

Dokumentace k výsledku typu software (R)

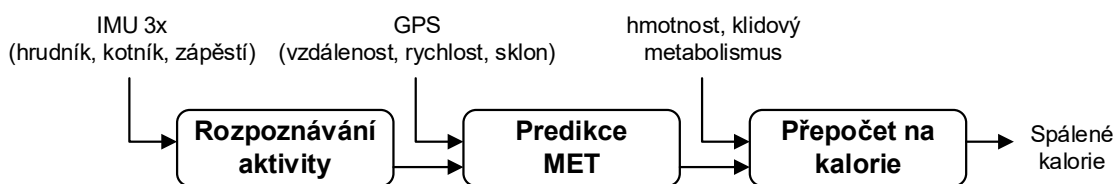
Přesné určení spálených kalorií z parametrů nositelných zařízení

- Autoři:** Lukáš Smital, Radovan Smíšek, Andrea Němcová, Martin Králík, Daniela Chlíbařová, Jana Kolářová, Vojtěch Myška, Jakub Arm, Michal Harvánek, Martin Kolařík, Jaromír Hubálek
- Projekty:** H2020 ECSEL JU, ID: 8A20007 – Next generation smart perception sensors and distributed intelligence for proactive human monitoring in health, wellbeing, and automotive systems (NextPerception)
- Součást:** RG-1-02, CEITEC, VUT Brno
- Datum:** 30.5.2023
- Popis:**

Algoritmus byl vyvinut v rámci tohoto projektu na základě určování tzv. MET jednotek (Metabolic equivalent), což je bezrozměrná jednotka vyjadřující násobek klidového metabolismu při určitém typu a intenzitě dané aktivity. Znalost rozložení jednotek MET v čase umožňuje odhadnout celkový počet kalorií spálených během aktivity. Předpokladem je znalost hmotnosti a rychlosti klidového metabolismu daného jedince.

Metodologie

Navržený software pro určení spálených kalorií se skládá ze tří funkčních částí, které na sebe navazují. Jedná se o rozpoznávání aktivit, predikce míry fyzické zátěže MET, přepočítání MET jednotek na spotřebované kalorie. Náporné blokové schéma je na Obrázku 1. V následující části Popisu algoritmu je každý z těchto bloků podrobně popsán.



Obrázek 1: Blokové schéma algoritmu pro odhad spálených kalorií.

Rozpoznávání aktivity je důležitý z pohledu výpočtu MET, které závisí na druhu aktivity, každá aktivita má jinou predikční rovnici. Samotná predikce využívá vedle času také vzdálenost a sklon (inklinace) trasy, po které se uživatel pohybuje. Rychlost se počítá z času a vzdálenosti. Samotný algoritmus přepočítání kalorií je korigován ještě vzhledem ke stáří, pohlaví, hmotnosti a klidového metabolismu, který je nutné nejprve stanovit z těchto zadaných parametrů.

Popis algoritmu

Rozpoznávání aktivity

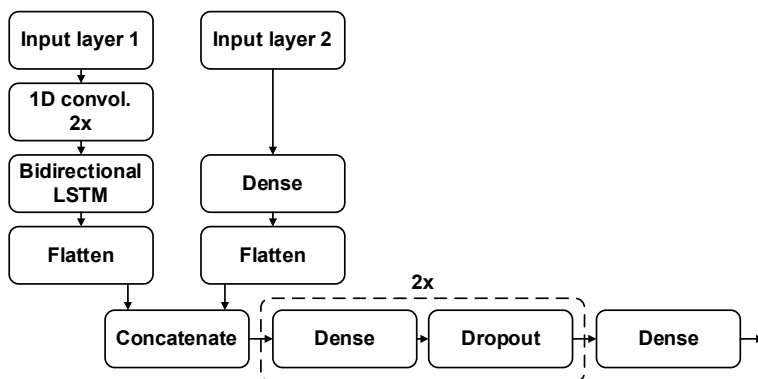
Algoritmus klasifikace aktivit analyzuje 10-ti sekundové datové úseky, které obsahují nepracovaná data ze 3 tříosých akcelerometrů. Vzhledem k záměně činností, jako je chůze a

běh, bylo rozhodnuto extrahovat příznaky ze surových dat a použít je společně se surovými daty v procesu klasifikace. Extrakce příznaků se provádí tak, aby se získaly následující skupiny příznaků (počet příznaků):

- statistické (5),
- časová oblast (6),
- frekvenční oblast (9).

