



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

DEPARTMENT OF BIOMEDICAL ENGINEERING

DETEKCE PÁDU A AUTONEHODY PROSTŘEDNICTVÍM ANDROID SMARTPHONU

DETECTION OF CAR ACCIDENT AND COLLAPSE BY ANDROID SMARTPHONE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Tomáš Repčík

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Denisa Maděránková, Ph.D.

BRNO 2019



Bakalářská práce

bakalářský studijní obor **Biomedicínská technika a bioinformatika**

Ústav biomedicínského inženýrství

Student: Tomáš Repčík

ID: 195194

Ročník: 3

Akademický rok: 2018/19

NÁZEV TÉMATU:

Detekce pádu a autonehody prostřednictvím Android smartphonu

POKYNY PRO VYPRACOVÁNÍ:

1) Proveďte literární rešerši problematiky automatické detekce pádu a autonehody a problematiky strojového učení. 2) Navrhněte model detektoru pro aplikaci v Android zařízení. 3) Navrhněte metodu sběru trénovacích dat a jejich zpracování do modelu. 4) Navržený model implementujte do aplikace pro Android zařízení a aplikaci otestujte. 5) Diskutujte dosažené výsledky a zhodnoťte možnosti využití daného řešení detektoru pádu a autonehody.

DOPORUČENÁ LITERATURA:

[1] MITCHELL, T. Machine Learning. McGraw Hill. 1997. ISBN 978-0-07-042807-2.

[2] GARETH, James, et al. An Introduction to Statistical Learning. Springer. 2013

Termín zadání: 4.2.2019

Termín odevzdání: 24.5.2019

Vedoucí práce: Ing. Denisa Maděránková, Ph.D.

Konzultant:

prof. Ing. Ivo Provazník, Ph.D.
předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Bakalárska práca „Detekcia pádu a autohavárie pomocou Android smartphonu“ popisuje teoretické spracovanie pádu človeka a autonehody. Pre vytvorenie vhodného modelu pre detekcie bol vykonaný zber údajov simulovaných pádov, denných aktivít a jazdy autom za pomoci 23 dobrovoľníkov. Pre detekciu pádu bola vytvorená neurónová sieť s presnosťou 86 %. Detekcia autohavárie prináša algoritmus využívajúci akcelerometer a GPS, ktorý podľa prítomnosti vysokého preťaženia a absencie jazdy dokáže vyhodnotiť prítomnosti kolízie. Aplikácia bola následne testovaná pomocou všetkých dostupných dát pomocou dobrovoľníkov, ktorí ju používali po dobu 520 h.

Kľúčové slová

Pád, autohavária, Android, Python, Java, strojové učenie, neurónová sieť

Abstract

Bachelor thesis “Detection of car accident and collapse by Android smartphone” describes theoretical basics of a collapse and a car accident. With help of 23 volunteers, data has been collected from daily tasks, simulating collapses and driving. Based on that a model with capability to classify a car accident was created. The detection of collapse uses neural network with accuracy of 86 %. The detection of car accident uses accelerometer and GPS. To indicate the car accident a high acceleration created by the accident and absence of moving car is required. Next, the application was tested by all available data and by the volunteers in daily life, who used the application for 520 h.

Keywords

Collapse, car accident, Android, Python, Java, machine learning, neural network

Bibliografická citácia:

REPČÍK, Tomáš. *Detekce pádu a autonehody prostřednictvím Android smartphonu*. Brno, 2019, 72 s. Bakalárska práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačných technológií, Ústav biomedicínskeho inžinierstva. Vedúci práce: Ing. Denisa Maděránková, Ph.D.

Prehlásenie

„Prehlasujem, že svoju bakalársku prácu na tému *Detekce pádu a autonehody prostřednictvím Android smartphonu*, som vypracoval samostatne pod vedením vedúceho bakalárskej práce a s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce. Ako autor uvedenej záverečnej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto záverečnej práce som neporušil autorské práva tretích osôb, hlavne som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a som si plne vedomý porušenia ustanovení § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., taktiež možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníku č. 40/2009 Sb.

