



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

DEPARTMENT OF INTELLIGENT SYSTEMS

**INTELIGENTNÍ SENZOROVÁ SÍŤ S VYUŽITÍM MO-
BILNÍCH ZAŘÍZENÍ**

SMART SENSOR NETWORK WITH USING MOBILE DEVICES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MILAN TOMČÍK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAN SAMEK, Ph.D.

BRNO 2017

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav inteligentních systémů

Akademický rok 2016/2017

Zadání bakalářské práce

Řešitel: **Tomčík Milan**

Obor: Informační technologie

Téma: **Inteligentní senzorová síť s využitím mobilních zařízení**
Smart Sensor Network with Using Mobile Devices

Kategorie: Umělá inteligence

Pokyny:

1. Prostudujte a popište problematiku využití mobilních zařízení jako snímačů (senzorů) různých veličin (využití gyroskopu, akcelerometru, kompasu, GPS, atd.). Popište především různé typy snímačů a jejich možností v současných mobilních zařízeních.
2. Navrhněte aplikaci, která s využitím snímačů mobilního zařízení (anebo více zařízení a fúzí dat z jejich snímačů) vhodně demonstuje možnosti využití mobilních zařízení jako uzlu senzorové sítě.
3. Vámi navrženou aplikaci implementujte.
4. Výslednou aplikaci otestujte v praxi a vyhodnoťte efektivitu a možnosti využití vaší implementace.

Literatura:

- RUIZ, David Cuartielles. *Professional android wearables*. ISBN 1118986857.
- Building Apps for Wearables: <https://developer.android.com/training/building-wearables.html>

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Samek Jan, Ing., Ph.D., UITS FIT VUT**

Datum zadání: 1. listopadu 2016

Datum odevzdání: 17. května 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav inteligentních systémů
602 00 Brno, Božetěchova 2

doc. Dr. Ing. Petr Hanáček
vedoucí ústavu

Abstrakt

Táto bakalárska práca sa zaoberá problematikou využitia mobilných zariadení ako snímačov rôznych aktivít. Využívajú sa rôzne typy snímačov mobilného zariadenia a to predovšetkým gyroskopu, akcelerometru a magnetometru. Práca sa zaoberá možnosťou využitia mobilných zariadení ako súčasťou uzlov sensorovej siete, pričom pri tom využíva mobilné zariadenie a fúziu dát z ich snímačov.

Abstract

This bachelor thesis deals with the use of mobile devices as sensors of various activities. Different types of mobile device sensors are used, in particular the gyroscope, accelerometer and magnetometer. The thesis tackles the possibility of utilization of mobile devices as part of the nodes in a sensor network, while employing mobile device and fusing data obtained from their sensors.

Kľúčové slová

Mobilné zariadenia, Snímač, Fúzia dat, Sensorová sieť, Weka software, J48 algoritmus, Rozhodovacie stromy, Strojové učenie

Keywords

Mobile devices, Sensor, Fusion of data, Sensor network, Weka software, J48 algorithm, Decision trees, Machine learning

Citácia

TOMČÍK, Milan. *Inteligentní sensorová síť s využitím mobilních zařízení*. Brno, 2017. Bakalárska práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jan Samek, Ph.D.

Intelligentní senzorová síť s využitím mobilních zařízení

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Jana Samka Ph.D. Uviedol som všetky literárne zdroje a publikácie z ktorých som čerpal.

.....

Milan Tomčík

17. mája 2017

Podakovanie

Týmto by som sa chcel poďakovať Ing. Janu Samkovi Ph.D za vedenie, pomoc a odborné rady pri písaní a riešení problémov, ktorými sa moja bakalárska práca zaoberá.

Obsah

1	Úvod	3
2	Senzory v mobilných zariadeniach	4
2.1	Akcelerometer	5
2.2	Gyroskop	6
2.3	Krokový senzor	7
2.4	Barometer	8
2.5	Svetelný senzor	9
2.6	Senzor priblíženia	9
2.7	Tepelný senzor	10
2.8	Senzorová fúzia	11
2.9	Chybovosť senzorov mobilných zariadení	13
3	Návrh aplikácie	14
3.1	Analýza aplikácie	15
3.1.1	Ciele aplikácie	16
3.2	Zpracovanie dát	17
3.2.1	Odstránenie šumu	17
3.2.2	Segmentácia dát	18
3.2.3	Súbor ARFF	19
3.3	Senzorová fúzia	19
3.4	Extrakcia príznakov	21
3.5	Nástroj WEKA	22
3.6	Klasifikačný algoritmus	23
3.6.1	Rozhodovacie stromy	24
4	Implementácia	26
4.1	Prístup k senzorom	27
4.2	Komplementárna senzorová fúzia	28
4.3	Nástroj WEKA	29
4.4	Predikcia	30
5	Testovanie	31
5.1	Efektivita a využitie aplikácie	32
6	Záver	34
	Literatúra	35

A	Obsah CD	37
B	Manuál	38
	B.1 Postup inštalácie aplikácie	38
	B.2 Postup preloženia aplikácie	38
C	Súbor typu ARFF	39

Kapitola 1

Úvod

Ako už vyplýva z názvu bakalárskej práce, táto práca sa zameriava na využitie senzorov, ktoré sa stali už neodmysliteľnou súčasťou mnohých moderných mobilných zariadení a aplikácií, ktoré bežne používame v každodennom živote bez toho aby sme si vôbec uvedomili, že ich neustále používame. Na začiatku sa práca sústreďuje na oboznámenie čitateľa s rôznymi druhmi senzorov implementovaných v mobilných zariadeniach, aké užitočné dáta poskytujú pre užívateľov a príklad ako sa dajú využiť v dnešnej dobe. Ďalej sa bude práca zaoberať fúziou dát zo senzorov a ako sa dá využiť pri demonštrovaní možnosti využitia mobilných zariadení ako uzlov sensorovej siete. Cieľová mobilná platforma pre ktorú bude aplikácia, algoritmus vyvíjaný bude mobilný operačný systém Android. Neskôr sa budeme ďalej v práci venovať, návrhu aplikácie, algoritmu, problematike spojeného s návrhom tejto aplikácie a jeho následné navrhnuté riešenie. Nasleduje implementácia navrhnutej aplikácie. V poslednej kapitole sa oboznámime s výslednou implementovanou aplikáciou, vyhodnotí sa jej efektivita a budeme sa zaoberať prípadnými ďalšími možnosťami využitia danej implementácie.

Kapitola 2

Senzory v mobilných zariadeniach

V tejto kapitole sa budeme venovať súčasným senzorom v mobilných zariadeniach, aké druhy senzorov existujú, aké užitočné dáta poskytujú pre vývojárov aplikácií, budeme sa zaoberať aj ich presnosťou, či chybami pri meraní. Taktiež si uvedieme príklad, ako sa využíva daný druh senzora v mobilnom zariadení.

So senzormi sa v dnešnej dobe stretneme už s pomaly každým mobilným zariadením. Moderné mobilné zariadenia prichádzajú s rôznymi druhmi senzorov, ktoré automatizujú a uľahčujú viacero z našich bežných každodenných činností. Pod pojmom snímač s predstavíme technické zariadenie reagujúce na zmeny meranej veličiny, alebo taktiež reaguje na časový priebeh zmeny. Elektronické senzory, ktoré sa používajú v mobilných zariadeniach transformujú informáciu o meranej veličine na elektrický signál alebo elektrický parameter. Umožňujú teda transformáciu neelektrickej veličiny na elektrické signály. Senzory sa taktiež využívajú v medicíne pri lekárskech prístrojoch, v priemysle pri priemyselnej výrobe ale aj v automobiloch a robotike. Medzi základné druhy senzorov, ktoré sa najčastejšie vyskytujú v mobilných zariadeniach patrí:

- Akcelerometer,
- gyroskop,
- barometer,
- krokový senzor,
- svetelný senzor,
- tepelný senzor,
- senzor priblíženia.

Ako sa dočítame v [1] každý druh senzora má iné využitie v zariadení. Aby sme mohli s týmito senzormi v zariadeniach s operačným systémom Android pracovať, poskytuje Android SDK podporu, ktorá sa nachádza v Java balíku `android.hardware`. Tento Java balík sa skladá z rozhraní a tried. Medzi základné triedy v tomto balíku patria: **SensorManager**, **Sensor**, **SensorEvent**, **SensorEventListener**. Jednotlivé triedy poskytujú pre vývojárov rôzne metódy, ako napríklad trieda **SensorManager**, ktorá poskytuje metódu `getSensorList(int type)`, pomocou ktorej vieme zistiť zoznam senzorov, ktoré sú dostupné na mobilnom zariadení s operačným systémom Android. Ďalej sa budeme venovať jednotlivým druhom senzorom a ich možnosti využitia v dnešnej dobe.

2.1 Akcelerometer

Akcelerometer je jeden z najpoužívanějších druhov senzorov v mobilnom zariadení, ktorý je určený k meraniu zrýchlenia, pričom je navrhnutý tak, aby pri zmene z konštantnej alebo nulovej rýchlosti vedel zaznamenať túto zmenu. Jeho časté využitie najdeme teda pri detekcii pohybu, pri rozpoznávaní polohy mobilného zariadenia a podľa nej otáčanie obrazu na displeji zariadenia, ale taktiež aj v hrách pre mobilné zariadenia, kde sa pomocou nakláňania zariadenia dokáže ovládať hra bez použitia tlačítok. Mobilné zariadenia so zabudovaným akcelerometrom sú teda schopné pohodlne a objektívnym spôsobom kvantifikovať každodenné pohyby a rozpoznať podľa týchto pohybov, aká fyzická činnosť práve prebieha. [7]

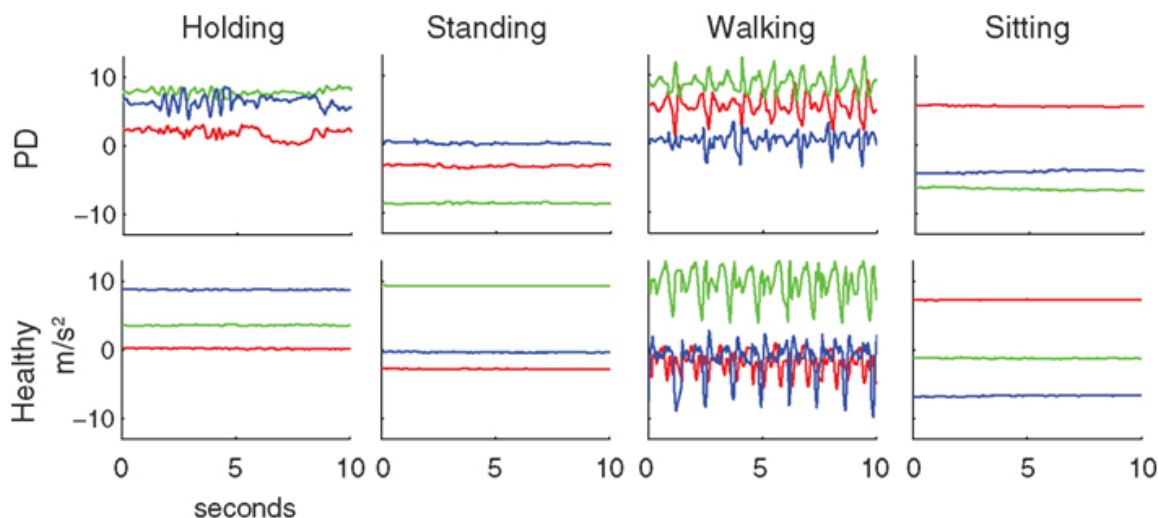
Akcelerometer sa využíva napríklad aj v medicíne, presné sledovanie pacientov s obmedzenými pohybovými schopnosťami má vysoký potenciál lepšie informovať oddelenie starostlivosti o pacientov. S presnejšími a objektívnejšími meraniami liečivé alternatívy môžu byť vyhodnotené s konečnou platnosťou. Obzvlášť je to veľmi dôležité pri pacientoch s notorickým postihnutím ako je Parkinsonova choroba¹, ktorá reaguje na rastúce možnosti liečby, vrátane liekov a rôznych pohybových cvičení. Momentálne metódy určenia pohyblivosti pacientov sú obmedzené. Vyžaduje sa aby pacient vycestoval na miesto, kde je poskytovaná zdravotná starostlivosť pretože klinické testy² sú nákladné z hľadiska peňazí a času. To obmedzuje frekvenciu klinického hodnotenia iba niekoľkokrát do roka pre osobnú potrebu a iba pár krát v rámci niekoľkých týždňov, kedy prebiehajú klinické štúdie. Aby sa klinické štúdie neobracali na pacienta s dôvodom aby pravidelne do denníka zapisovali svoje činnosti ktorými trpí s problémami je výhodnejšie na monitorovanie pohybových aktivít pacienta využiť akcelerometer. Mobilné zariadenie so zabudovaným senzorom akcelerometra je možné pripevniť na pacienta a tým automaticky monitorovať a zbierať dáta o ich pohybových aktivitách, lepšie sledovať reakciu pacienta na liečbu a liečebný postup. Pre budúcnosť monitorovania bol uskutočnený test, kedy senzor akcelerometra monitoroval pacientov s Parkinsonovou chorobou a zdravých jedincov pri štyroch pohybových aktivitách a to pri prechádzaní sa, státim na mieste, sedením a držaním predmetu. Čiary znázornené červenou, zelenou a modrou farbou v grafe znázornenom na obrázku 2.1 znamenajú súradnice x,y a z v osi zrýchlenia. Pri nahliadnutí na graf je možné si všimnúť u pacienta s Parkinsonovou chorobou jej príznaky ako je napríklad nepravidelná chôdza, Dyskinéza³ v paži, ktorá je viditeľná pri držaní predmetu, ale taktiež pri státi a aj sedení pacienta s Parkinsonovou chorobou. [7]

Namerané dáta z monitorovania pacientov sú ľahko prenositeľné do zdravotníckych zariadení a je možné monitorovať zdravie pacienta pohodlne z jeho domova, ako aj jeho reakciu na novú liečebnú metódu, liek. Tieto dáta môžu zefektívniť postup a vývoj v liečbe pacientov s pohybovým postihnutím a urýchliť tým vynájdenie nového efektívnejšieho lieku alebo liečebnej metódy, ktorá by aspoň minimalizovala nepriaznivé následky postihnutia na pacienta. [7]