Celkem je z jednoho úseku dat pro každou osu akcelerometru extrahováno 22 příznaků. Tyto příznaky jsou analyzovány společně s nezpracovanými daty. Navržený model více vstupové neuronové sítě je na Obrázku 2. K implementaci modelu rozpoznávání aktivity byly použity Tensorflow, programovací jazyk Python, Keras (vysokoúrovňové API Tensorflow pro rychlý vývoj modelů neuronových sítí), Pandas a další knihovny.

Výstupem algoritmu pro rozpoznávání aktivity je klasifikace 10-ti sekundových záznamů do čtyř tříd: klid, chůze, běh a jízda na kole.



Obrázek 2: Navržený model více vstupové neuronové sítě pro úlohu rozpoznávání aktivity.

Predikce MET

Jako metriku fyzické zátěže jsme zvolili jednotky MET (Metabolic equivalent). Jednotka 1 MET odpovídá množství spotřebovaného kyslíku v klidovém režimu (sed, leh) a je přibližně rovna 3,5 ml na kilogram váhy za minutu. Stupeň fyzické námahy je tedy reprezentován násobkem klidového metabolického ekvivalentu a jedná se tudíž o bezrozměrnou jednotku.

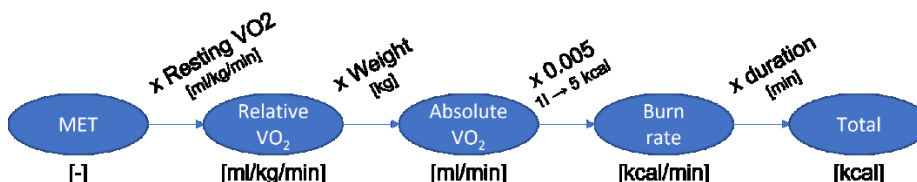
Vstupem algoritmu pro odhad MET jednotek je klasifikovaná aktivita měřeného subjektu a rychlost subjektu, překonaná vzdálenost a převýšení měřena pomocí GPS. Samotný algoritmus pro odhad MET jednotek je založen na predikčních rovnicích, které jsou přizpůsobeny individuálně pro každou prováděnou aktivitu. Tyto predikční rovnice jsou známé a v dostupné literatuře publikované např. v „ACSM's Metabolic Calculations Handbook“ vydané na Americké akademii sportovní medicíny.

Výstupem algoritmu je odhadnutá hodnota MET pro každých 10 sekund měřeného záznamu.

Přepočítání na kalorie

Znalost rozložení jednotek MET v čase umožňuje odhadnout celkový počet kalorií spálených během aktivity. Postup výpočtu je schematicky naznačen na Obrázku 3. Nejprve se bezrozměrná jednotka MET vynásobí klidovou spotřebou kyslíku daného subjektu a jeho váhou. Jestliže není klidová spotřeba kyslíku známá nahrazuje se (bez významné újmy na přesnosti odhadu)

hodnotou 3,5 ml na kilogram váhy za minutu. Dostáváme se k absolutní spotřebě kyslíku daného subjektu v jednotce ml za minutu. K rychlosti spalování kalorií potřebujeme vědět, že každý spotřebovaný litr kyslíku odpovídá 5 spáleným kilokaloriím. Nyní už stačí rychlost spalování kalorií integrovat požadovanou dobu a dostaneme se k celkovému objemu spálených kalorií.

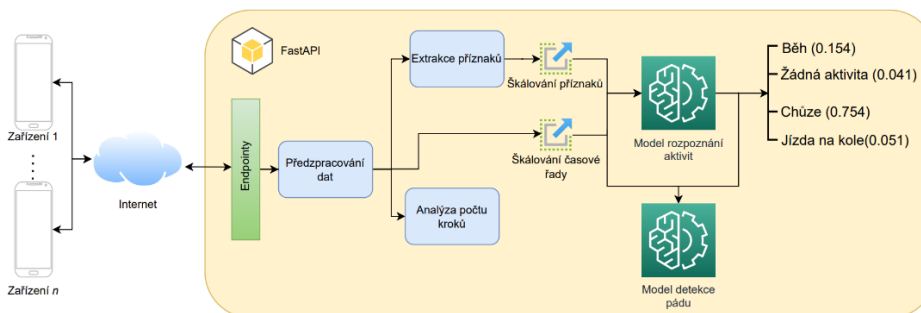


Obrázek 3: Schematické znázornění výpočtu spálených kalorií ze spočítané hodnoty MET jednotek.

