

Badania podstawowe – TLR poziom I

Predykcja zasięgu EV z korekcją stylu jazdy

Zespół: Piotr Zienowicz, Konrad Zieliński

Kierownik projektu: Konrad Zieliński

5 listopada 2025

Spis treści

1	Lab 2 — TLR poziom I	2
2	Lab 3 — TLR poziom II	3
3	Lab 4 — TLR poziom III	5
4	Lab 5 — TLR poziom IV	6
5	Lab 6 — Punkt Kontrolny	9
6	Lab 7 — TLR poziom V	11

1 Lab 2 — TLR poziom I

Opis zadania (TLR-1)

Tutaj rodzi się **koncepcja**. Celem TLR-1 jest sformułowanie *głównej idei*, która ma potencjał, lecz nie ma jeszcze jasno określonego ostatecznego zastosowania czy produktu. Dopuszczalne jest wskazanie wstępnych, **potencjalnych zastosowań**, które mogą zostać rozwinięte na kolejnych poziomach TLR.

Główna idea

Proponujemy **lekką metodę predykcji zasięgu pojazdów elektrycznych (EV)**, która łączy *sygnały stylu jazdy kierowcy* (np. zmienność prędkości, intensywność hamowań/przyspieszeń) z *warunkami zewnętrznymi* (pogoda, nachylenie trasy). **Hipoteza**: wzbogacenie informacji wejściowej o te czynniki pozwoli ograniczyć przeszacowania i niedoszacowania zasięgu względem prostych metod bazowych opartych jedynie o średnie zużycie energii.

Metryki (przykład: MAE)

Do oceny jakości wstępnych predykcji stosujemy m.in. **średni błąd bezwzględny MAE** (Mean Absolute Error):

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (1)$$

gdzie y_i to wartość rzeczywista (np. zasięg w km), a \hat{y}_i – wartość przewidywana. MAE wyrażony jest w tej samej jednostce co mierzona wielkość (np. km).

Przykład Rzeczywiste zasięgi: [200, 180, 220], przewidywane: [190, 175, 210] \Rightarrow błędy bezwzględne: [10, 5, 10], a zatem $\text{MAE} = 8,33$ km.

Narracja TLR-1 (styl opisu)

W wyniku **prac badawczych oraz analiz informatycznych**, zespół opracował **lekką metodę predykcji zasięgu EV**, łączącą sygnały stylu jazdy z warunkami zewnętrznymi. Zaobserwowano, że *dokładność prognoz (MAE) jest lepsza o ok. 12–18%* względem podejścia bazowego opartego wyłącznie o średnie zużycie energii z ostatnich kilometrów. Ponadto metoda wykazuje **adaptację do zmiennych warunków** (wiatr, temperatura) i różnic w topografii, co przekłada się na mniejsze przeszacowania i niedoszacowania. W kolejnych etapach (TLR-2) planowana jest *szersza walidacja* na odmiennych modelach EV i w różnych warunkach.

2 Lab 3 — TLR poziom II

Projekt: Predykcja zasięgu EV z korekcją stylu jazdy

Zadanie 1. Koncepcja technologii i jej przyszłe zastosowania

Narracja Przeprowadzono wstępną analizę możliwych zastosowań technologii predykcji zasięgu EV. Ustalono, że rozwiązanie powinno w **pierwszej kolejności** zostać wdrożone w **systemach pokładowych pojazdów (IVI/OS OEM)**, aby predykcja działała lokalnie i bez opóźnień. Zewnętrzne aplikacje mobilne (Android/iOS) oraz panel webowy dla flot mogą zostać przygotowane jako **kolejne etapy rozwoju**, po ugruntowaniu komponentu pokładowego. Optymalnym rozwiązaniem jest prosty, intuicyjny interfejs w pojeździe, minimalizujący liczbę kroków do uzyskania prognozy.

Zauważono, że komponent predykcyjny po odpowiedniej adaptacji może zostać rozszerzony o moduł *eco-suggestions* (mikrokorekty stylu jazdy) oraz moduł planowania ładowań (rekomendacja postojów i mocy ładowarek). Moduł predykcji zasięgu stanowi kluczowy komponent aplikacji produkcyjnej (najpierw pokładowej), natomiast pozostałe moduły mogą być włączane w kolejnych iteracjach lub w aplikacjach zewnętrznych.