¹Parkinsonova choroba je neurodegeneratívna choroba centrálnej nervovej sústavy

²Klinické testy sú séria výzkumných štúdií, ktorých hlavnou úlohou je overenie bezpečnosti a účinnosti nového lieku, liečebného postupu

³Dyskinéza je pohybová porucha, charakterizovaná nedobrovoľnými pohybmi svalov



Obr. 2.1: Graf znázorňujúci dáta pacientov s Parkinsonovou chorobou pri rôznych pohybových aktivitách. [7]

2.2 Gyroskop

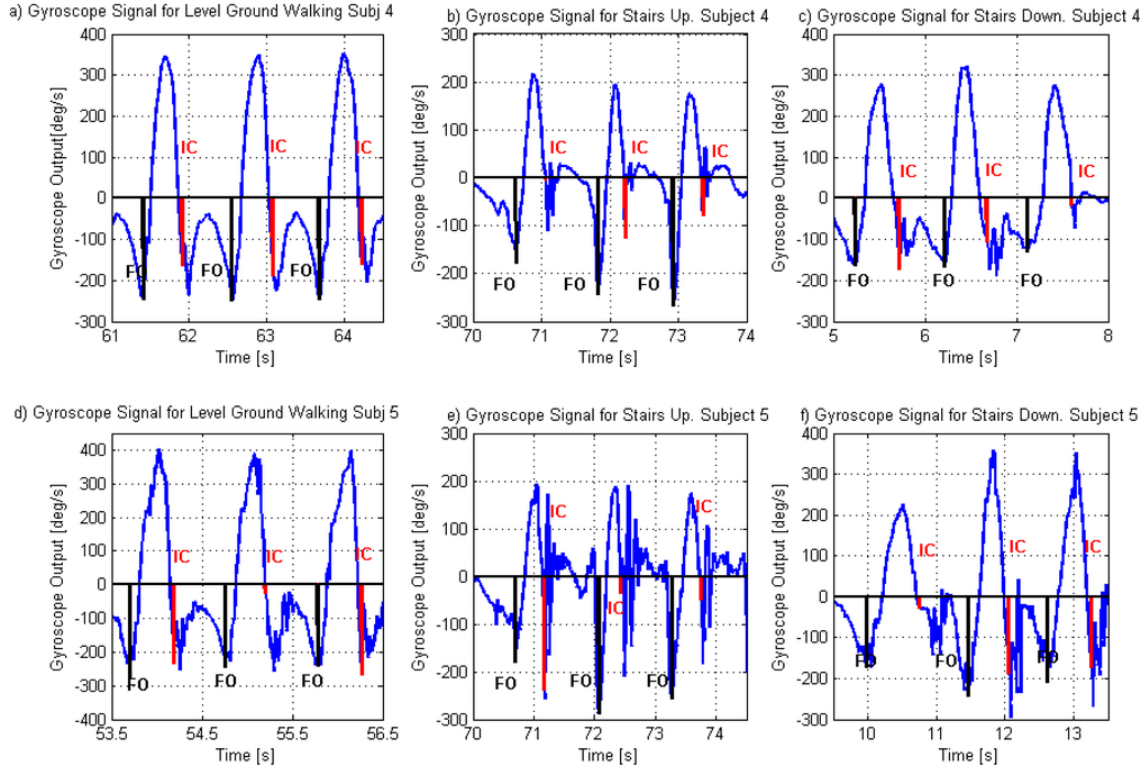
Gyroskop je jeden z ďalších senzorov, ktoré mobilné zariadenie v sebe obsahuje, pričom je primárne určený na udržiavanie rovnakej orientácie, respektíve rovnakého smeru. Tento senzor využíva zákon zachovania momentu hybnosti⁴. Jednou z hlavných súčastí gyroskopu nazývame zotrvačník diskového tvaru, ktorý má ramená upevnené k trom pohyblivým rámom prstencového tvaru, ktoré rotujú okolo zotrvačníka. Ak by bol zotrvačník roztočený, dostáva zvláštnu vlastnosť, zotrvačník si udržiava rovnaký smer, aj keď púzdro v ktorom je uložený budeme otáčať ľubovoľným smerom. Tento jav sa nazýva moment zotrvačnosti alebo aj gyroskopická zotrvačnosť. Uvedený jav sa neprejaví ak zotrvačník nebude rotovať veľkou rýchlosťou. Medzi zariadenia ako napríklad ambulantné systémy merajúce denné aktivity alebo využitie systémov typu FES⁵ na napravenie prepádavania nohy v členku často vyžadujú rozpoznanie prvotného kontaktu chodidla so zemou alebo momentu odlepenia sa chodidla od zeme počas dlhších prechádzok vonku. Používanie zariadení, ktoré sa spravidla nachádzajú v laboratóriách zaoberajúcich sa výzkumom chôdze a sú používané pre detekciu javov chôdze vo vnútri miestnosti nemusia predstavovať vhodnú voľbu pre úlohu merania chôdze vo vonkajšom prostredí. [10]

Gyroskopy umiestnené na spodnej časti nohy, boli navrhnuté na ambulantnú analýzu chodze, taktiež pre načasovanie a spätná väzba pre systémy typu FES ale napríklad aj pre systémy, ktoré sú využívané na kontrolu protéz a ortéz dolných končatín. Gyroskop sa preukázal ako prijateľne presný pri zdravej a chorobnej chôdzi kračajúc na rovnej zemi a v zdravej chôdzi kračajúc po stúpajúcom alebo klesajúcom povrchu. Taktiež boli gyroskopy navrhnuté aj pre klasifikáciu vzorov chôdze vrátane chôdze hore a dole schodami a bolo zistené, že je možné spoľahlivo určiť typ fyzickej aktivity, ktorý je vykonávaný jednotlivými subjektami. [10]

⁴V izolovanej sústave sa celkový moment hybnosti s časom nemení

⁵ Funkčná elektrická stimulácia (FES) je technika, ktorá používa nízkoenergetické elektrické impulzy na umelé vytváranie pohybu tela

V znázornenom grafe na obrázku 2.2 bol použitý gyroskop na detekciu prvotného kontaktu chodidla so zemou a chodidla od zeme. (a) a (d) zobrazujú signál gyroskopu subjektu chodiaceho po rovnej zemi. (b) a (e) zobrazujú signál pri stúpaní po schodoch, (c) a (f) pri zostupovaní po schodoch.



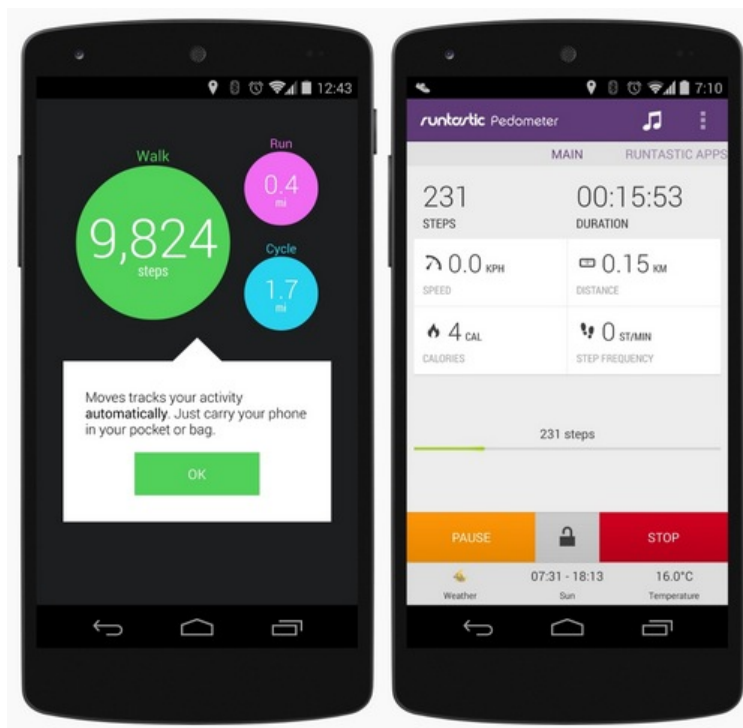
Obr. 2.2: Graf znázorňujúci signál gyroskopu. [10]

2.3 Krokový senzor

Krokový senzor alebo krokomer, je senzor v mobilnom zariadení určený primárne na detekciu krokov užívateľa, pričom aby zariadenie správne detekovalo kroky, je potrebné zariadenie s týmto senzorom nosiť so sebou počas detekovania. Jedná sa o senzor, ktorý bol založený na senzore akcelerometra, ale je presnejší a energeticky menej náročný pre danú činnosť. Senzor pozostáva z 2 senzorov, zo senzoru na detekciu krokov a z druhého senzoru ktoré umožňuje počítať kroky. Ak senzor na detekciu krokov detekuje krok, vráti túto informáciu ako hodnotu rovnú jednej. Rýchlosť získavania udalostí zo senzoru je pritom nastaviteľná. Ak užívateľ vykoná krok, tak krokový senzor aktivuje udalosť vždy keď je tento krok vykonaný. [15]

Aplikácie, ktoré využívajú krokový senzor sú väčšinou športového charakteru, určené pre športovcov alebo aj pre obyčajných užívateľov, ktorí si chcú prípadne monitorovať svoju dennú fyzickú aktivitu. Aplikácia Runtastic znázornená na obrázku 2.3 je určená pre presne pre týchto užívateľov a športovcov, ktorá pomocou krokového senzoru je schopná detekovať, kedy užívateľ vykoná krok a pomocou senzoru, ktorý počíta kroky zobrazuje užívateľovi ubehnutú vzdialenosť. Do aplikácie je užívateľ požiadaný o zadanie svojej celkovej hmot-

nosti, výšky, veku a pohlavia. Pomocou zadaných atributov sa počítajú napríklad aj spalené kalórie, ktoré sa odvíjajú od počtu krokov a ďalšie športové aktivity. [15]



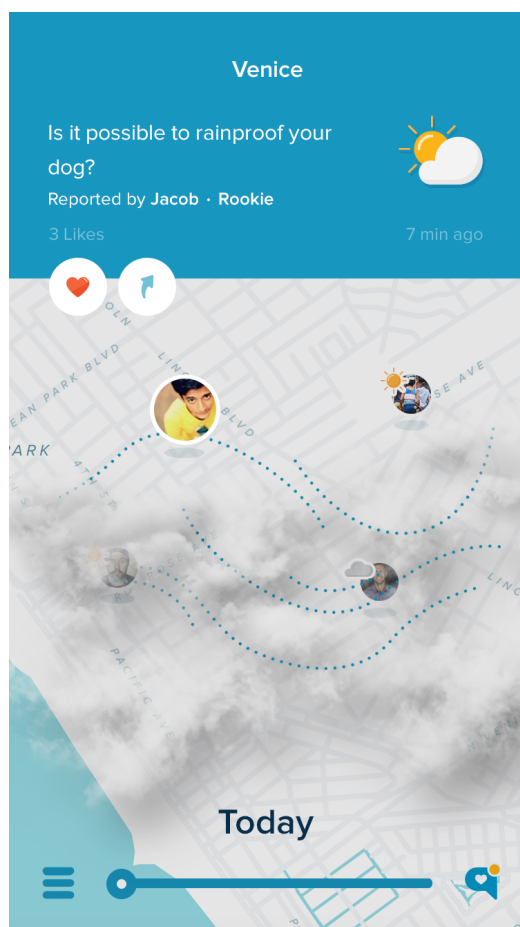
Obr. 2.3: Aplikácia Runtastic na mobilnom zariadení. [15]

2.4 Barometer

Barometer je senzor vďaka ktorému je mobilné zariadenie schopné určiť nadmorskú výšku, v ktorej sa práve nachádza. Barometer je pri tom citlivý senzor, ktorý je schopný rozpoznať rozdiel aj pri výstupe o 1 schod. Tento senzor vie rozpoznať, aké prevýšenie mobilné zariadenie prekonalo a to sa využíva hlavne v aplikáciách určených pre športovcov, ktoré vedú lepšie vyhodnotiť ako náročný tréning naozaj bol. Pomocou barometra vieme byť schopní získať informácie o atmosférickom tlaku tam kde sa užívateľ práve nachádza. Pridaním barometra do mobilných zariadení vznikla nová príležitosť pre predpoveď počasia. Pretože oficiálne predpovede počasia predpovedajú počasie 1 deň dopredu, niekedy sa nemusí predpoveď naplniť a preto by bolo vhodné využiť v mobilných zariadeniach dáta z barometrov. [13]

Aplikácia, ktorá využíva čítanie dát z barometrov mobilných zariadení sa nazýva PressureNet [3]. Pomocou aplikácie sú užívatelia schopní získať ich dáta z barometra a následne je im dovolené poskytnúť tieto dáta do meteorologickej siete. Následne boli dáta použité k presnejšej predikcii počasia, blížiaccej sa búrky alebo silnému vetru, kedy aplikácia upozorní užívateľov na blížiacie sa nebezpečenstvo. S celkovým počtom okolo 500 tisíc senzorov zapojených v sieti sa stala aplikácia PressureNet⁶ jednou z najväčších senzorových sietí pre predikciu počasia, kde zdrojom dát boli barometre užívateľov, ktorí sa rozhodli zdieľať svoje údaje a dopomôcť tým k spresneniu predikcií počasia.