Na konci aktivity (nebo více aktivit během dne) bude mít měřená osoba přehled a záznam nejen o tom, jaké aktivity vykonával, ale také jakou intenzitou a především kolik kalorií skutečně jeho tělo spotřebovalo.

Softwarové řešení

Služba rozpoznání aktivit je implementována pomocí frameworku FastAPI, viz Obrázek 4. Jedná se o moderní a rychlý webový framework pro vytváření programovatelného aplikačního rozhraní (API) v Pythonu. Tento výběr byl učiněn kvůli snadné integraci vyvinutých AI algoritmů, které byly vytvořeny a trénovány pomocí knihovny TensorFlow.



Obrázek 4: Architektura služby rozpoznání aktivit.

Ve frameworku FastAPI jsou vytvořeny endpoints (Obrázek 5) pro zpracování příchozích požadavků z mobilních zařízení. Tyto požadavky přicházejí ve formě JSON zpráv, z nichž se extrahují surová data z akcelerometrů. Služba obsahuje celkem pět různých koncových bodů. Na základě typu příchozího požadavku na specifický endpoint je vybrána odpovídající verze algoritmů pro strojové učení. Tato modularita a schopnost výběru různých verzí algoritmů je klíčová pro zajištění zpětné kompatibility během vývoje služby pro rozpoznávání aktivit a integrace s mobilními aplikacemi.

Metoda `predict_activity_v5` (Obrázek 6) je určena pro predikci aktivit a detekci pádů v rámci systému pro rozpoznávání pohybových aktivit. Je volána z koncového bodu `/v5/predict` a začíná inicializací datové struktury pro uchování předzpracovaných vstupních dat z akcelerometru a jejich převodem do numpy pole. Následuje využití předem definovaných modelů a škálovačů (pro časové řady i vlastnosti) specifických pro verzi 5 AI modelů. Součástí je také extrakce statistických dat popisujících surová data.

```
@app.post("/v5/predict")
async def predict(request_data: Request):
    request_data = await request_data.json()

    df_preprocessed_data, faros_present = data_processing.preprocess_data_faros_model(request_data)

    steps = postprocess_steps(step_counter.count_steps(df_preprocessed_data))

    probabilities, predicted_activity, fall_probabilities, fall_detected = models.predict_activity_v5(df_preprocessed_data, steps)

    return {
        "status": "SUCCESS",
        "activity_probabilities": probabilities[0],
        "predicted_activity": predicted_activity,
        "step_counted": steps,
        "fall_probabilities": fall_probabilities,
        "fall_detected": str(fall_detected)
    }
```

Obrázek 5: Endpoint /v5/predict služby rozpoznání aktivit.

Výsledkem predikce jsou pravděpodobnosti různých aktivit, z nichž je vybrána ta s nejvyšší pravděpodobností. Kromě toho, pokud je klasifikována jiná aktivita než jízda na kole, tak se provádí detekce pádu pomocí speciálního AI modelu.

Metoda nakonec vrací seznam pravděpodobností aktivit, identifikovanou aktivitu, pravděpodobnosti pádu a boolean hodnotu indikující, zda byl pád detekován.

Predikce MET jednotek a výpočet celkových spálených kalorií probíhá na straně serveru a je implementována v programovém prostředí Matlab. MET jednotky jsou počítány průběžně v 10 s oknech a tuto konkrétní službu zajišťuje funkce „*METandCALfromGPS*“, jejíž náhled je možné vidět na Obrázku 7. Vstupem funkce jsou veličiny odvozené z GPS senzoru, které se vztahují pro konkrétní 10 s okno (průměrná rychlost, průměrná nadmožská výška, překonaná nadmožská výška, uražená vzdálenost), dále rozpoznaná aktivita a také personální informace jako je hmotnost a hodnota klidového metabolismu. Pro výpočet MET je volána samostatná funkce „*METfromGPS*“, která realizuje samotné predikční rovnice pro konkrétní rozpoznanou aktivitu a její náhled je na Obrázku 8.