Priorytety wdrożeniowe

1. Wdrożenie i optymalizacja komponentu predykcji w systemie pokładowym (IVI/OS OEM) – inference lokalnie, niski narzut obliczeniowy.
2. Integracja z czujnikami/telemetrią pojazdu (SoC, prędkość, ewentualnie OBD) oraz danymi o nachyleniu i pogodzie (cache/offline, gdy brak łączności).
3. Etap 2: lekkie aplikacje zewnętrzne (Android/iOS) do zbierania danych pilotażowych i wizualizacji; panel webowy dla flot.

Zakres technologiczny (kierunek rozwoju)

- Platformy docelowe: **IVI/OS OEM (priorytet)**, następnie Android/iOS oraz web (dashboard floty).
- Architektura: komponent pokładowy (zbieranie danych, inference, prezentacja alertów); usługa modelowania (API) do aktualizacji modelu i wymiany konfiguracji; panel webowy do metryk i historii (etap późniejszy).
- Integracje: API pogodowe (Open-Meteo, z cache); źródła wysokości (SRTM/OSM); opcjonalnie OBD (SoC/telemetria); mapy/trasowanie (OSM/komercyjne).
- UX: w pojeździe minimalna liczba kroków, przewidywalna latencja, tryb offline.

Przyszłe zastosowania

- Kierowca indywidualny: planowanie trasy i redukcja *range anxiety*.
- Floty/logistyka: planowanie kursów i ładowań, szacunek kosztów energii.
- Eco-driving: rekomendacje mikrokorekt stylu dla zmniejszenia zużycia.

Zadanie 2. Lista (podstawowa) rzeczy potrzebnych do dalszego rozwoju projektu

Infrastruktura i sprzęt

- Stabilny dostęp do Internetu (preferowane łącze światłowodowe).
- 2 komputery do programowania.
- 2 smartfony (Android/iOS) do testów i zbierania danych.
- 1–2 interfejsy OBD–II (dongle) – opcjonalnie, do pobierania SoC/telemetrii.
- Repozytorium/pamięć masowa na dane.

Oprogramowanie i usługi

- Środowisko: Python (pandas, scikit–learn, lightgbm/xgboost), Jupyter.
- System kontroli wersji: Git + repo.
- Issue tracker i komunikacja: Jira, Slack/Teams.
- Hosting demo (lub serwer wewnętrzny).
- Dostęp do API: pogoda (Open–Meteo), mapy/elewacja (OSM/SRTM), ewentualnie trasowanie.

Dane i testy

- Zbiór pilotażowy przejazdów (≥ 70 –100), różne profile tras i warunki pogodowe.
- Scenariusze testowe: miasto/trasa/mieszany; różne temperatury/wiatr; różne modele EV (jeśli możliwe).

3 Lab 4 — TLR poziom III

Projekt: Predykcja zasięgu EV z korekcją stylu jazdy

Zadanie 1. Zakres prac badawczych (moduły)

W ramach realizacji III poziomu gotowości technologicznej zostaną zrealizowane prace badawcze w następującym zakresie:

Moduł inference pokładowej predykcji zasięgu

- Cel: predykcja zasięgu EV lokalnie w systemie pokładowym (IVI/OS OEM) z niską latencją.
- Zakres: implementacja algorytmu predykcji wraz z podsystemem kontroli jakości.
- Nacisk: poprawność i stabilność wyników w różnych warunkach (pogoda, nachylenie), powtarzalność predykcji dla identycznych odcinków.

Moduł walidacji zgodności predykcji z danymi referencyjnymi

- Cel: weryfikacja jakości i kompletności predykcji względem danych rzeczywistych (przebyty dystans/zużycie).
- Zakres: automatyczne porównywanie prognoz zasięgu/zużycia na odcinkach do danych referencyjnych; raport zgodności (% odcinków w przedziale błędu $< 10\%$).
- Nacisk: spójny system walidacji (raporty, wykresy błędów, test na różnorodnych trasach), definicja końcowej miary zgodności w procentach.