⁶Sieť pre ohlasovanie údajov o barometrickom tlaku



Obr. 2.4: Mapa užívateľov zapojených do aplikácie pre zdieľanie dát z barometra. [3]

2.5 Svetelný senzor

Svetelný senzor sa stará o automatickú reguláciu jas displeja vzhľadom na úroveň svetla v miestnosti v ktorej sa práve mobilné zariadenie nachádza. Obsahuje ho väčšinou už každé moderné mobilné zariadenie, pretože výrazne prispieva k výrazne dlhšej výdrži mobilného zariadenia. Senzor určuje aktuálne osvetlenie v miestnosti, v ktorom sa pritom nachádzame. Súčiastka, ktorú svetelný senzor využíva sa nazýva fotorezistor, ktorá využíva fotoelektrický jav. Tento jav je pozorovaný, kedy pri dopade svetla na kov vyráža z atómov látky elektróny, ktoré sa potom v látke voľne pohybujú a zvyšujú jej vodivosť. Senzor sa hlavne v dnešnej dobe využíva kvôli dlhšej výdrži batérie na mobilných zariadeniach a taktiež na automatické zvýšenie jas displeja pre lepšiu čitateľnosť a orientáciu na zariadení. [5]

2.6 Senzor priblíženia

Senzor priblíženia, taktiež nazývaný ako proximity sensor, je umiestnený na prednej časti mobilného zariadenia nad displejom vedľa slúchadkového reproduktora. Senzor sa skladá z infradiody a infradetektora. Tento senzor sa využíva napríklad pri telefonovaní, kedy je telefon priložený k uchu, senzor toto detekuje a preto vypne dočasne dotykovú obrazovku



Obr. 2.5: Znázornenie polohy svetelného senzoru na mobilnom zariadení. [6]

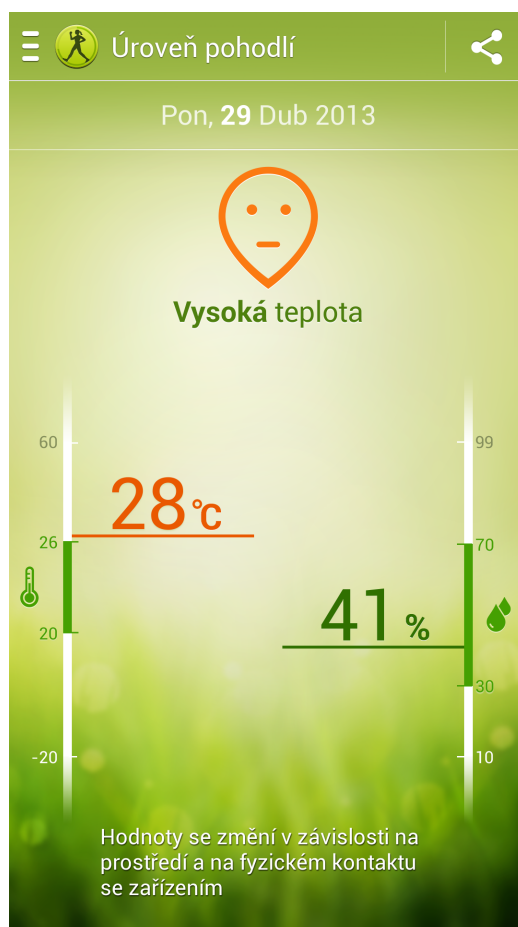
telefonu. Senzor detekuje IR⁷ paprsok, ktorý je pri položení telefónu k uchu odrazený od tváre späť do detektora. [13]

2.7 Tepelný senzor

Tepelný senzor alebo taktiež teplomer je senzor mobilného zariadenia, určený na meranie teploty okolitého prostredia v ktorom sa mobilné zariadenie práve nachádza. Obsahuje ho väčšinou už každé moderné mobilné zariadenie, pretože senzor taktiež monitoruje vnútorné hardwarové súčiastky spolu aj s batériou. Toto sa využíva hlavne aby sa predišlo poškodeniu mobilného zariadenia z dôsledku vysokej teploty, ktoré zariadenie rozpozná a preto aby sa predišlo poškodeniu, sa zariadenie automaticky vypne alebo vypne displej s hláškou informujúcou užívateľa, že je mobilné zariadenie potrebné schlaadiť na nižšiu teplotu aby ho bolo možné opätovne používať. Vysoká teplota mobilného zariadenia veľmi negatívne ovplyvňuje celkový výkon zariadenia. Mobilné zariadenie v dôsledku vysokej teploty znižuje výkon zariadenia, hľadá ďalšie možnosti ako dané zariadenie v danej chvíli čo najrýchlejšie schlaadiť. [13]

Tepelný senzor je pritom veľmi citlivý a preto by sme pri používaní mali dbať aby sme mobilné zariadenie umiestnili mimo iných zariadení, ktoré vyžarujú teplo, prípadne na iné miesta na ktoré nesvieti slnečné žiarenie a tým pádom by skreslili alebo vychýlili výsledok merania teploty daného mobilného zariadenia. Využitie tepelného senzora je možné nájsť v rôznych mobilných aplikáciach, ktoré monitorujú napríklad počasie alebo aj prostredie v ktorom sa práve s mobilným zariadením nachádzame. Jednou z takých aplikácií je práve aplikácia s názvom S Health, znázornená na obrázku 2.6, ktorá slúži na meranie teploty mobilného zariadenia, pričom užívateľovi poskytuje aj informácie ohľadom vlhkosti vzduchu a rôzne iné ďalšie vlastnosti okolitého prostredia. [13]

⁷Infračervené žiarenie



Obr. 2.6: Meranie teploty v aplikácii S Health. [13]

2.8 Senzorová fúzia

Senzorová fúzia patrí medzi jednu z techník, ktorá využíva dáta, informácie, ktoré mu poskytujú viaceré zdroje súčasne a využíva túto techniku, aby získala presnejšiu výslednú informáciu z viacerých senzorov naraz, ktoré nesú príznaky rôznych typov chýb ako napríklad šum. Výsledná informácia je teda viac presnejšia ako keď by boli použité zdroje z jednotlivých senzorov individuálne. Pri Senzorevej fúzii využívame dáta, ktoré sú poskytnuté napríklad zo senzorov poskytnuté mobilným zariadením. [9]

Senzorovú fúziu delíme na 3 základné typy:

- Komplementárna senzorová fúzia.
- Konkurenčná senzorová fúzia.
- Spolupracujúca senzorová fúzia.

V nasledujúcej časti práce čerpam informácie zo zdroja [9]. Senzorová fúzia nazývame typom komplementárna, ak senzor nie je závislý vzhľadom na ostatné senzory, pričom ale údaje z tohto senzora, sa môžu kombinovať s údajmi iných senzorov, aby ich kombináciou

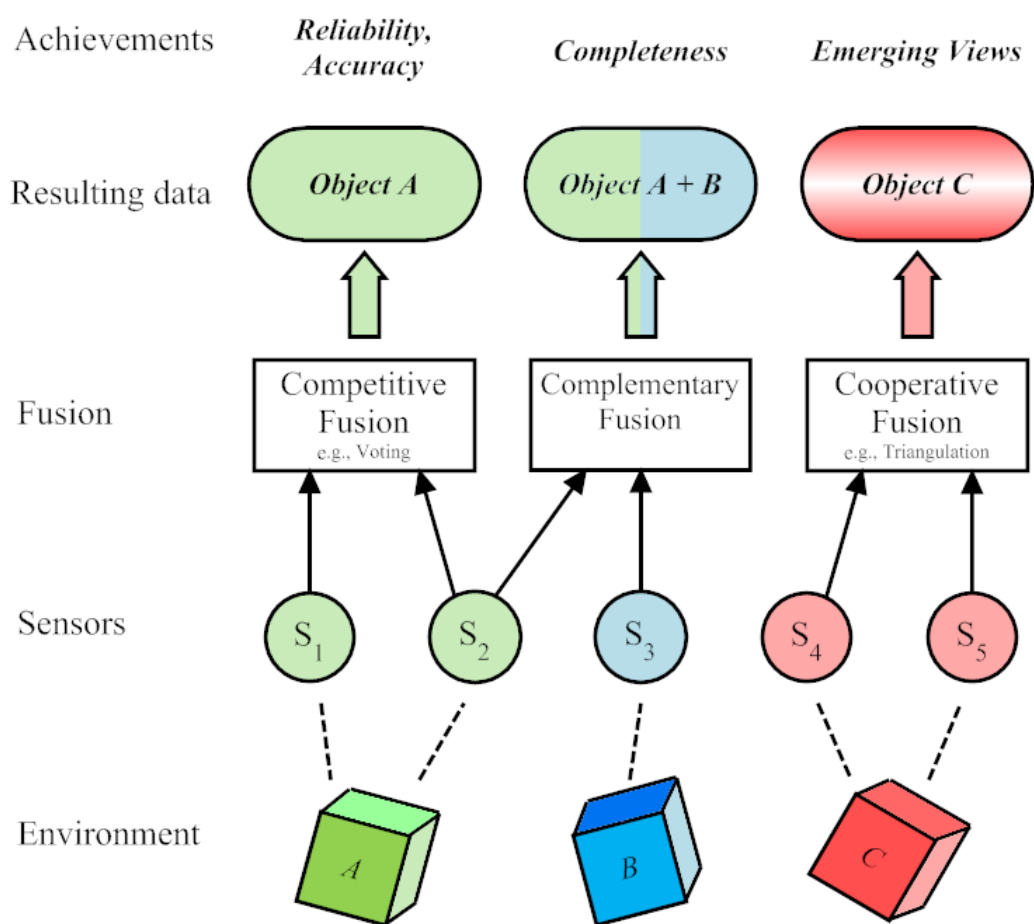
vznikol ucelený výstup. Tento typ sensorovej fúzie predchádza prípadným stratám údajov zo sensorov.

Konkurenčná sensorová fúzia nazývame vtedy, ak nám senzor poskytuje nezávislé údaje z merania. V špeciálnych prípadoch je táto metóda tolerantná k chybám, ale je potrebné, aby služba systému, ktorú má vykonávať bola presne špecifikovaná, taktiež je potrebná špecifikácia chybových stavov systému.

Pri Spolupracujúcej sensorová fúzií hovoríme o sensorovej sieti, ktorá sa skladá z 2 a viac sensorov, pri jednom senzore by táto sieť nebola dostupná. Využívajú sa údaje zo 2 a viac sensorov, ktoré snímajú ten istý objekt a spojením týchto údajov získame výsledok merania.

Predovšetkým sa využívajú tieto typy sensorovej fúzie v kombinácii s iným typom. Napríklad pri regióne, ktorý je snímaný dvomi a viac kamerami, je výhodné použiť typ spolupracujúcej alebo konkurenčnej sensorovej siete. Ak by bol región snímaný iba jednou kamerou, využil by sa typ komplementárnej sensorovej siete.

Jednotlivé typy sensorovej fúzie sú znázornené na obrázku 2.7, ktorý demonštruje fúziu na základe sensorov a objektov.



Obr. 2.7: Základné typy sensorovej fúzie. [9]

2.9 Chybovosť senzorov mobilných zariadení

Pri čítaní údajov zo senzorov mobilného zariadenia je veľmi dôležitá presnosť a správnosť týchto údajov. Už pri veľmi malej odchylke nepresnosti môže dôjsť za krátky čas k oveľa väčšej odchylke, a teda menej presným údajom. Medzi najčastejšie príčiny nepresnosti údajov patria hlavne:

- Šum.
- Odchýlka.
- Posunutie nulového bodu.
- Náhodná, ľudská chyba.
- Oneskorenie spôsobené časom.

Šum je náhodná odchylka signálu, ktorá sa mení v čase a má za následok nepresnosť údajov senzora. Väčšinou je spôsobená náhodnou chybou senzora. O posunutí nulového bodu hovoríme vtedy, ak nastane udalosť, kedy vystupný signál nie je rovný nule, ale meraná vlastnosť je nula. Pri odchylke dochádza k väčšej nepresnosti postupom času, ktorá vznikla z malej zmeny pri meraní. Chyba zapríčinená človekom môže vzniknúť pri čítaní dát a ich zlej interpretácii, ktorá je spôsobená nepozornosťou, prípadne zobrazenie dát v nepresnom formáte. Všetky odchylky ktoré boli spomenuté klasifikujeme medzi systematické a náhodné chyby. Niektoré typy chýb sú spôsobené nedostatočnou kalibráciou senzora a je možné opätovne vhodnou kalibráciou zmenšiť prípadne aj eliminovať riziko vzniku týchto chýb, pri ďalších typoch chýb, ktoré nie sú odstraniteľné pomocou kalibrácie je nutné použiť vhodné metódy filtrovania signálu [2], ale taktiež je možné použiť senzorovú fúziu. [20]

Kapitola 3

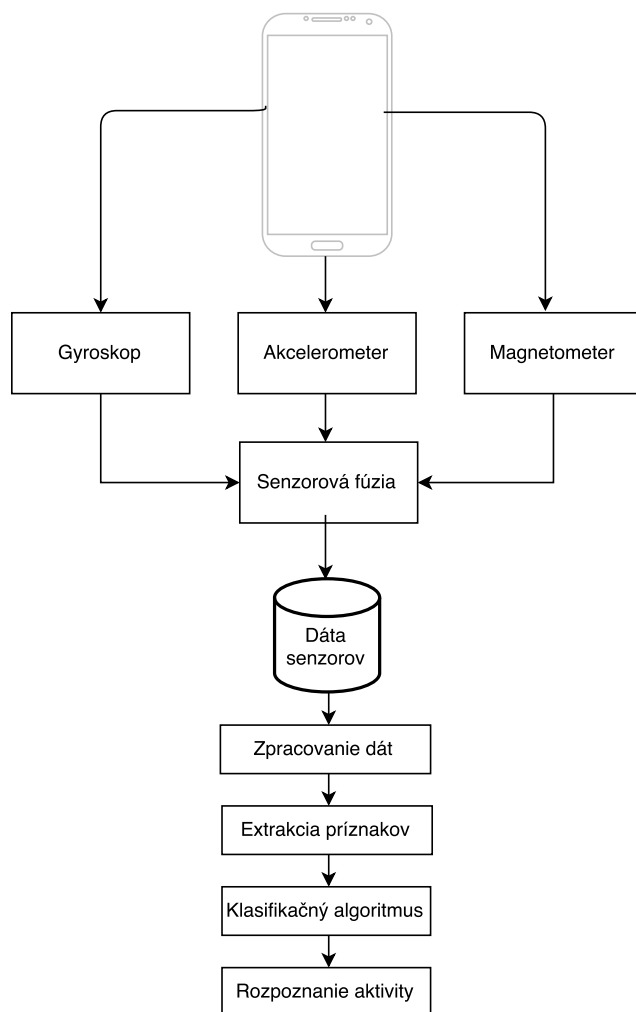
Návrh aplikácie

Pred samotnou implementáciou aplikácie, je nutné aplikáciu najprv navrhnuť, rozdeliť na menšie časti, podčasti, zamyslieť sa nad fungovaním samotnej aplikácie a vytvoriť graf zobrazujúci zjednodušený algoritmus na ktorom aplikácia funguje. Aplikácia bude pozostávať z použitia senzorov, ktoré su dostupné na mobilnom zariadení. Z jednotlivých senzorov vystupujú dáta, ktoré je nutné vhodným spôsobom spracovať, ošetriť od chybných meraní, výchylok. Pri aplikácii budeme využívať dáta senzorov mobilných zariadení pričom kombináciou dát z jednotlivých senzorov budeme môcť byť schopný rozpoznat zadanú aktivitu užívateľa, teda využijeme typ sensorovej fúzie nazývanú ako komplementárnú. Aby sme mohli byť schopný detekovať aktivitu užívateľa je potrebné si nazbierať, vytvoriť natrénovanú sadu dát zo senzorov, pri vykonávaní fyzických aktivít jednotlivými užívateľmi.

