametryzacja środowiska testowego

- 4.1.3 Procedury inicjalizacyjne i przygotowawcze

- 4.1.4 Metodologie testowania

- 4.2 Ewaluacja funkcjonalności i zgodności

- 4.4 Analiza i optymalizacja zużycia zasobów

- 4.5 Identyfikacja i rekomendacje wobec wykrytych nieprawidłowości

- 4.6 Analiza kodu przy wykorzystaniu testów statycznych

5. Dyskusja

- 4.4 Analiza wyników

- 4.5 Ograniczenia i Wyzwania

- 4.6 Możliwości dalszego rozwoju

- caching z wykorzystaniem redis

- wykorzystanie analitycznej bazy danych (cassandra/clickhouse)

- zbudowanie dedykowanego narzędzia wizualizacji danych

6. Podsumowanie

7. Bibliografia