Výpočet spálených kalorií je realizován přepočtem z MET jednotek dle postupu na Obrázku 3 a je názorně vidět na posledním řádku kódu na Obrázku 7. Výsledkem je časový vektor *CAL_GPS* vzorkovaný každých 10 s, který popisuje rychlost spalování kalorií a jeho vývoj v čase. Prostou integrací tohoto vývoje je možné zjistit celkové množství spálených kalorií.

```
def predict_activity_v5(self, request_data, steps):
    data = np.zeros((1, 200, request_data.shape[1])).astype('float32')
    data[0] = request_data.to_numpy()

    model_v5 = self.model_v5
    timeseries_scaler_v5 = self.timeseries_scaler_v5
    features_scaler_v5 = self.features_scaler_v5

    activity = activ_dict_v2

    features = self.data_processing.extract_features_v5(data[0], features_scaler_v5, steps)
    raw_data = self.data_processing.scale_raw_data_v5(data[0], timeseries_scaler_v5)

    data = [raw_data, features]

    prediction = model_v5.predict(data)

    activity_probabilities = prediction
    activity_idx = np.argmax(activity_probabilities, axis=1)[0]

    fall_probabilities = [1.0, 0.0]
    fall_detected = False
    if activity_idx < 3:
        data = np.zeros((3, 200, request_data.shape[1])).astype('float32')
        data[0] = request_data.to_numpy()

        model_fall = self.fall_detection_v4
        fall_probabilities = model_fall.predict([raw_data, features[:, :, :88]])
        fall_detected = (np.argmax(fall_probabilities, axis=1) == 1)[0]
        fall_probabilities = fall_probabilities.tolist()

    return activity_probabilities.tolist(), activity[activity_idx], fall_probabilities, fall_detected
```

Obrázek 6: Metoda zajišťující predikci aktivity a pádu na základě vstupních dat.

```
function [MET_GPS, CAL_GPS] = METandCALfromGPS(avgSp, avgAlti, difAlti, difDist, okno, anAkt, RestVO2, Weight)
% Odhad MET na základě typu aktivity, GPS, klidového metabolismu a hmotnosti
N = length(avgSp);
MET_GPS = zeros(N,1);
METlast = 1; iexp = 0;
for i = 1 : N % vytvoření vektoru pro typy aktivit podle času
    [MET_GPS(i), iexp] = METfromGPS(avgSp(i), avgAlti(i), difAlti(i), difDist(i), anAkt(i), RestVO2, Weight, METlast, iexp);
end
CAL_GPS = MET_GPS * RestVO2 * Weight * 0.005 * okno/60;
end
```

Obrázek 7: Funkce implementovaná v Matlabu pro predikci MET jednotek a spálených kalorií.

```

function [MET, iexp] = METfromGPS (SPD, alt, ALT, DIS, ACT, RestVO2, Weight, METlast, iexp)
% MET: Metabolic equivalent
% iexp: pořadí zvoleného časového okna v 'rest'
% SPD: Průměrná rychlost v okně [km/h]
% alt: Průměrná nadmožská výška v okně [m]
% ALT: Změna nadmožské výšky v okně [m]
% DIS: Uražená vzdálenost v okně [km]
% ACT: Typ aktivity {'chůze', 'běh', 'kolo', 'rest'}
|
lambda = 0.035;           % [-] slope of the exponential function
RestMET = 1;             % Hodnota MET v klidu
incline = ALT/(DIS*1000); % Stoupání [%]
switch ACT
case 'chůze'
    spd_walk = SPD * (1000/60); % Speed in [m/min]
    MET = (0.1*spd_walk + 1.8*spd_walk*incline + RestVO2)/RestVO2; iexp = 0;
case 'běh'
    spd_runn = SPD * (1000/60); % Speed in [m/min]
    MET = (0.2*spd_runn + 0.9*spd_runn*incline + RestVO2)/RestVO2; iexp = 0;
case 'kolo'
    spd_cycl = SPD * (1000/60); % Speed in [m/min]
    RV = 0; % Rychlost Větru [m/s]
    g = 9.80665; % Gravitační konst. [m/s²]
    m = 14; % Hmotnost kola [kg]
    loss = (1.5 + 3)/100; % Ztráty [%] (1.5 = konst + X = podle řetězu 3-oil; 4-dry; 5-old)
    Crr = 0.0050; % Crr is the rolling resistance coefficient.
    Cd_A = 0.408; % position Cd * A
    ro = 1.225 * exp(-0.00011856 * alt); % Hustota vzduchu v závislosti na nadmožské výšce
    Fg = g * sin(atan(incline)) * (Weight + m); % Gravitační složka
    Fr = g * cos(atan(incline)) * (Weight + m) * Crr; % Valivý odpor
    Fa = 0.5 * Cd_A * ro * (spd_cycl/60 + RV).^2; % Odpor vzduchu
    Watt = ((Fg + Fr + Fa) .* (spd_cycl/60)) / (1-loss); % Výpočet watů
    Watt(Watt<0) = 0; % Korekce watů pro jízdu z kopce
    MET = (1.163 * Watt/Weight * 1/0.24)+1; iexp = 0;
otherwise
    iexp = iexp + 1;
    MET = RestMET + (METlast-RestMET)*exp(-lambda*iexp);
end