Hlavné senzory mobilného zariadenia, z ktorých budeme sledovať fyzickú aktivitu jednotlivých užívateľov sú akcelerometer, magnetometer v kombinácii s gyroskopom, taktiež je možné použiť krokový senzor. Následné zozbierané dáta je potrebné spracovať, odstrániť chybné výchyľky, zredukovať vzniknutý šum. Zozbieraným dátam na základe toho pri akej fyzickej aktivite boli zozbierané označíme labelom. Vyfiltrované dáta následne budeme segmentovať. Dáta je nutné následne uložiť do formátu typu ARFF¹. Po segmentácii použijeme techniku extrakcie príznakov a aby sme mohli predbežne spracované dáta analyzovať použijeme jeden z klasifikátorov pre klasifikáciu dát v nástroji WEKA. Zvolenému klasifikačnému algoritmu najprv dáme na vstup natrénovanú sadu dát uloženú vo formate ARFF. Klasifikačný algoritmus sa na základne trenovacích dát naučí rozpoznat jednotlivé odlišnosti v sadách trenovacích dát a je tým pádom schopný odlíšiť rôzne typy fyzickej aktivity. Natrénovaný klasifikačný algoritmus je potom už schopný predikovať, rozpoznat akú fyzickú aktivitu užívateľ daného mobilného zariadenia vykonáva.

Hlavný algoritmus zjednodušeného návrhu aplikácie je zobrazený na obrázku 3.1. Algoritmus sa skladá z niekoľkých základných častí: snímania dát zo senzorov mobilného zariadenia, pedspracovania týchto dát, extrakcie príznakov, tréovania zvoleného klasifikačného algoritmu a klasifikácie dát, ktoré sú klasifikačnému algoritmu vkladané, pričom si jednotlivé časti návrhu algoritmu popíšeme a navrhujeme viac podrobnejšie v nasledujúcich podsekciiach.

¹Súbor typu ARFF (Attribute-Relation File Format) je textový súbor ASCII, ktorý opisuje zoznam inštancií zdieľajúcich súbor atribútov



Obr. 3.1: Zjednodušený návrh algoritmu aplikácie.

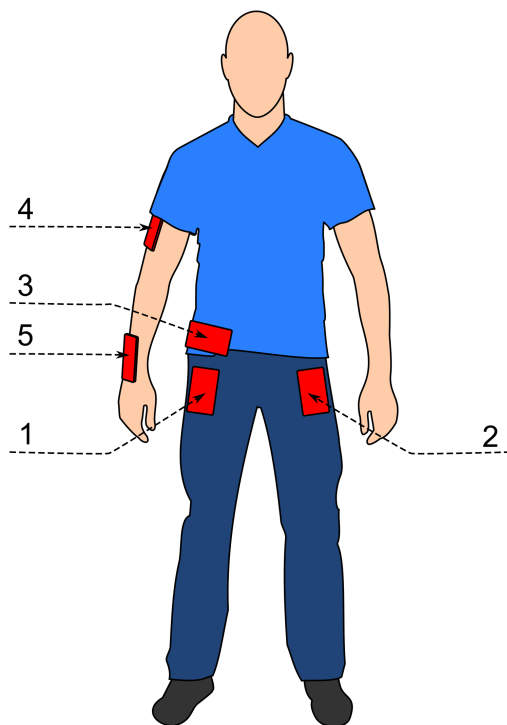
3.1 Analýza aplikácie

Dôležitou súčasťou návrhu aplikácie je stanovenie si, ktoré aktivity budeme požadovať aby systém rozoznal. Na rozpoznávanie rôznych aktivít budeme používať senzor akcelerometra, magnetometra a gyroskopu. Ak by sme použili na rozoznanie aktivít iba dáta z gyroskopu, nebudeme môcť rozlíšiť aktivitu pri sedení a statí. Na druhej strane ak by sme použili iba akcelerometer nebudeme môcť rozlíšiť alebo iba nepresne by sme rozlíšili aktivitu pri chodení po schodoch hore a dole. Preto je nutné použiť kombináciu dát z gyroskopu, akcelerometra a magnetometra. Kombináciu týchto dát je možné dosiahnuť komplementárnou senzorovou fúziou týchto dát. Medzi základne aktivity, ktoré budu rozoznatelné systémom patria:

- Státie, chodenie.
- Behanie, skákanie.

Taktiež je dôležité si ešte pred zbieraním dát zo senzorov, určiť pozíciu mobilného zariadenia na tele osoby, na ktorej budú dáta zo senzorov pri určitej aktivite nahrávané. Pozícia mobilného zariadenia významne vplýva na rozpoznatelnosť aktivity užívateľa. V prípade

použitia dát z viacerych pozícií mobilného zariadenia sa taktiež zväčší aj presnosť rozpoznávacieho systému, pretože klasifikačný algoritmus bude natrénovaný na väčšej pestrosti dát. V prípade aplikácie bolo preto navrhnuté použitie dát z viac ako jednej pozície na tele užívateľa a to napríklad nohy a ruky. Rôzne pozície mobilného zariadenia sú znázornené na obrázku 3.2.



Obr. 3.2: Rôzne pozície mobilného zariadenia na tele užívateľa [18]

Vhodné je stanoviť si aj počet používateľov, z ktorých budú dáta zbierané. Každá osoba ma trochu odlišný typ chôdze a preto je v prípade lepšej presnosti systému dôležité dbať aj na počet užívateľov z ktorých budú dáta získavané. Preto bude na začiatku stanovený presný počet aktivít a počet osôb, ktoré tieto aktivity vykonávajú a získané dáta sa neskôr použijú na spracovanie. Čas za ktorý budú opakovane vykonávať osoby stanovenú aktivitu bude stanovený na 5 minút.

3.1.1 Ciele aplikácie

Cieľom navrhnutej aplikácie je implementovať komplementárnu senzorovú fúziu, tak aby sa senzory mobilného zariadenia typu akcelerometer, magnetometer, gyroskop, navzájom dopĺňali a teda, aby keď sme získavali dáta zo senzorov, boli tieto dáta ošetrené, vyfiltrované. Vysledná aplikácia musí byť schopná natrénovať klasifikátor tak, aby bol schopný rozpoznať fyzickú aktivitu užívateľa. Užívateľovi bude na začiatku aplikácie ponuknutý výber z jednotlivých aktivít. Po zakliknutí jednej z daných aktivít a kliknutí na tlačítko zbierať dáta, bude

užívateľovi umožnené vytvoriť si vlastný súbor so sadou tréningových dát. Aplikácia bude schopná identifikovať jednotlivé fyzické aktivity, ktoré užívateľ vykonal pomocou už predom natrénovaného klasifikátora. Užívateľské prostredie výslednej aplikácie je znázornené na obrázku 3.3.



Obr. 3.3: Užívateľské prostredie aplikácie.

3.2 Zpracovanie dát

Dáta, ktoré boli zozbierané pri užívateľoch vykonávajúcich jednotlivé fyzické aktivity pomocou senzorov mobilného zariadenia je potrebné ošetriť a zlepšiť tak presnosť pri rozpoznávaní aktivity. Ošetrovanie dát je možné pomocou komplementárnej sensorovej fúzie a filtrov typu dolná a horná priepusť.

3.2.1 Odstránenie šumu

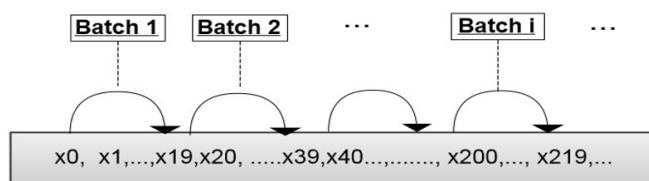
Ešte predtým než budeme schopný analyzovať rôzne aktivity, je dôležité pripraviť na to dáta. Jednotlivé senzory nie sú perfektné a je preto potrebné dáta, ktoré senzory generujú ošetriť od výchylok, chýb a šumu ktoré môžu nastať vplyvom okolitého prostredia alebo sa aspoň pokúsiť dané výchyly, čo najviac minimalizovať. Aby sme minimalizovali šum a výchyly senzorov bol v práci použitý komplementárny filter [12]. Komplementárny filter sa

skladá z filtrov dolnej priepusti a taktiež hornej priepusti. Tento filter je taktiež použitý na implementáciu sensorovej fúzie. Medzi techniky, ktoré sa pritom používajú patrí aj odstránenie šumu. Surové dáta, ktoré sme získali zo sensorov pri nahrávaní určitých aktivít môžu obsahovať aj šum, ktorý znamená že signál môže obsahovať nechcené zmeny, nerovnosti, ktoré budú skresľovať prípadné výpočty na dátach. Preto je potrebné signál odstrániť od šumu. Aby sme boli schopný namerať skutočnú akceleráciu zariadenia, je potrebné predovšetkým silu gravitácie odstrániť [2]. Používa sa k tomu vyhladzovacie metódy, filter typu horná a dolná priepusť. Takto vyfiltrované data sú ďalej zpracované. Aby sme teda silu gravitácie odstránili, bol použitý filter s hornou priepustou. Pomocou filtra s dolnou priepustou sme schopný taktiež silu gravitácie izolovať 4.2. Pomocou týchto filtrov sme schopný dáta akcelerometra filtrovať aby sme odstránili silu gravitácie a zredukovali šum v dátach akcelerometra. Akcelerometer poskytuje gravitačný vektor² zatiaľ čo magnetometer funguje ako kompas.

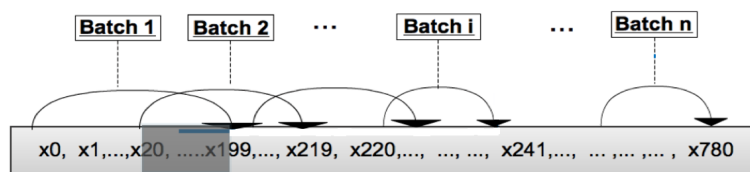
3.2.2 Segmentácia dát

Ďalší dôležitý krok pri zpracovávaní dát je ich segmentácia, rozdelenie céleho toku dát na menšie časti, do menších segmentov pomocou prístupu posuvného okna (Sliding window approach). Dôležité je určiť veľkosť posuvného okna, ktorá určuje dostatočnú rýchlosť a presnosť pri rozpoznávaní aktivít. Segmentáciu rozdeľujeme do dvoch rôznych kategórií. [21]

- Segmentácia bez prekrývania dát.
- Segmentácia s prekrývaním dát.



Obr. 3.4: Segmentácia bez prekrývania dát. [21]

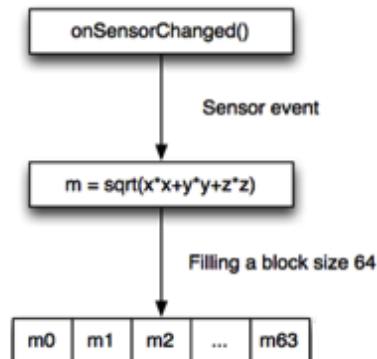


Obr. 3.5: Segmentácia s prekrývaním dát, šeda farba - prekryté dáta. [21]

Vo väčšine systémov na rozpoznanie aktivity je použitá segmentácia bez prekrývania dát, s danou hodnotou veľkosti posuvného okna. Tento prístup je vhodný, pretože redukuje vypočetnú náročnosť a je vhodné ho použiť pri opätovnom získavaní dát po určitom čase.

²Vektor smerujúci smerom k centru zeme

V bakalárskej práci budeme segmentovať po sériach 64 magnítúd do buffru, pričom sa to nazýva vektor vlastností (Feature vector). Použitá segmentácia v práci bola bez prekryvania dát, pretože je efektívnejšia. [21]



Obr. 3.6: Feature vector. [8]

3.2.3 Súbor ARFF

Dôležitou súčasťou práce je aj vytvorenie trénovacej sady dát, ktorú uložíme do súboru s názvom `features.arff`. Obsah súboru musí byť pripravený v súlade s formátom ARFF. Súbor typu ARFF (Attribute-Relation File Format) je textový súbor ASCII³, ktorý opisuje zoznam inštancií zdieľajúcich súbor atribútov. Tento typ súboru bol vyvinutý projektom strojového učenia na katedre informatiky Univerzity Waikato pre použitie s nástrojom WEKA. Súbor sa skladá z 2 sekcií, prvá časť súboru obsahuje informácie o hlavičke a list atribútov aj s typom daného atribútu znázornené na zozname C.1. Druhá časť súboru typu ARFF obsahuje jednotlivé dáta. Deklarácia `@data`, slúži na označenie začiatku dátového segmentu v súbore. [4]

Formát atribútu je zadefinované ako:

@attribute <meno-atribútu> <datatyp>

Deklarovanie atributov dát má nasledujúcu podobu:

@attribute <meno> dátum [<formát-dátumu>]

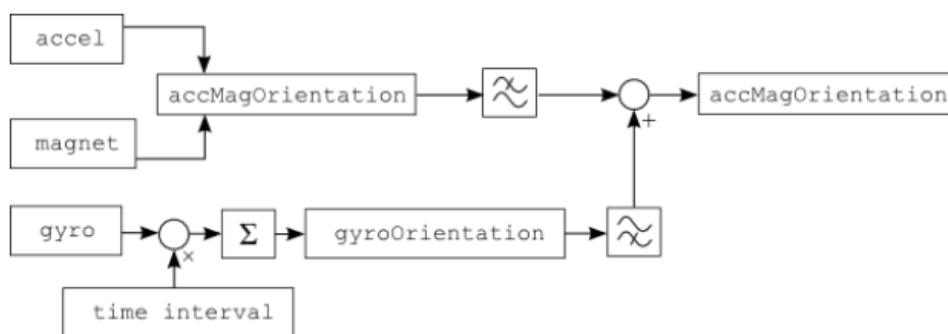
Príklad obsahu súboru formátu ARFF je zobrazený v prílohe C.