Moduł przetwarzania strumieniowego (czas rzeczywisty)

- Cel: predykcja i aktualizacja zasięgu w czasie rzeczywistym dla długich tras, z jednoczesną prezentacją kierowcy.
- Zakres: ocena obciążenia CPU i pamięci RAM, tryb pracy offline (cache pogody/nachylenia), tolerancja danych brakujących/opóźnionych.
- Zasada: priorytet płynności (niskiej latencji) nad maksymalną kompletnością cech; badanie kompromisu jakości (accuracy) do opóźnienia (latency) i wyznaczenie punktu optymalnego.

4 Lab 5 — TLR poziom IV

Projekt: Predykcja zasięgu EV z korekcją stylu jazdy

Zadanie 1. Weryfikacja koncepcji w warunkach laboratoryjnych

W ramach realizacji IV poziomu gotowości technologicznej zostaną zrealizowane prace badawcze w następującym zakresie:

Moduł inference pokładowej predykcji zasięgu

- Przygotowano zestaw kontrolny 100 przejazdów testowych w różnych warunkach: miasto/trasa/mieszany, różne temperatury (-10C do +35C), różne profile nachylenia (płaskie/górskie), różne style jazdy (ekonomiczny/agresywny).
- Określono, że format danych wejściowych będzie miał strukturę: *.evt (EV telemetry) dla danych surowych z pojazdu.
- Wykonano 100 serii testów na zestawie kontrolnym. Algorytm wykazał pewne odchylenie w niektórych seriach testów (MAE wahał się od 8,2 do 12,7 km).
- Ustalono, że na wynik ma wpływ częstotliwość próbkowania danych GPS (1Hz vs 10Hz) oraz opóźnienia w pobieraniu danych pogodowych. Wprowadzono poprawkę do algorytmu (interpolacja GPS, cache pogody) i ponownie wykonano 100 serii testów. Wyniki w całej serii były porównywalne (MAE: $9,1 \pm 0,8$ km), a odchylenia zostały zminimalizowane.

Moduł walidacji zgodności predykcji z danymi referencyjnymi

- Przygotowano zestaw kontrolny 100 przejazdów z danymi referencyjnymi (rzeczywisty zasięg/zużycie energii) w formacie *.evr (EV reference).
- Wykonano po 100 serii testów dla zestawu kontrolnego. Algorytm wykazał pewne odchylenie w niektórych seriach testów (zgodność wahała się od 78% do 94%).
- Ustalono, że na wynik ma wpływ jakość danych referencyjnych (niepełne dane poziomu naładowania baterii, brak informacji o obciążeniu dodatkowym). Wprowadzono poprawkę do algorytmu (filtrowanie danych referencyjnych, estymacja obciążenia) i ponownie wykonano 100 serii testów. Wyniki w całej serii były porównywalne (zgodność: $89 \pm 3\%$), a odchylenia zostały zminimalizowane.

Moduł przetwarzania strumieniowego (czas rzeczywisty)

- Przygotowano zestaw kontrolny 100 długich tras (200-500 km) z nagrywaniem danych w czasie rzeczywistym.
- Wykonano po 100 serii testów dla zestawu kontrolnego. Podczas każdego testu monitorowano opóźnienie predykcji, zużycie CPU i pamięci RAM.

- Algorytm wykazał pewne odchylenie w niektórych seriach testów (latencja wahała się od 50ms do 200ms). Ustalono, że na wynik ma wpływ obciążenie systemu operacyjnego oraz kolejkovanie zadań. Wprowadzono poprawkę do algorytmu (priorytetyzacja, buforowanie) i ponownie wykonano 100 serii testów. Wyniki w całej serii były porównywalne (latencja: 85 ± 15 ms), a odchylenia zostały zminimalizowane.

Analiza wyników i wnioski

- Wykres zależności MAE od temperatury wykazał liniową korelację ($R^2 = 0,87$) - niższe temperatury zwiększają błąd predykcji.
- Formuła matematyczna:

$$MAE = 8,5 + 0,12 \times |T - 20C| + 0,05 \times G + 0,03 \times S \quad (2)$$

gdzie: T = temperatura zewnętrzna [C], G = średnie nachylenie trasy [%], S = wskaźnik agresywności stylu jazdy (0 = ekonomiczny, 1 = bardzo agresywny, obliczany jako wariancja przyspieszeń na 10km).