V Brne dňa

.....

podpis autora

Pod'akovanie

Ďakujem vedúcej bakalárskej práce Ing. Denise Maděránkovej, Ph.D. za pedagogickú a odbornú pomoc a ďalšie rady pri spracovaní bakalárskej práce. Rovnako by som chcel poďakovať za podporu Ing. Filipa Maleňáka z aplikácie Záchranka, vďaka ktorému došlo k vzniku aplikácie. Rovnaká vďaka patrí aj všetkým účastníkom, ktorí sa zapojili do zberu dát a testovania aplikácie.

V Brne dňa

.....

podpis autora

Obsah

Úvod.....	1
1. Teoretický úvod.....	2
1.1 Popis pádu osoby a podobných situácií	2
1.2 Popis akcelerácie pri rôznych situáciach.....	3
1.3 Popis iných veličín	6
1.4 Problematika strojového učenia	7
1.5 Popis a detekcia autohavárie	10
2. Návrh modelu pre Android	12
2.1 Detekcia pádu.....	13
2.2 Výpočet parametrov pre detekciu pádu.....	14
2.2.1 Parametre špecifické pre akceleráciu.....	14
2.2.2 Všeobecné parametre	17
2.3 Trénovanie neurónovej siete	17
2.4 Detekcia autohavárie	20
3. Zber, spracovanie a analýza dát	22
3.1 Zber dát	22
3.2 Spracovanie dát detekcie pádu	24
3.3 Výsledky tréovania neurónovej siete	29
3.4 Zber a analýza dát pre detekciu autohavárie	32
4. Zloženie Android aplikácie.....	35
4.1 Základne rozhranie aplikácie	35
4.2 Fungovanie služby detekcie	37
4.3 Doze režim	39
4.4 Optimalizácia služby detekcie.....	41
5. Vyhodnotenie aplikácie	43
5.1 Interné testovanie pomocou Google Play	44
Záver	48
Literatúra.....	49
Zoznam skratiek a značiek.....	52
Zoznam príloh.....	53
Príloha č. 1	54
Príloha č. 2	60
Príloha č. 3	61
Príloha č. 4	63

Zoznam obrázkov

Obr. 1.1 Zmena akcelerácie v čase, prebraté z [3].....	2
Obr. 1.2 Zmena magnitúdy akcelerácie počas pádu človeka a mobilu.....	3
Obr. 1.3 Zmena magnitúdy akcelerácie pri sadnutí na mäkký/tvrдый povrch.....	4
Obr. 1.4 Zmena magnitúdy akcelerácie skoku a chôdze	5
Obr. 1.5 Zmena uhlovej rýchlosti pri rôznych situáciách.....	6
Obr. 2.1 Diagram všeobecného algoritmu detekcie.....	12
Obr. 2.2 Diagram algoritmu detekcie pádu.....	13
Obr. 2.3 Diagram algoritmu detekcie autonehody.....	21
Obr. 3.1 Vizualizácia štandardizovaných parametrov akcelerácie	25
Obr. 3.2 Vizualizácia štandardizovaných parametrov uhlovej rýchlosti	26
Obr. 3.3 Vizualizácia štandardizovaných parametrov uhlového zrýchlenia	26
Obr. 3.4 ICA aplikované na vybrané parametre	28
Obr. 3.5 Úspešnosť/chybovosť neurónových sietí v postupnosti od 3 do 10 neurónov .	30
Obr. 3.6 Distribúcia menej špecifického modelu voči iným udalostiam	30
Obr. 3.7 Distribúcia menej špecifického modelu voči pádom.....	30
Obr. 3.8 Test NN na reálnych dátach - úspešnosť modelov	31
Obr. 3.9 Popis akcelerácie, rýchlosti a zmeny pohybového stavu (červené línie).....	32
Obr. 3.10 Zmena pravdepodobnosti jednotlivých pohybových stavov počas jazdy.....	32
Obr. 3.11 Histogram rozptylu akcelerácie pri jazde auta	33
Obr. 4.1 Ukážka aplikácie – naľavo hlavné menu, v strede alarm, napravo notifikácia služby na pozadí.....	37

Zoznam tabuliek

Tab. 1.1 Porovnanie presnosti modelov detekcie pádu prebrané z [12]	9
Tab. 3.1 Počet dobrovoľníkov/udalostí	24
Tab. 5.1 Využitie aplikácie medzi 12 dobrovoľníkmi	45

ÚVOD

Podľa štatistík, pád človeka je druhá najčastejšia príčina neúmyselnej smrti, čo činí takmer 700 000 úmrtí ročne. Za rok sa udeje 40 000 000 pádov, ktoré si vyžadujú zdravotnú starostlivosť, pričom najrizikovejšia skupina ľudí sú seniori nad 65 rokov [1]. Ďalšou vedúcou príčinou usmrtenia sú autonehody, ktoré si ročne zoberú aj 40 000 životov. Našťastie vďaka neustále zvyšujúcej sa bezpečnosti áut sa toto číslo každým rokom znižuje [2].