3.3 Senzorová fúzia

Návrh komplementárnej sensorovej fúzie, ktorý bol použitý pri jej implementácii v bakalárskej práci je znázornená na obrázku 3.7. [12]

Návrh sa skladá z použitia 3 senzorov, akcelerometru, magnetometru a gyroskopu. Pre implementáciu sensorovej fúzie v práci bola použitá časť z frameworku `AndroidSensorFusion` od autora Jose Collas.

³Americký štandardný kód pre výmenu informácií



Obr. 3.7: Návrh komplementárnej sensorovej fúzie. [12]

V nasledujúcej sekcií čerpam informácie z [12]. Z informácií, ktoré poskytujú senzory akcelerometra a magnetometra sme schopný získať orientáciu mobilného zariadenia `SensorManager.getOrientation()`, ale senzory nie sú presné a to predovšetkým výstup zo senzora magnetometra.

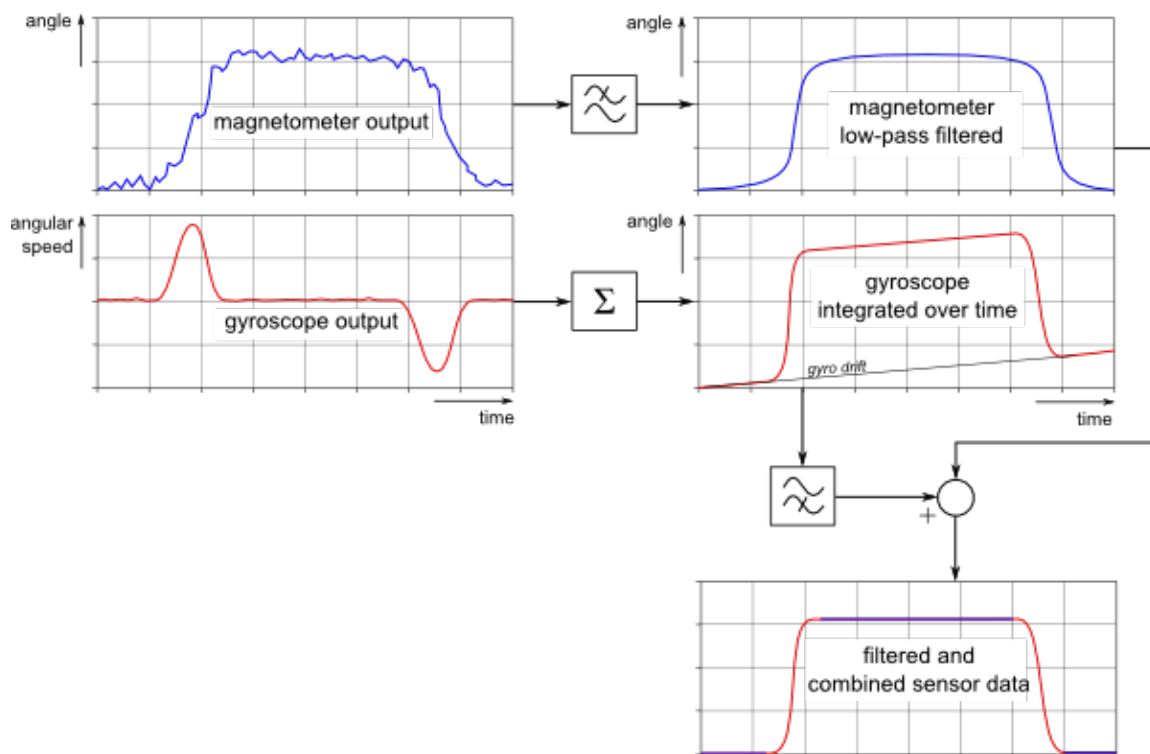
Zatiaľ čo senzor typu gyroskop je v mobilnom zariadení oveľa presnejší ako akcelerometer. Najväčšou nevýhodou gyroskopu je gyro drift. Gyroskop poskytuje uhlové rýchlosti na všetkých troch osiach. Aby sme získali aktuálnu orientáciu tieto hodnoty uhlových rýchlostí je potrebné integrovať v priebehu času a to vynásobením hodnôt uhlových rýchlostí s časovým intervalom medzi posledným a súčasným výstupom senzora. To prináša prírastok rotácie. Súčtom nárastov rotácie získame absolútnu orientáciu zariadenia. Počas týchto výpočtov sa môžu vyskytnúť malé chyby, ktoré sa postupom času nazbierajú a spôsobia konštantnú pomalú rotáciu vypočítanej orientácie. Tento výpočet bol použitý v práci a je taktiež prístupný pre vývojárov v dokumentácii pre Android. Táto chyba sa taktiež nazýva gyro drift. Štandardný gyroskop poskytuje dáta bez žiadnych dodatočných korekcií pre šum a drift. Preto je nutné určiť tento drift a šum a to napríklad monitorovaním iných senzorov ako akcelerometer alebo magnetometer.

Implementáciou komplementárnej sensorovej fúzie sme schopný predísť malým chybám v senzoroach a dosiahnuť tým presnejšie dáta v práci a to skombinovaním viacerých senzorov aby sme predišli chybám, ktoré samotné senzory nevedia, nie sú schopné odstrániť. Dáta z gyroskopu sú aplikované na zmenu orientácie v krátkych intervaloch, zatiaľ čo dáta zo senzorov z akcelerometru a gyroskopu sú použité počas dlhších časových úsekov. Ďalej sa sensorová fúzia skladá z funkcií, `accMagOrientation()` a `gyroOrientation()`.

Filtrovanie akcelerometru a gyroskopu filtrom typu dolná priepusť (`accMagOrientation`) sú orientačné uhly spriemerované v priebehu času v konštantnom časovom okne. Neskôr v implementácii sa to dosiahlo pomalým zavedením nových hodnôt zo senzorov z akcelerometra / magnetometra do absolútnej orientácie.

Na obrázku 3.8 sú znázornené grafy dát z jednotlivých senzorov a taktiež grafy dát zo senzorov po vyfiltrovaní filtrom typu dolná priepusť alebo taktiež graf dát gyroskopu po integrácii v priebehu času. Posledný graf zobrazuje skombinované, už vyfiltrované dáta z jednotlivých senzorov mobilného zariadenia. Obrázok 3.8 slúži pre lepšiu predstavu o

tom, ako jednotlivé funkcie, ktoré komplementárna senzorová fúzia obsahuje, ovplyvňujú jednotlivé dáta zo senzorov.



Obr. 3.8: Znázornenie dát senzorov počas jednotlivých operácií sensorovej fúzie. [12]

3.4 Extrakcia príznakov

Pre rozpoznanie aktivity, budeme extrahovať príznaky z frekvenčnej a časovej oblasti. Časová oblasť obsahuje základné štatistické údaje o každom segmente dát. [16]

- Mean (stredná hodnota).
- Max, Min (maximum, minimum).
- Standard deviation, Variance (štandardná odchylka).
- Correlation (korelácia).
- Signal Magnitude (signál magnitúdy).

Príznaky z frekvenčnej oblasti popisujú periodicitu signálu.

- Energy (energia).
- Entropy (entropia).
- Time between peaks (čas medzi vrcholmi).
- Binned distribution (vedierková distribúcia).

Štatistické údaje o segmente dát a príznaky z frekvenčnej oblasti boli prevzaté z [16]. Pre každé posuvné okno je potrebné vyextrahovať rôzne príznaky z uvedených oblastí. Hlavným dôvodom výberu príznakov bolo kvôli potvrdeniu si výsledkov pomocou rôznych typov príznakov pre dôveryhodnejšie výsledky. Je vhodné extrahovať príznaky typu prechod cez nulu a stredná kvadratická hodnota, ktoré sa používajú pre rozoznávanie gest. Taktiež su menej citlivé na zmenu polohy v prípade akcelerometra a gyroskopu.

3.5 Nástroj WEKA

Pri analýze aplikácie bol zvolený nástroj, ktorý obsahuje klasifikačné algoritmy, tak aby sme ich boli schopný v práci použiť. Nástroj WEKA⁴, ktorý poskytuje rôzne algoritmy a sadu nástrojov pre prediktívne modelovanie a analýzu dát napísané v programovacom jazyku Java, vyvinutý na University of Waikato. Jedná sa o slobodný software dostupný podľa licencie GNU General Public License⁵. Nástroj WEKA taktiež podporuje predzpracovanie dát, clustering, klasifikáciu, regresiu, vizualizáciu a extrahovanie príznakov z dát. Z nástroja WEKA bol použitý v práci klasifikačný algoritmus, ktorý sa nazýva J4.8, pričom sa jedná o WEKA implementáciu algoritmu C4.5. Tento algoritmus je podrobnejšie popísaný v sekcii 3.6 pre klasifikačný algoritmus. [22]



Obr. 3.9: Logo softwaru WEKA vyvinutý University of Waikato. [22]

Medzi hlavné výhody použitia nástroja WEKA patria:

- Obsahuje veľa rôznych algoritmov pre vyťažovanie dát a strojové učenie.
- Je voľne dostupný pre užívateľov.
- Nezávislý na platforme.
- Udržiava sa v aktuálnom stave, pričom sa pridávajú nové algoritmy.

⁴Waikato Environment for Knowledge Analysis (Weka)

⁵Licencia pre slobodný software

3.6 Klasifikačný algoritmus

Pomocou vhodne vybraného klasifikačného algoritmu sme schopný analyzovať a rozoznať fyzickú aktivitu užívateľa. Klasifikačný algoritmus alebo taktiež klasifikátor, implementuje matematickú funkciu, ktorá zobrazuje vstupné dáta do tried. Základnou úlohou klasifikátora je zaradiť do ktorej kategórie dát, dané pozorovanie patrí, pričom je k dispozícii tréningová množina obsahujúca pozorovanie, pre ktorú sú kategórie správne určené. Pre rôzne druhy klasifikácie používame rôzne druhy klasifikačných algoritmov. [16]

- Rozhodovacie stromy.
- Algoritmus k-NN.
- Rozhodovacie tabulky.
- Hidden Markov Model .
- Neuronové siete.
- Naive Bayes.

Postupne sa vhodne zvolený klasifikátor najprv naučí na základe vytvorených tréningových dát, aké majú dáta informáciu, jednotlivé atribúty, teda naučí sa odlíšiť dáta od iných dát. Až po naučení je vhodné klasifikátor použiť pre klasifikáciu ďalších dát, ktoré mu budú vkladané ako vstup. Tréningovanie je možné uskutočniť dvoma spôsobmi a to buď online na mobilnom zariadení alebo offline na počítači. Ide o prípravný krok, ktorým sme schopný získať parametre modelu, ktoré využijeme neskôr pri klasifikácii. Klasifikáciu dát sme taktiež schopný vykonať offline pomocou nástroja pre strojové učenie WEKA alebo online na mobilnom zariadení.

Klasifikačný algoritmus, ktorý bol zvolený v bakalárskej práci sa nazýva J4.8, pričom sa jedná o Java⁶ implementáciu algoritmu C4.5 v nástroji WEKA, ktorý radíme do kategórie klasifikátorov rozhodovacie stromy⁷. Jedná sa o klasifikačný algoritmus, schopný generovať rozhodovací strom, kde každý uzol rozdeľuje triedy na základe získavania informácií. Atribút s najvyšším normalizovaným ziskom informácií sa používa ako kritérium rozdelenia. C4.5 algoritmus je nástupcom ID3⁸, v ktorom koreň a rodič sú vybrané nielen na základe zisku informácií, ale aj na pomere zisku ako výberu rodičov nájdením rozdeľujúcej informácie. Následne takto vygenerovaný rozhodovací strom môžeme použiť pre klasifikáciu dát. Aby bol algoritmus schopný vygenerovať tento rozhodovací strom, je potrebné mu ale dodať sadu tréningových dát, ktorá sa skladá z už klasifikovaných vzoriek dát, ktoré majú p-dimenzionálne vektory definujúce atribúty vzorky. Pre natréningovanie klasifikačného algoritmu na sade dát, je potrebné vytvoriť súbor typu ARFF, ktorý je podrobnejšie popísaný v sekcii 3.2.3. [22]

Algoritmus J4.8 bol vybraný prevažne kvôli vysokej presnosti v rozhodovaní. V práci používame nástroj WEKA, ktorý obsahuje implementáciu algoritmu J4.8 a sme ho tým pádom schopný použiť pomocou tohto voľne dostupného nástroja.