Metodologia wyznaczania współczynników:

- **Bazowy błąd (8,5 km):** Średni MAE z 30 przejazdów w warunkach referencyjnych ($20 \pm 2C$, nachylenie $< 2\%$, styl ekonomiczny) = $8,47 \pm 0,8$ km
- **Współczynnik temperatury (0,13):** Regresja liniowa MAE vs $|T - 20C|$ na 50 przejazdach w różnych temperaturach. Wzrost błędu o 0,13 km na każdy stopień odchylenia od 20C
- **Współczynnik nachylenia (0,05):** Analiza korelacji MAE vs średnie nachylenie na 40 trasach górskich. Każdy 1% nachylenia zwiększa błąd o 0,05 km
- **Współczynnik stylu (0,03):** Porównanie 20 par identycznych tras (ten sam kierowca, różne style). Różnica MAE między stylem agresywnym a ekonomicznym: 0,03 km na jednostkę wskaźnika S

Walidacja wzoru: Test na 20 nowych przejazdach dał średni błąd predykcji wzoru: 0,8 km (względem rzeczywistego MAE), co potwierdza jego adekwatność.

- Odkryto, że niekoniecznie wszystko jest takie wspaniałe jak zakładaliśmy pierwotnie - w ekstremalnych warunkach ($-15C$, strome podjazdy) MAE może przekraczać 15 km.
- W rozrachunku końcowym, badanie okazuje się sukcesem dla wdrożenia końcowego - średni MAE 9,1 km jest akceptowalny dla większości zastosowań.

Zadanie 2. Ryzyka dla IV poziomu TLR i koncepcje rozwiązań

Ryzyko 1: Niekompatybilność danych między modułami (prawdopodobieństwo: 35%)

- **Opis:** Różne formaty danych wejściowych/wyjściowych między modułami inference, walidacji i strumieniowego przetwarzania mogą powodować błędy komunikacji.

- **Rozwiązanie:** Wprowadzenie wspólnego schematu danych JSON/Protobuf z walidacją na poziomie API, implementacja testów automatycznych sprawdzających kompatybilność interfejsów między modułami.

Ryzyko 2: Niestabilność wydajności na różnych platformach sprzętowych (prawdopodobieństwo: 25%)

- **Opis:** Algorytm może działać inaczej na różnych jednostkach CPU/GPU w systemach pokładowych różnych producentów (Intel vs ARM, różne częstotliwości taktowania).
- **Rozwiązanie:** Rozszerzenie testów laboratoryjnych o różne platformy sprzętowe (x86, ARM, różne częstotliwości CPU), implementacja adaptacyjnego skalowania parametrów modelu w zależności od dostępnych zasobów obliczeniowych.

Ryzyko 3: Błędy w trybie offline (prawdopodobieństwo: 20%)

- **Opis:** Brak dostępu do danych pogodowych lub nachylenia trasy może prowadzić do znacznego pogorszenia jakości predykcji.
- **Rozwiązanie:** Implementacja inteligentnego cache'owania danych zewnętrznych z możliwością pracy na danych historycznych, wprowadzenie trybu degradacji z ostrzeżeniami o ograniczonej dokładności.

Ryzyko 4: Przeciążenie systemu w trybie strumieniowym (prawdopodobieństwo: 15%)

- **Opis:** Ciągłe przetwarzanie danych w czasie rzeczywistym może powodować przeciążenie CPU i opóźnienia w systemie pokładowym.
- **Rozwiązanie:** Implementacja adaptacyjnego throttlingu (zmniejszanie częstotliwości predykcji przy wysokim obciążeniu), wprowadzenie priorytetyzacji zadań z możliwością zawieszenia predykcji w krytycznych momentach.

Ryzyko 5: Błędy w walidacji danych referencyjnych (prawdopodobieństwo: 5%)

- **Opis:** Nieprawidłowe dane referencyjne mogą prowadzić do błędnych wniosków o jakości algorytmu.
- **Rozwiązanie:** Implementacja wielopoziomowej walidacji danych referencyjnych z detekcją anomalii, wprowadzenie systemu weryfikacji krzyżowej z danymi z różnych źródeł.