```

Obrázek 8: Funkce implementovaná v Matlabu, predikční rovnice MET jednotek.

Validace a dosažená přesnost

Validaci dosažených výsledků prezentujeme ve dvou kapitolách. V první je samostatně zhodnocen algoritmus pro klasifikaci aktivit, druhá kapitola popisuje výsledky dosažené při odhadu jednotek MEA a celkových spálených kaloriích.

Rozpoznávání aktivity

Dosažená přesnost klasifikace aktivit dosahovala při trénování asi 93 %, na testovací sadě dat přesnost klasifikace přesáhla hodnotu 95 %, viz Tabulka 1.

Tabulka 1: Výkonnost AI modelu pro klasifikaci aktivit.

Trénovací přesnost [%]	Validační přesnost [%]	Testovací přesnost [%]
93,2	98,1	95,4

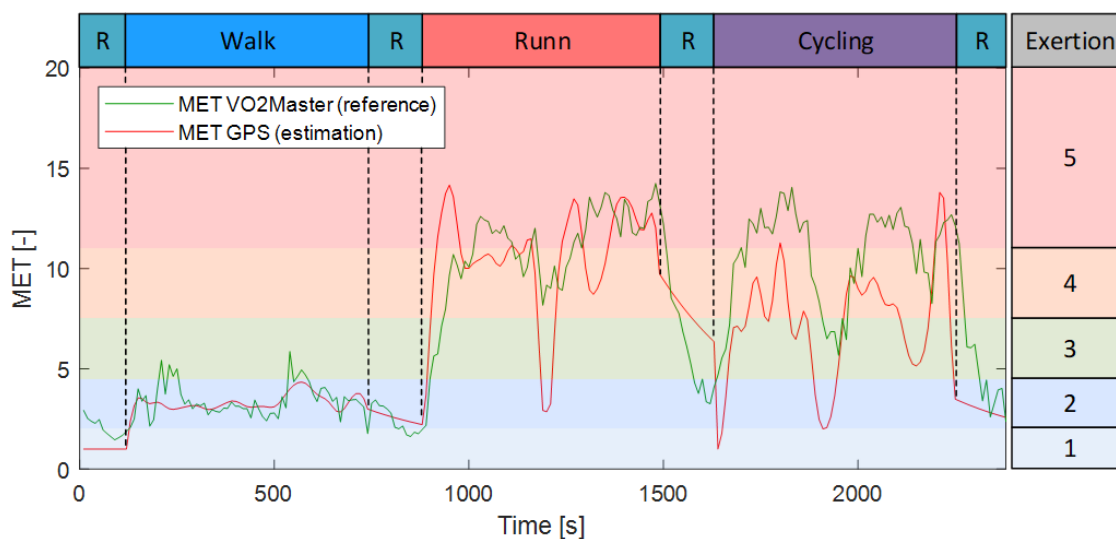
Zaměřili jsme se také na experimentování s korekčním mechanismem, který využívá metodu klouzavého okna. Tento princip je založen na kombinaci jednoduché statistiky a logiky. Korekční proces spočívá v analýze posledních n klasifikovaných datových úseků. Pokud dojde k výskytu izolovaných odlišných úseků v rámci těchto dat, systém provede korekci na nejčastěji se

vyskytující aktivitu v rámci klouzavého okna. Jako příklad lze uvést sekvenci klasifikovaných aktivit, například: běh, běh, jízda na kole, běh, běh. V této situaci je zřejmé, že jeden úsek aktivity byl chybně klasifikován. V našem příkladu by korekční algoritmus opravil aktivitu z 'jízda na kole' na 'běh'. Takováto korekce vedla ke zvýšení přesnosti algoritmu o 0,5 % v trénovací datové množině a až o 2,7 % v testovací datové množině.