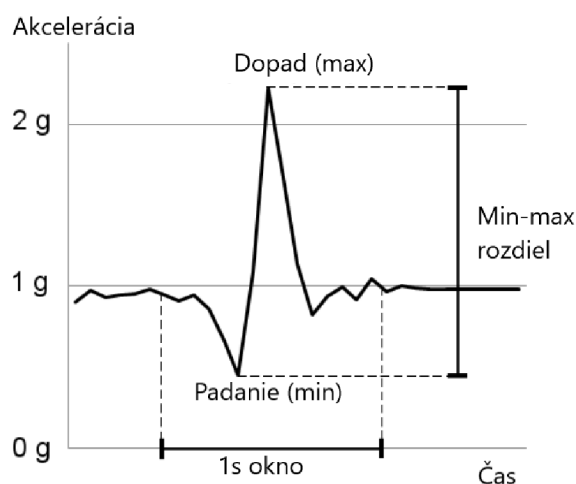
V tejto bakalárskej práci sa pokúsime priblížiť možné riešenie pre včasnú detekciu pádu človeka, ako aj autohavárie pomocou Android smartphonu. Dnešné Androidy majú takú výpočtovú silu, že nie je problém na nich kľásť požiadavky umelej inteligencie. Súčasní výrobcovia sa predbiehajú, kto vytvorí inteligentnejší fotoaparát, rozpoznanie tváre, objektov a pod., čo vyžaduje vysokú mieru optimalizácie. Najnovšie verzie Androidu majú nové adaptácie v podobe hardwarovej akcelerácie a rozhrania pre modely umelej inteligencie, ktoré znižuje nároky na systém. Tieto nové vytvorené možnosti sa pokúsime implementovať do Android aplikácie, ktorá zastrešuje tieto detekcie.

Podľa dostupnej literatúry a technických možností zvolíme najlepší možný postup, ktorý bude dosiahnuteľný. Pre vytvorenie modelu je vykonaný zber dát od dobrovoľníkov, aby sme zabezpečili čo najvyššiu variabilitu údajov a generalizačnú schopnosť modelu. Aplikácia je zostrojená v Android studio pomocou jazyka Java a na testovanie je distribuovaná na Google Play pomocou interného testovania.

1. TEORETICKÝ ÚVOD

1.1 Popis pádu osoby a podobných situácií

Pád je definovaný ako udalosť, ktorá má za následok neúmyselný kolaps človeka na zem, povrch alebo inú rovnú plochu [1]. Závažnosť zranení je potom závislá od druhu pádu a hlavne aj od veku človeka. Tieto zranenia sú hlavne spôsobené vysokou zmenou akcelerácie, ktorá sa stane hlavným parametrom pre detekciu pádu. Priebeh môžeme rozdeliť na 3 fázy: fázu pred pádom, samotný pád a obdobie po páde. V tejto časti sa budeme zaoberať len samotným pádom, ale pre samotnú detekciu sú ostatné dve fázy rovnako kľúčové. Pre lepšiu predstavu poslúži zjednodušený graf zmeny akcelerácie pri páde. Pri meraní dochádza k záznamu akcelerácie z troch osí, ale pre zjednodušenie budú všetky tieto tri osi prepočítané do zjednoteného priebehu magnitúdy celkovej akcelerácie [3].