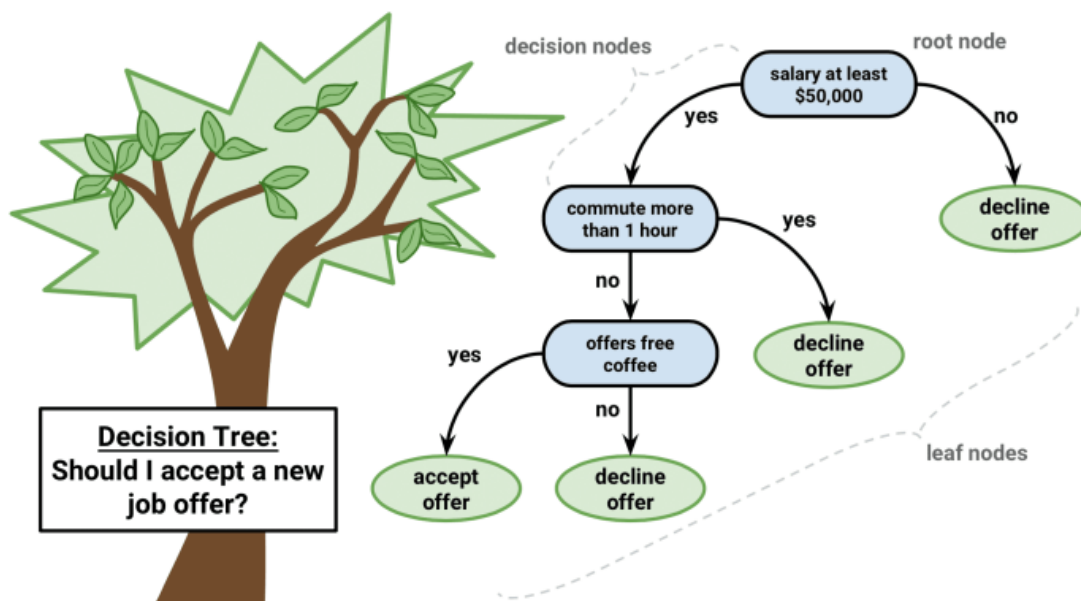
⁶Objektovo orientovaný programovací jazyk

⁷Klasifikátor so stromovou štruktúrou

⁸Iterative Dichotomiser 3 je algoritmus schopný generovať rozhodovací strom z datasetu

3.6.1 Rozhodovacie stromy

Rozhodovacie stromy patria medzi algoritmy, ktoré je vhodné použiť pri klasifikácii a predikcii alebo aj dolovaní dát. Rozhodovací strom reprezentuje pravidlá, ktoré sú jednoduché na vyjadrenie prirodzeným jazykom narozdiel od umelých neuronových sietí⁹. Hlavný motív používania rozhodovacích stromov je vytvorenie modelu výcviku, ktorý možno použiť na predpovedanie triedy alebo taktiež aj hodnoty cieľových premenných pomocou učenia sa pravidiel rozhodovania odvodených z predchádzajúcich dát (trénovacej sady dát). Algoritmus rozhodovacieho stromu sa pokúša vyriešiť problém pomocou reprezentácie stromu. Každý vnútorný uzol rozhodovacieho stromu zodpovedá atribútu a každý uzol listu zodpovedá zodpovedá označeniu triedy. Príklad štruktúry rozhodovacieho stromu je zobrazený na obrázku 3.10.

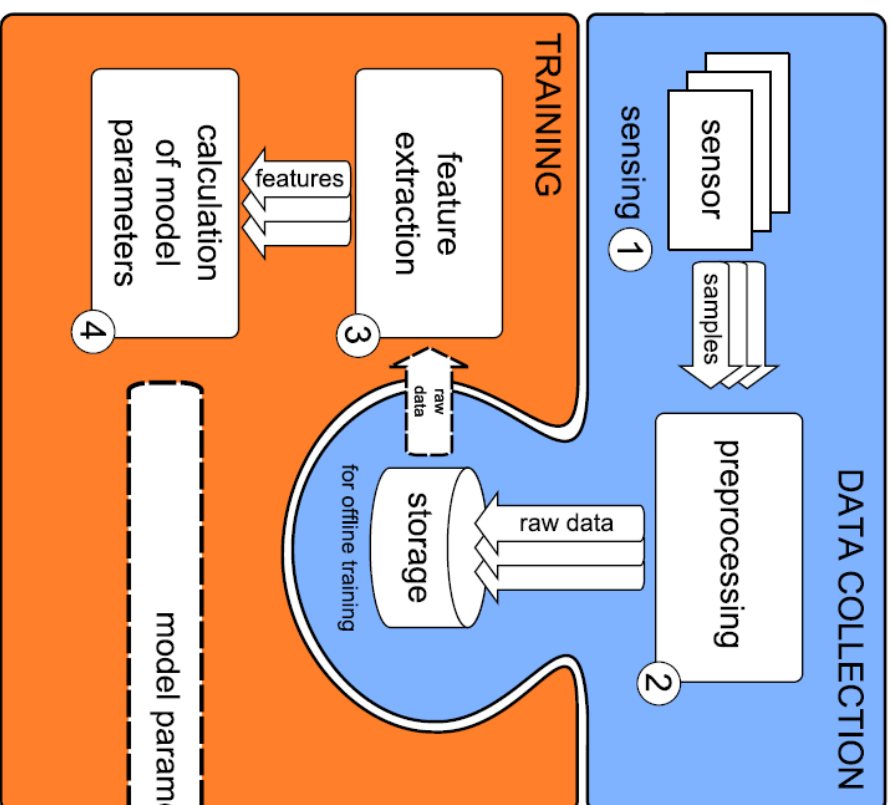


Obr. 3.10: Stromová štruktúra rozhodovacieho stromu. [17]

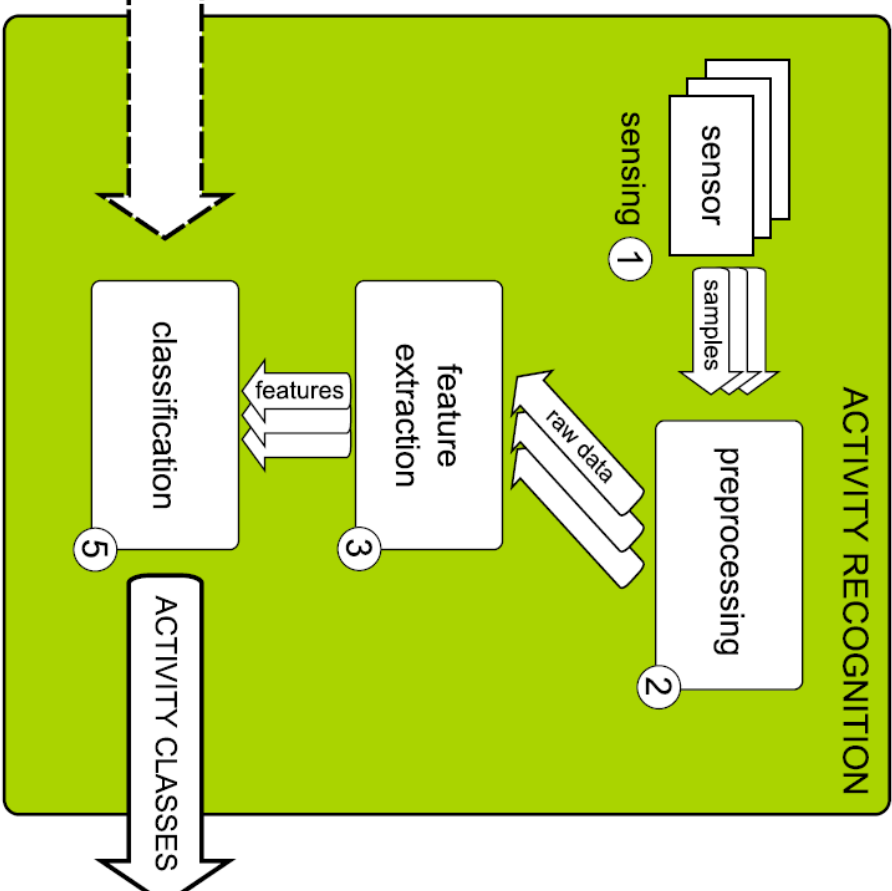
Na obrázku 3.11 je zobrazený algoritmus strojového učenia tréningu a klasifikácie dát spolu aj s ich zpracovaním.

⁹Vypočetný model používaný v umelej inteligencii

PREPARATION



USE



Obř. 3.11: Trénovanie a klasifikácia dát. [19]

Kapitola 4

Implementácia

Práca bola implementovaná v objektovo-orientovanom programovacom jazyku Java. Vývojové prostredie, v ktorom bola práca vyvíjaná a priebežne testovaná sa nazýva Android Studio od firmy Google. Mobilné zariadenie, z ktorého som čerpal dáta zo senzorov pre testovanie a vývoj bolo zvolené Oneplus X, s mobilným operačným systémom Android s verziou Nougat 7.0. Názov aplikácie a projektu bol zvolený sensorfusion, takto sa aplikácia taktiež bude nazývať po nainštalovaní do mobilného zariadenia. Súbor v ktorej bola práca implementovaná má názov MainActivity.java. V tejto kapitole budú rozpísané z ktorých tried, súborov a funkcií ktoré majú dôležitú úlohu sa práca skladá a budú podrobnejšie opísané akú funkcionality jednotlivé triedy majú, taktiež budú spomenuté komplikácie a problémy, ktoré sa počas implementovania bakalárskej práce vyskytli.

Pri implementovaní aplikácie bol použitý framework¹, ktorý poskytuje Android a umožňuje vytvárať nové aplikácie a hry pre mobilné zariadenia v jazykovom prostredí Java. Android framework poskytuje množstvo rôznych funkcií, komponentov a nástrojov, ktoré uľahčujú prácu a ponúkajú prístup k senzorom v mobilných zariadeniach. Platforma Android podporuje senzory, ktoré rozdelíme do 3 kategórií:

- Pohybové senzory.
- Environmentálne senzory.
- Polohové senzory.

Senzor	Typ senzoru	Jednotka	Použitie
TYPE_ACCELEROMETER	Hardware	m/s^2	Detekcia pohybu (chvenie, náklon, ...)
TYPE_GYROSCOPE	Hardware	rad/s	Detekcia rotácie (otočenie, vývrtka, ...)
TYPE_MAGNETIC_FIELD	Hardware	mT	Kompas

Tabuľka 4.1: Popis jednotlivých senzorov

¹Softwarová štruktúra, ktorá slúži ako podpora pri programovaní a vývoji softwaru

4.1 Prístup k senzorum

Pre prístup a získavanie dát zo senzorov mobilného zariadenia bol použitý Android Sensor framework. Tento framework poskytuje niekoľko tried a rozhraní, ktoré uľahčia vykonávať najrôznejšie úlohy pre senzorové účely. Framework taktiež poskytuje prístup k rôznym typom senzorom, ktoré sa nachádzajú na mobilnom zariadení. Táto práca sa zaoberá senzormi typu: akcelerometer, gyroskop a magnetometer. Jednotlivé charakteristiky senzorov sú znázornené v tabuľke 5.2. Systém súradníc je znázornený na obrázku 5.2.

- Senzor typu akcelerometer meria zrýchlenie sily, ktorá je aplikovaná na zariadenie vo všetkých 3 osiach sústavy súradníc (x, y, z).
- Senzor typu gyroskop meria gravitačnú silu, ktorá je aplikovaná na zariadenie vo všetkých 3 osiach sústavy súradníc (x, y, z).
- Senzor typu magnetometer meria okolité geomagnetické pole vo všetkých 3 osiach sústavy súradníc (x, y, z).

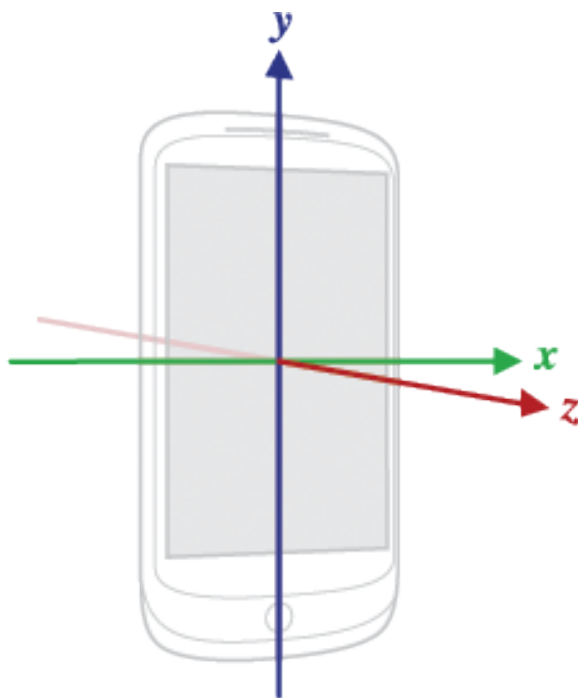
Aby sme boli schopný zbierať dáta z týchto senzorov, je potrebné použiť sensor framework, ktorý je súčasťou balíčku `android.hardware` a obsahuje triedy a rozhrania. Aby sme vytvorili inštanciu služby senzoru bola použitá trieda `SensorManager`, ktorá taktiež obsahuje metódy k prístupu, odposluchu, registrovaniu a odregistrovaniu snímaču udalostí rôznych senzorov v mobilnom zariadení. Pomocou triedy `Sensor` bola vytvorená inštancia špecifického senzoru, v našom prípade akcelerometra, gyroskopu a magnetometra. Trieda `SensorEvent` je použitá k vytvoreniu objektu `SensorEvent`, ktorý obsahuje informácie o udalosti senzoru. Pomocou rozhrania `SensorEventListener` sme boli schopný monitorovať dáta zo senzorov, implementovaním metód `onAccuracyChanged()` a `onSensorChanged()`. Obidve metódy sa nazývajú callback metódy, tieto metódy sú automaticky volané systémom android ak senzor ohlásí novú hodnotu, bude volaná metóda s názvom `onSensorChanged()`. Podobne je zavolaná metóda `onAccuracyChanged()`, vždy ak sa presnosť senzora zmení. V práci bola použitá metóda `onSensorChanged()` pomocou ktorej, sme boli schopný získať nové dáta od senzorov z objektu `SensorEvent`, ktorý obsahuje informácie o nových dátach, presnosť týchto dát, informáciu z ktorého senzoru tieto dáta pochádzajú a časové razítko, kedy boli dané dáta vygenerované.