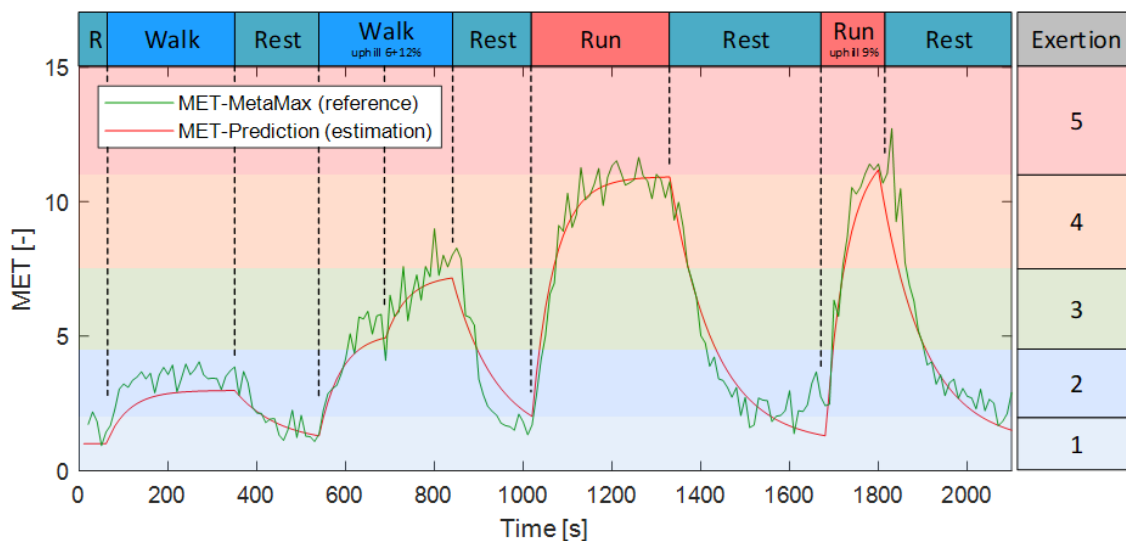
Predikce MET a spálených kalorií

Validace přesnosti predikce MET a spálených kalorií byla provedena měřeními na dobrovolnících při dvou typech protokolů přizpůsobených pro venkovní a vnitřní měření. Pro validaci byly jako reference použity dvě obličejové masky se senzorem a analýzou vydechovaných plynů. VO2 Master pro venkovní a MetaMax pro vnitřní měření. Tyto masky jsou schopny v reálném čase stanovit množství/rychlost spotřeby kyslíku a mimo jiné zaznamenat průběh MET v závislosti na čase a celkové množství spálených kalorií.

Porovnání naměřené referenční hodnoty MET a naším algoritmem spočítané hodnoty MET pro vybranou osobu je vidět na Obrázcích 9 (venkovní měření) a 10 (vnitřní měření). Venkovního měření se účastnilo 20 dobrovolníků a průměrná absolutní chyba (MAE) odhadu dosahovala 0,67 MET. Vnitřního měření se účastnilo 44 dobrovolníků s MAE 0,63 MET.



Obrázek 9: MET jednotky stanovené naším algoritmem (červeně) a referenční hodnoty (zeleně) při aktivitách prováděných ve venkovních prostorách.



Obrázek 10: MET jednotky stanovené naším algoritmem (červeně) a referenční hodnoty (zeleně) při aktivitách prováděných ve vnitřních prostorách.

V rámci stejného měřicího protokolu jsem validovali také množství celkových spálených kalorií. Pro venkovní měření (na 20 dobrovolnících) jsme zaznamenali průměrnou absolutní odchylku kolem 9 %. Pro vnitřní měření (na 44 dobrovolnících) jsme dosáhli MAE pouhých 7 %. Tento výsledek ukazuje přesnost, kterou nedosahuje většina nositelných zařízení dostupných na trhu.