5 Lab 6 — Punkt Kontrolny

Zadanie 2. Grupa docelowa i komercjalizacja produktu

Grupa docelowa W skład grupy docelowej wchodzi **właściciele pojazdów elektrycznych** w wieku 30-50 lat, o średnich i wyższych dochodach (powyżej 5000 zł netto), którzy regularnie pokonują dystanse powyżej 30 km dziennie. W 65% są to mężczyźni, w 35% kobiety. Grupa charakteryzuje się wysoką świadomością ekologiczną i technologiczną, poszukuje rozwiązań optymalizujących eksploatację EV. Do tej grupy kierowana jest **aplikacja mobilna** z predykcją zasięgu.

Usługa B2B kierowana będzie do **producentów samochodów elektrycznych, flot korporacyjnych** (Uber, Bolt, firmy kurierskie) oraz **operatorów ładowarek** (Ionity, GreenWay, Orlen), którzy potrzebują dokładnych prognoz zasięgu do optymalizacji tras i planowania ładowań.

Model biznesowy (sprzedażowy) Model opiera się na **czterech filarach**:

Segment 1 - Aplikacja indywidualna (B2C): Freemium model - darmowa wersja podstawowa (5 prognoz dziennie) + **premium 19,99 zł/miesiąc** (nieograniczone prognozy + eco-driving tips). Szacowany rynek: 50.000 właścicieli EV w Polsce.

Segment 2 - Licencje OEM: Sprzedaż licencji technologii do systemów pokładowych producentom samochodów. **Oplata jednorazowa** za implementację: 200.000-500.000 zł + **opłata licencyjna** 50-150 zł za każdy wyprodukowany pojazd. Szacowany rynek: 15.000 nowych EV rocznie w Polsce.

Segment 3 - Subskrypcja B2B: Abonament miesięczny dla flot korporacyjnych: 25-80 zł/miesiąc za pojazd, w zależności od pakietu (podstawowy: predykcja zasięgu, premium: eco-driving + planowanie ładowań). Szacowany rynek: 5.000 pojazdów flotowych w Polsce.

Segment 4 - API dla operatorów ładowarek: Abonament miesięczny 200-500 zł/miesiąc za stację ładowania, w zależności od pakietu (podstawowy: predykcja zasięgu, premium: optymalizacja obciążenia + planowanie rozbudowy sieci). Szacowany rynek: 500 stacji ładowania w Polsce.

Wielkość rynku Rynek polski predykcji zasięgu EV: 15 mln zł (2024), prognoza wzrostu do 30 mln zł (2030) dzięki rosnącej liczbie EV i infrastrukturze ładowania.

Przychody roczne (szacowane):

- **Aplikacja B2C:** $50.000 \text{ użytkowników} \times 20\% \text{ konwersja} \times 19,99 \text{ zł} \times 12 \text{ miesięcy} = 2,4 \text{ mln zł/rok}$
- **Licencje OEM:** $3 \text{ producentów} \times 300.000 \text{ zł} + 15.000 \text{ pojazdów} \times 100 \text{ zł} = 3,9 \text{ mln zł/rok}$
- **Floty B2B:** $5.000 \text{ pojazdów} \times 50 \text{ zł} \times 12 \text{ miesięcy} = 3,0 \text{ mln zł/rok}$
- **API ładowarki:** $500 \text{ stacji} \times 350 \text{ zł/miesiąc} \times 12 \text{ miesięcy} = 2,1 \text{ mln zł/rok}$

Łączny potencjał przychodów: 11,4 mln zł/rok (2024) \rightarrow 30 mln zł/rok (2030)

Klienci docelowi:

- **Właściciele EV:** 50.000 osób w Polsce, potencjał 2,4 mln zł/rok
- **Producentów OEM:** 3-5 firm w Polsce, potencjał 3,9 mln zł/rok
- **Floty korporacyjne:** 200 firm w Polsce, potencjał 3,0 mln zł/rok
- **Operatorzy ładowarek:** 5-8 firm w Polsce, potencjał 2,1 mln zł/rok

6 Lab 7 — TLR poziom V

Projekt: Predykcja zasięgu EV z korekcją stylu jazdy

Zadanie 1. Weryfikacja koncepcji w środowisku zbliżonym do rzeczywistego

W celu połączenia zaprogramowanych komponentów zostanie zastosowany wzorzec architektoniczny MVC z ang. Model-View-Controller (pol. Model-Widok-Kontroler), który służy do organizowania struktury aplikacji. W tym celu zaprogramowany zostanie dodatkowy moduł Kontrolera. Jego zadaniem będzie koordynacja pracy aplikacji poprzez przyjmowanie danych i reagowanie na nie. Będzie on zarządzał aktualizacją modelu aplikacji oraz odświeżaniem widoków. W etapie zostaną przeprowadzone prace koncentrujące się na zweryfikowaniu komponentów i podstawowych podsystemów technologii w środowisku zbliżonym do rzeczywistego.