Akcelerometer určuje akceleráciu, ktorá je aplikovaná na dané zariadenie meraním síl, ktoré sa aplikujú na daný senzor a to pomocou nasledovného vzťahu. Nasledovné vzorce boli prebraté z [2]. Zrýchlenie, ktoré sa aplikuje na zariadenie označujeme A_d . Sila, ktorá je aplikovaná na samotné zariadenie označujeme F_s . Silu gravitácie, označujeme g , pričom, ak je zariadenie položené na stole, akcelerometer číta veľkosť $g = 9,81m/s^2$. $mass$ označuje hmotnosť.

$$A_d = - \sum F_s / mass \quad (4.1)$$

Nameraná akcelerácia obsahuje taktiež ale silu gravitácie g .

$$A_d = -g - \sum F / mass \quad (4.2)$$



Obr. 4.1: Znáznorenie x, y, z súradníc vzhľadom na mobilné zariadenie. [1]

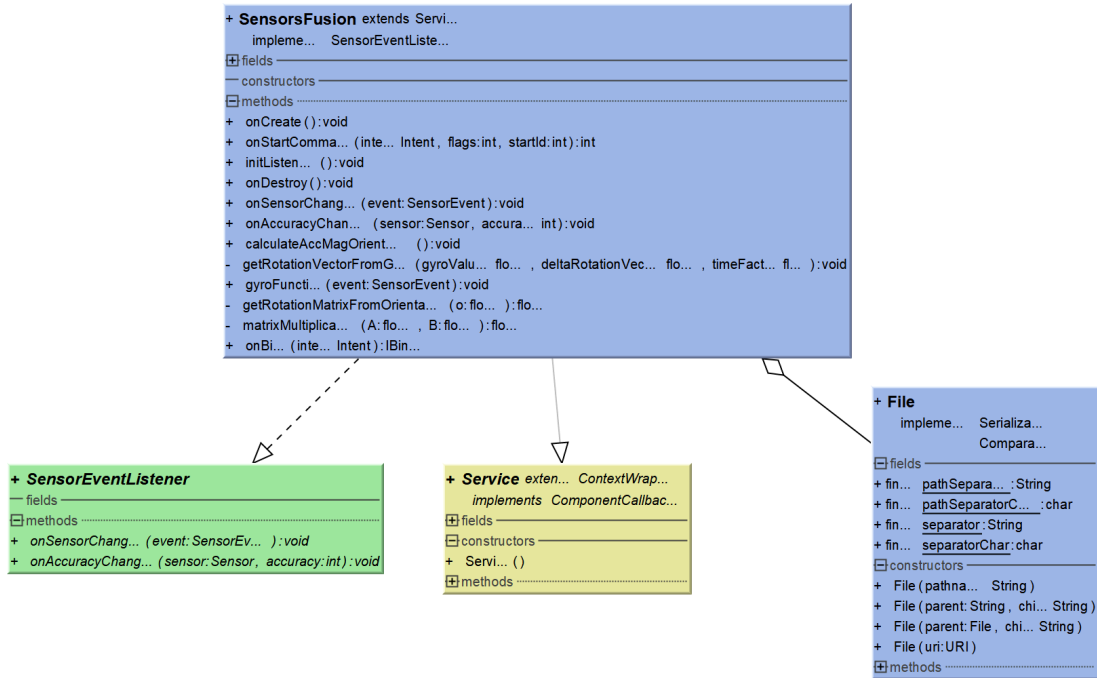
Funkcia `onCreate(Bundle)` je metóda, ktorá implementuje podtriedy aktivity. Je to metóda, v ktorej inicializujeme svoju aktivitu, činnosť. Na tomto mieste je volaná funkcia `setContentView(int)` pomocou ktorej definujeme používateľské rozhranie, ktoré bolo navrhnuté v návrhu. Následne vytvárame inštanciu `SensorManager` triedy, volaním metódy `getSystemService()`. Ďalej získame senzory, ktoré budeme ďalej v práci používať a to pomocou metódy `getDefaultSensor()`, ktorá prijíma konštantu pre špecifický senzor, ktorý chcem použiť. Prvotnou inicializáciou senzorov sme schopný ich ďalej použiť v kóde. [14]

Funkciou `onSensorChanged(SensorEvent event)` následne získaváme dáta zo senzorov. Aby sme mohli dáta získať, teda aby boli doručené hneď ako je to možné, je potrebné zaregistrovať `SensorEventListener` pre daný senzor pri danej vzorkovacej frekvencii, ktorá bola zvolená ako `SENSOR_DELAY_FASTEST`. Pre daný senzor registrujeme `SensorEventListener` pomocou funkcie `registerListener (SensorEventListener listener, Sensor sensor, int samplingPeriodUs)` Jednotlivé dáta od senzorov sú uložené do polí, vektorov. Následne sme pripravený získavané dáta zpracovať, filtrovať.

4.2 Komplementárna senzorová fúzia

Komplementárna senzorová fúzia je implementovaná v súbore `SensorFusion.java` v triede `calculateFusedOrientationTask`, pričom sa skladá z niekoľkých ďalších metód. Trieda, ktorá implementuje komplementárny filter sa nazýva `calculateFusedOrientationTask()` a dedí z triedy `TimerTask`. `TimerTask` je Úloha, ktorá môže byť naplánovaná na jednorazové alebo opakované spustenie pomocou časovača. Na triedu komplementárnej fúzie bol aplikovaný časovač, ktorý vytvára inštanciu triedy komplementárnej fúzie každú sekundu, ak sú dáta snímačov dostupné. Dáta skombinované zo všetkých troch snímačov sú dostupné

v poli fusedOrientation, z ktorého skladáme trénovaciu sadu dát. Class diagram sensorovej fúzie je zobrazený na obrázku 4.3.

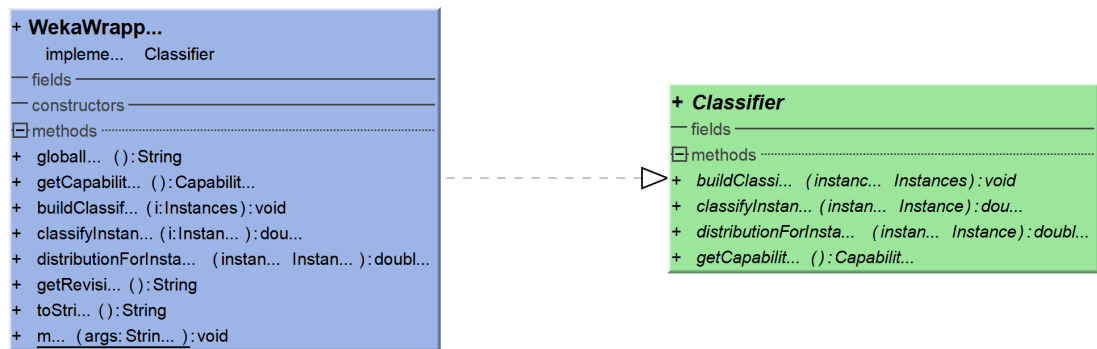


Obr. 4.2: Class diagram sensorovej fúzie.

Fúzia je implementovaná ako trieda SensorFusion, pričom sa jedná o službu (Service), ktorá beží na pozadí aplikácie a priebežne zbiera dáta od jednotlivých senzorov. Užívateľ aplikácie môže zbierať dáta aj počas toho ako zavrel aplikáciu. Pri implementovaní fúzií bolo dôležité využiť funkcie, ktoré poskytuje Android API, aby sme získali napríklad absolútnu orientáciu z akcelerometra a magnetometra. Taktiež nám Android API poskytol funkciu ako získať rotačný vektor z dát gyroskopu. Pri implementácii komplementárnej sensorovej fúzie som vychádzal z [12].

4.3 Nástroj WEKA

Nástroj WEKA bol používaný pri implementácii a bol taktiež využitý v kóde programu. Aby sme importovali knižnicu nástroja WEKA bolo potrebné pridať do projektu súbor **wekaSTRIPPED.jar** a pridať ho do cesty vytvárania projektu Android. Najdôležitejšia funkcia tejto triedy sa nazýva **classify()**, ktorá umožňuje a používa sa na klasifikovanie neoznačených dát, ktoré sú klasifikátoru predané. Z balíčku dostupných klasifikátorov bol vybraný klasifikátor J48. Nástroj je možné nainštalovať z oficiálnej stránky WEKA. Potom čo máme dáta zo senzorov akcelerometra, magnetometra a gyroskopu vyfiltrované, ošetrované je potrebné dáta pripraviť do takého formátu, aby im bol schopný algoritmus a nástroj WEKA porozumieť. Pomocou triedy **CollectorActivity** sme implementovali a zadefinovali vzhľad aplikácie. V tejto triede sa spúšťa služba **SensorsService**, ktorá má za úlohu zbierať dáta na pozadí a vytvoriť súbor, ktorý použijeme pri trénovaní klasifikátora v nástroji WEKA. Pri implementácii **CollectorActivity** a služby **SensorsService** som vychádzal z [8].



Obr. 4.3: Class diagram triedy WekaWrapper.

4.4 Predikcia

Na detekovanie fyzickej aktivity je potrebná trieda `WekaClassifier` a jej metóda `classify()`. Do tejto metódy sú dáta vkladane už bez označenia aktivity. Natrénovaný klasifikačný algoritmus musí byť už schopný rozpoznať o akú aktivitu sa jedná a poslať na výstup túto aktivitu. Nástroj je implementovaný v triede `WekaWrapper` ktorá implementuje rozhranie klasifikátor. Samotný natrénovaný klasifikačný algoritmus, ktorý používame na klasifikáciu dát sa nachádza v triede `WekaClassifier`.

Kapitola 5

Testovanie

Pri vývoji a implementovaní bakalárskej práce, bola aplikácia priebežne testovaná na mobilnom zariadení Oneplus X s operačným systémom Android s verziou Nougat 7.1, ktorý obsahuje senzory akcelerometra, gyroskopu a magnetometra. Aplikácia bola taktiež testovaná a debugovaná vo vývojovom prostredí Android Studio založené na IntelliJ IDEA od firmy Google. Pre odstraňovanie bugov¹ bol využitý Debugger zabudovaný priamo v Android Studio. Pri testovaní aplikácie, bola hlavne testovaná presnosť natrénovaného klasifikačného algoritmu pri rozpoznávaní fyzických aktivít užívateľov, ktorý sme zvolili. Je veľmi dôležité, aby klasifikačný algoritmus bol natrénovaný na veľkej sade tréningových dát, aby sa zväčšila jeho presnosť, a tým pádom sa zmenšila jeho chybovosť. Aby bola vytvorená tréningová sada dát dostatočná pre natrénovanie klasifikátora, musí nazbierať dáta do tejto tréningovej sady dát veľké množstvo užívateľov, pretože každý užívateľ má rozdielny pohyb pri chôdzi alebo pri behaní. Na testovanie bol využitý taktiež nástroj WEKA, ktorý nám poskytol určité štatistiky o danom natrénovanom klasifikátore, ktoré sú zobrazené v tabuľke. Natrénovaný klasifikátor bol vyhodnotený a otestovaný krížovou validáciou² na základe zozbieraných dát pomocou jedného užívateľa v časovom úseku 11 minút, teda pre každú aktivitu boli zbierané dáta zhruba okolo 3 minút.

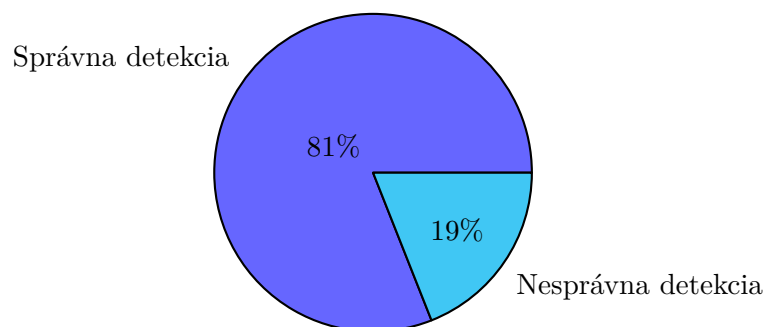
Zhrnutie	
Celkový počet inštancií	2529
Správne klasifikované inštancie	2057 (81.33%)
Nesprávne klasifikované inštancie	472 (18.66%)
Stredná absolútna chyba	0.1397

Tabuľka 5.1: Štatistické údaje klasifikačného algoritmu.

Z tabuľky sú dôležité počty správne a nesprávne klasifikovaných inštancií. Údaj správne klasifikované inštancie udáva počet správne detekovaných aktivít. Stredná absolútna chyba označuje množstvo, ktoré sa používa na meranie toho, aké blízke predpovede sú k výsledným výsledkom. Na základe presnosti klasifikačného algoritmu, ktorá je 81,3% môžeme rozhodnúť, že daný klasifikačný algoritmus je pomerne dobre presný pri detekovaní aktivít. Pri testovaní bola taktiež vygenerovaná matica zámen, tabuľka ktorá sa veľmi často používa na opis výkonnosti klasifikačného modelu na súbor testovacích údajov, pre ktoré sú známe

¹ Chyba v počítačovom programe

² Metóda zisťovania, ako moc bude model štatistickej analýzy ovplyvňovať nezávislé vzorky dát.



Obr. 5.1: Úspešnosť detekovania jednotlivých aktivít.

		Predpovedané		
		Státie	Chôdza	Beh
Aktuálna aktivita	Státie	760	28	9
	Chôdza	18	588	188
	Beh	23	206	709

Tabuľka 5.2: Matica zámen.

skutočné hodnoty. Ak by bol klasifikačný systém vycvičený na rozlíšenie medzi státím na mieste, chodením a behaním, tak táto tabuľka matica zámen obsahuje výsledky testovania algoritmu. Za predpokladu, že vzorka obsahuje 797 státí, 794 chodení a 938 behaní, môže matica zámen vyzeráť nasledujúco

V tejto matici zámen, zo 797 dát státí, systém predpovedal, že 28 dát zle rozpoznal, pretože išlo o chôdzu a 9 o beh. Podobne pri 794 dát o chôdzi, systém predpovedal, že 188 dát zle rozpoznal, pretože išlo o beh a nie chôdzu, takisto zle predpovedal ďalších 18 dát, kedy sa jednalo o státie. Z tabuľky sa môžeme dozvedieť, že daný systém má väčšie problémy pri odlišovaní aktivít chôdze a behu, ale vie veľmi dobre rozlíšiť státie od iných aktivít. Všetky správne odhady sa nachádzajú v diagonále tabuľky, ľahko sa dá tabuľka teda skontrolovať vzhľadom na chyby, ktoré sa budú vyskytovať mimo tejto diagonály. Z testovania pomocou matice zámen nám vyplýva, že klasifikačný algoritmus má menšie problémy pri detekovaní medzi aktivitami chôdze a behom. Tento problém by sa vyriešil, dlhším zbieraním dát pre dané aktivity.