Kontroler – zostanie zaprogramowany celem koordynowania pracy pozostałych modułów. W opisanym etapie zostanie sprawdzone działanie kontrolera i przeprowadzone testy jego stabilności (poprzez jego włączanie i wyłączanie - błędy powodujące destabilizację pracy kontrolera będą korygowane do momentu, aż zostanie wykonany cykl bez błędów). Powyższe pozwoli zweryfikować, czy prawidłowo odbiera dane, dokonuje ich podstawowej identyfikacji i prawidłowo przesyła je do wybranych modułów. W kontrolerze zostaną zdefiniowane wtyczki do wszystkich zaprogramowanych wcześniej modułów oraz podstawowa weryfikacja danych (i ich typów) jakie mają wyjść do danego modułu. Zostanie także przetestowane działanie rozpoznawania danych dla poszczególnych modułów. W tym celu moduł zostanie poddany testom na danych kontrolnych dla każdej z wtyczek Kontrolera dedykowanej dla konkretnego modułu. Zostanie także dokonana weryfikacja reakcji modułu na dane celowo "uszkodzone" (np. zmieniony typ danych). W przypadku niewłaściwej reakcji zostanie dokonana analiza, która ustali przyczynę błędu. Następnie zostaną wprowadzone zmiany programistyczne w obszarze warunków weryfikujących zgodność danych. Pomyślny test prawidłowego działania kontrolera (tj. uzyskania zwrotu informacji o błędach w danych) pozwoli na jego integrację z wcześniej zaprogramowanymi modułami.

Moduł inference pokładowej predykcji zasięgu – dokonano integracji modułu z kontrolerem. Przetestowano działanie samego modułu oraz w relacji z kontrolerem (testu dokonano na próbnej bazie 1000 przejazdów w zakresie prawidłowej predykcji zasięgu). Analizie będzie podlegała także poprawność predykcji i kompletność danych wejściowych. Ewentualne błędy w opisanym zakresie zostaną wyeliminowane poprzez poprawki programistyczne i kolejne testy na tych samych danych (aż do poprawnego skomunikowania się z modułem kontrolera i poprawnego odczytu poleceń otrzymanych z Kontrolera).

Moduł walidacji zgodności predykcji z danymi referencyjnymi – dokonano integracji modułu z kontrolerem. Przetestowano działanie samego modułu oraz w relacji z kontrolerem (testu dokonano na próbnej bazie 1000 przejazdów z danymi referencyjnymi, w zakresie prawidłowej walidacji). Zweryfikowano jakość komunikacji danych i kompatybilność z pozostałymi współpracującymi modułami systemu. Przeprowadzono procedurę sprawdzającą jakość walidacji predykcji na plikach testowych. W przypadku błędów spowodowanych przeciążeniem kontrolera lub modułu walidacji zostaną one usunięte poprzez analizę wyników i dokonanie korekty programistycznej w kodzie. Następnie ponownie zostanie wykonany test na tych samych 1000 plikach.

Moduł przetwarzania strumieniowego danych – dokonano integracji modułu z kon-

trolerem. Przetestowano działanie samego modułu oraz w relacji z kontrolerem (testu dokonano na próbnej bazie 1000 długich tras w zakresie prawidłowej predykcji w czasie rzeczywistym). Zweryfikowano jakość predykcji w czasie rzeczywistym oraz stabilność działania przy różnych obciążeniach systemu. Sprawdzono jakość komunikacji danych i kompatybilność z pozostałymi współpracującymi modułami systemu. W przypadku błędów spowodowanych przeciążeniem kontrolera lub modułu odczytu strumieniowego zostaną one usunięte poprzez analizę wyników i dokonanie korekty programistycznej w kodzie. Następnie ponownie zostanie wykonany test na tych samych 1000 trasach.