5.1 Efektivita a využitie aplikácie

Ako už bolo spomínané, efektivita výslednej aplikácie veľmi zaleží na akom počte užívateľov bol zvolený klasifikačný algoritmus natrénovaný. Čím je počet užívateľov, ktorý vytvárali tréningovú sadu dát väčší, tým je aj klasifikačný algoritmus pri rozpoznávaní jednotlivých aktivít presnejší. Ak by bola tréningová sada vytvorená za krátky čas a s veľmi malým počtom užívateľov, potom je aj klasifikačný algoritmus nepresnejší a ťažko rozoznáva rozdiel pri chôdzi a pri behaní pri rôznych užívateľoch, ktorý túto aplikáciu používajú, pretože vytvorený rozhodovací strom je veľmi malý a teda s malým počtom uzlov. Pri testovaní efektivity výslednej aplikácie, boli tieto dva spomenuté prípady otestované. Najprv bola vytvorená tréningová sada dát s malým počtom užívateľov a za krátky čas pre jednotlivé fyzické aktivity. Následne bolo skúmané s akou presnosťou je schopný natrénovaný algorit-

mus rozpoznať aktivity ak ho používa viacero užívateľov, aj tí ktorí túto tréningovú sadu nevytvárali. Pri tomto testovaní bola vytvorená aj tabuľka, ktorá reprezentuje akú presnosť takto natrénovaný klasifikátor dosiahol. Platí teda pravidlo, čím väčšia tréningová sada, tým efektívnejší algoritmus klasifikácie.

Výslednú aplikáciu môžeme ďalej využiť pri monitorovaní fyzickej aktivity jednotlivých užívateľov. Implementácia sa dá využiť v rôznych mobilných aplikáciách, ktoré monitorujú a pomáhajú užívateľom s modernými zariadeniami dosiahnuť dennú fyzickú aktivitu, ktorí pri sedavom zamestnaní majú väčšinou menej fyzickej aktivity ako je považované za zdravé, v dôsledku čoho vznikajú závažné komplikácie so zdravím. V dnešnej modernej dobe je dôležité aby súčasné najnovšie technológie pomáhali a boli tu nielen pre mladších ľudí ale aj pre seniorov. Implementácia sa dá využiť napríklad pri vzdialenom monitorovaní starších ľudí s menšími zdravotnými komplikáciami, ktoré nevyžadujú neustály dohľad v nemocnici. Ak má starší pacient problémy s pohybovou aktivitou, je dôležité aby bola pohybová aktivita tohto pacienta neustále monitorovaná a v prípade náhleho pádu daného pacienta sa mohlo zakročiť včas a predišlo sa tým zbytočným komplikáciám, kedy sa pacient nie je schopný dovolať na tiesňovú linku a zavolať o pomoc. Veľká väčšina mladých ľudí si v dnešnej dobe nenájde čas pre starších rodičov, ktorých nechcú zbytočne umiestniť do zariadení pre dohľad nad staršími ľuďmi a preto väčšinou zostávajú sami doma, bez dozoru. Je preto veľmi žiadúce a potrebné, aby boli títo starší ľudia obzvlášť s pohybovými komplikáciami a ochoreniami chrbtice, kĺbov monitorovaní a v prípade nehody im bola poskytnutá neodkladná zdravotná pomoc.



Obr. 5.2: Využitie aplikácie v dnešnej dobe. [11]

Kapitola 6

Záver

Cielom tejto práce bolo zoznámiť sa s problematikou využitia mobilných zariadení ako snímačov rôznych veličín, využitím týchto snímačov v rôznych aplikáciach, prístupom k týmto snímačom, ošetrovanie chýb a výchylok snímačov, algoritmom pre klasifikáciu dát zo snímačov a Android aplikácie, ktorá vhodne demonštruje fúziu dát zo snímačov mobilného zariadenia pomocou komplementárnej sensorovej fúzie. Pre využitie snímačov mobilného zariadenia ako uzlov sensorovej siete, bola navrhnutá aplikácia, ktorá číta dáta súčasne z troch snímačov a využíva fúziu dát z týchto snímačov, pri rozpoznávaní jednoduchých fyzických aktivít. Práca sa zaoberala aký veľký vplyv fúzia dát na efektivitu a presnosť rozpoznávania aktivít má. Taktiež bolo skúmané ako veľký súbor trénovacej sady dát je potrebné na to, aby bola detekcia týchto aktivít dostatočne rozpoznateľná pre rôznych užívateľov používajúcich túto aplikáciu. Skúmané boli taktiež chyby, výchyľky, ktoré snímače nie sú schopné odstrániť. Výsledná aplikácia FusionApp je schopná detekovať základné pohyby s vysokou presnosťou na dlhé časové intervaly, využíva pritom klasifikačný algoritmus J48, ktorý patrí pod kategóriu algoritmov rozhodovacích stromov, komplementárnu sensorovú fúziu, ktorá zabezpečuje aby sa dáta zo snímačov sa nerozkmitali a nespôsobili výchyľky pri detekovaní a software WEKA. Úspešnosť natrénovaného klasifikačného algoritmu pri detekcii aktivít bola otestovaná s výsledkom úspešnosti 81,3%, čo predstavuje vysoké skóre a to hlavne vďaka implementovanej sensorovej fúzii. Aplikácia je schopná sa naučiť detekciu aj iných aktivít ako napríklad pádu, ktoré by do budúcnosti mohlo slúžiť pre monitorovanie starších ľudí a včasné poskytnutie im zdravotnej pomoci.

Literatúra

- [1] Location and Sensors overview in Android. [online], naposledy cit. 08.05.2017.
URL https://developer.android.com/guide/topics/sensors/sensors_overview.html
- [2] Motion Sensors. [online], naposledy cit. 15.04.2017.
URL https://developer.android.com/guide/topics/sensors/sensors_motion.html
- [3] PressureNet is joining Sunshine. [online], naposledy cit. 20.01.2017.
URL <http://www.pressurenet.io/>
- [4] Attribute-Relation File Format (ARFF). [online], Apr 2008.
URL <http://www.cs.waikato.ac.nz/ml/weka/arff.html>
- [5] Senzory v mobilních telefonech od A do Z. [online], Nov 2014.
URL <https://www.beryko.cz/blog/recenze/senzory-v-mobilnich-telefonech-od-a-do-z.html>
- [6] Ambient Light Sensor Smartphone. [online], Apr 2017.
URL <http://hp.creativepedia.net/2017/04/27/ambient-light-sensor-smartphone/>
- [7] Albert, M.; Toledo, S.; Shapiro, M.; aj.: Using Mobile Phones for Activity Recognition in Parkinson's Patients. *Frontiers in Neurology*, ročník 3, 2012: str. 158, ISSN 1664-2295, doi:10.3389/fneur.2012.00158.
URL <http://journal.frontiersin.org/article/10.3389/fneur.2012.00158>
- [8] Campbell, A. T.: Activity Recognition. [online].
URL <http://www.cs.dartmouth.edu/~campbell/cs65/lecture22/lecture22.html>
- [9] Elmenreich, W.: The different types of sensor fusion: complementary, competitive, and cooperative. [online], Mar 2011.
URL <https://netwerkt.wordpress.com/2011/03/30/the-different-types-of-sensor-fusion-complementary-competitive-and-cooperative/>
- [10] Formento, P. C.; Acevedo, R.; Ghoussayni, S.; aj.: Gait Event Detection during Stair Walking Using a Rate Gyroscope. *Sensors*, ročník 14, č. 3, 2014: s. 5470–5485, ISSN 1424-8220, doi:10.3390/s140305470.
URL <http://www.mdpi.com/1424-8220/14/3/5470>
- [11] Holland, T. M.: Turning Smartphones Into Remote Monitoring Devices for Older Adults. [online], jan 2017.

- URL <https://insights.samsung.com/2016/01/12/turning-smartphones-into-remote-monitoring-devices-for-older-adults/>
- [12] Lawitzki, P.: Android Sensor Fusion Tutorial. [online], feb 2014, naposledy cit. 11.05.2017.
URL <https://www.codeproject.com/Articles/729759/Android-Sensor-Fusion-Tutorial>
- [13] Martin Chroust, F. K.: Smartphony mají 19 smyslů. Znáte je všechny? [online], Feb 2015.
URL <http://www.mobilmania.cz/clanky/smartphony-maji-19-smyslu-znate-je-vsechny/sc-3-a-1329584/default.aspx>
- [14] Olšák, P.: *TEXbook naruby*. Konvoj, 2001, ISBN I80-7302-0007-6.
- [15] Pai, A.: Google adds low-power step counting to Android 4.4. [online], Nov 2013.
URL <http://www.mobihealthnews.com/26977/google-adds-low-power-step-counting-to-android-4-4>
- [16] Ravi, N.; Dandekar, N.; Mysore, P.; aj.: Activity Recognition from Accelerometer Data. In *Proceedings of the 17th Conference on Innovative Applications of Artificial Intelligence - Volume 3*, IAAI'05, AAAI Press, 2005, ISBN 1-57735-236-x, s. 1541–1546.
URL <http://dl.acm.org/citation.cfm?id=1620092.1620107>
- [17] Saxena, R.: How Decision Tree Algorithm works. [online], jan 2017.
URL <http://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/>
- [18] Shoaib, M.; Bosch, S.; Incel, O. D.; aj.: Fusion of Smartphone Motion Sensors for Physical Activity Recognition. *Sensors*, ročník 14, č. 6, 2014: s. 10146–10176, ISSN 1424-8220, doi:10.3390/s140610146.
URL <http://www.mdpi.com/1424-8220/14/6/10146>
- [19] Shoaib, M.; Bosch, S.; Incel, O. D.; aj.: A Survey of Online Activity Recognition Using Mobile Phones. *Sensors*, ročník 15, č. 1, 2015: s. 2059–2085, ISSN 1424-8220, doi:10.3390/s150102059.
URL <http://www.mdpi.com/1424-8220/15/1/2059>
- [20] Su, J. B.: Sensors and Cellphones. [online], Apr 2012.
URL <https://web.stanford.edu/class/cs75n/Sensors.pdf>
- [21] Su, X.; Tong, H.; Ji, P.: Activity recognition with smartphone sensors. *Tsinghua Science and Technology*, ročník 19, č. 3, jun 2014: s. 235–249, doi:10.1109/tst.2014.6838194.
URL <https://doi.org/10.1109%2Ftst.2014.6838194>
- [22] Witten, I. H.; Frank, E.; Hall, M. A.: *Data Mining: Practical Machine Learning Tools and Techniques*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., třetí vydání, 2011, ISBN 0123748569, 9780123748560.

Príloha A

Obsah CD

Na priloženom CD disku sú uložené všetky zdrojové súbory, manuál pre preloženie aplikácie a inštalačné súbory, ktoré je potrebné mať nainštalované pri spúšťaní tejto aplikácie. Na disku sú taktiež priložené zdrojové súbory $\text{\LaTeX}$ pre zostavenie PDF súboru práce a súbor aplikácie formátu APK pre distribúciu aplikácií mobilného systému Android.

Príloha B

Manuál

Nasledujúci text slúži ako návod k nainštalovaniu, spusteniu a preloženiu aplikácie.

B.1 Postup inštalácie aplikácie

1. Pripojiť mobilné zariadenie do počítača a zapnúť USB Debugging Mode na mobilnom zariadení.
2. Z <https://developer.android.com/studio/releases/platform-tools.html#download> stiahnuť SDK Platform-Tools a vyextrahovať na pracovnú plochu alebo na CD disk použiť priečinok adb.
3. Do priečinka adb na pracovnej ploche presunúť SensorFusion.apk súbor v priečinku install, tak aby sa súbor SensorFusion.apk náchadzal pri súbore adb.exe.
4. Otvoriť terminál, a cez príkaz cd sa dostať k priečinku so súbormi SensorFusion.apk a adb.exe.
5. Príkazom v termináli: adb install SensorFusion.apk nainštalovať aplikáciu na zariadenie.

B.2 Postup preloženia aplikácie

1. Z <https://developer.android.com/studio/index.html> stiahnuť a nainštalovať Android Studio alebo použiť inštaláciu na CD disku v priečinku IDE.
2. Z <https://github.com/rjmarsan/Weka-for-Android/blob/master/wekaSTRIPPED.jar?raw=true> stiahnuť súbor WekaStripped.jar alebo použiť súbor na CD disku v priečinku Weka a pridať súbor ako knižnicu.
3. Otvoriť projekt pomocou Open Project v Android Studiu.
4. Pripojiť mobilné zariadenie.
5. Preložiť a spustiť aplikáciu.

Príloha C

Súbor typu ARFF

Listing C.1: Obsah súboru typu ARFF

```
// Informacie o hlavicke
// list atributov , feature vector
@attribute fft_coef_0042 numeric
@attribute fft_coef_0043 numeric
@attribute fft_coef_0044 numeric
@attribute fft_coef_0045 numeric
@attribute fft_coef_0046 numeric
@attribute fft_coef_0047 numeric
@attribute fft_coef_0048 numeric
@attribute fft_coef_0049 numeric
@attribute fft_coef_0050 numeric
@attribute fft_coef_0051 numeric
@attribute fft_coef_0052 numeric
@attribute fft_coef_0053 numeric
@attribute fft_coef_0054 numeric
@attribute fft_coef_0055 numeric
...

@attribute max numeric
@attribute label {standing , walking , running}

// Data suboru features.arff vyzeraju nasledovne

@data
14.614832,2.073224,1.465669,1.394093,0.640366,0.847062,0.512558,
0.603695,0.404458,0.160662,0.26455,0.287018,0.595104,standing
...
```