Obr. 1.1 Zmena akcelerácie v čase, prebraté z [3]

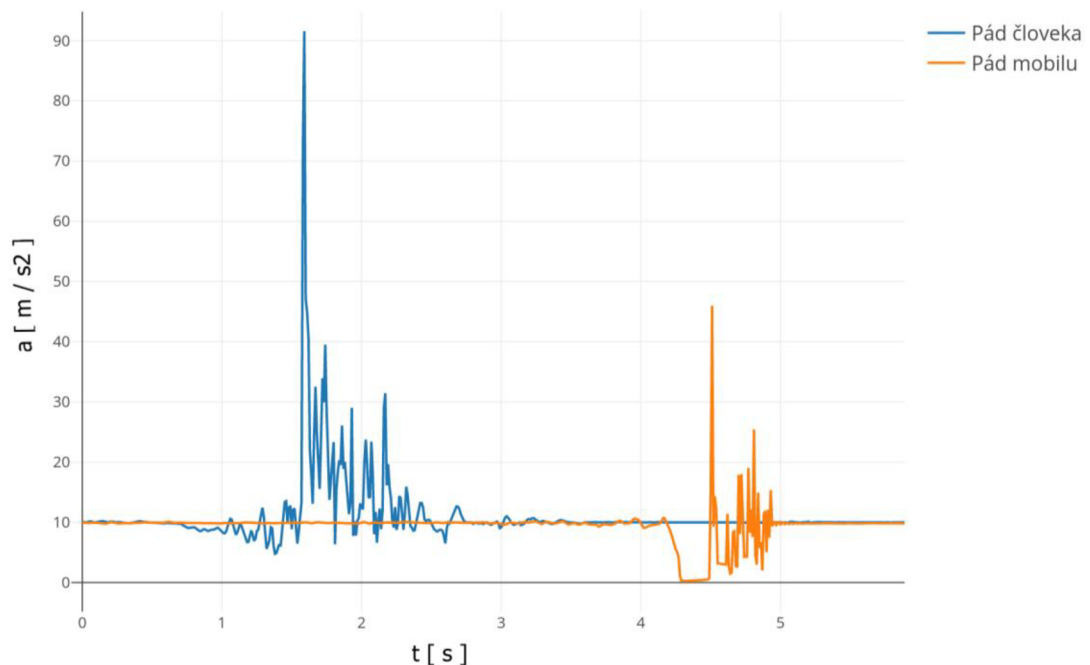
Príčiny pádu a ich priebehu môžu byť rôzne, ale budú ich spájať dve hlavné charakteristiky, ktoré sú reprezentované na Obr. 1.1. Pokles gravitačného zrýchlenia pod úroveň 1g spôsobený voľným pádom na zem (nikdy nedosiahne nulových hodnôt, pretože nikdy nedosiahneme dokonalých podmienok) a následný pík nad 2g spôsobený nárazom tela na zem [3]. Priebeh môže obsahovať aj viac pík, keď jednotlivé časti tela dopadajú v istej sekvencii ako napr. nohy, trup a ostatok tela [4]. Hlavný interval, počas ktorého človek prežíva pád, trvá v priemere maximálne jednu sekundu [3].

Problém nie je s detekciou týchto charakteristík, ale s ich odlíšením od situácií, pri ktorých sa tiež vyskytujú miesta s poklesom gravitačného zrýchlenia a následným vysokým píkom aj nad 3g, ako napr. kráčanie po schodoch, beh, skákanie, sad, ľah,

manipulácia s telefónom, až jeho samotný pád a podobne. Všetky tieto úkony sú si navzájom podobné, ale nie neodlíšiteľné, pričom sú vysoko závislé aj na polohe, pohybe, niektorých fyzických atribútoch človeka a od typu povrchu, s ktorým prichádza do interakcie. Z tohto dôvodu sa v tejto práci obmedzíme na mobilné zariadenia, ktoré človek nosí priamo pri sebe, teda vo vrecku v nohaviciach alebo košeli, bunde. Keby zariadenie bolo umiestnené mimo človeka, napr. v batohu alebo v kabelke, meranie by bolo výrazne skreslené druhom uloženia a manipuláciou s daným predmetom. Napr. prudšie uloženie kabelky by mohlo evokovať detekciu pádu, alebo jej poloha pri páde by zabránila jeho detekcii. Nevyklučuje sa úplná nefunkčnosť algoritmu, ale došlo by k drastickému obmedzeniu schopností detekcie. Je možné nájsť aj štúdie, ktoré popisujú potrebu umiestnenia akcelerometra na hrudníku pre najlepšiu záznam voľného pádu a následného dopadu [5], čo ale v našom prípade nebude dosiahnuté a budeme si musieť vystačiť s dátami z mobilu.

1.2 Popis akcelerácie pri rôznych situáciách

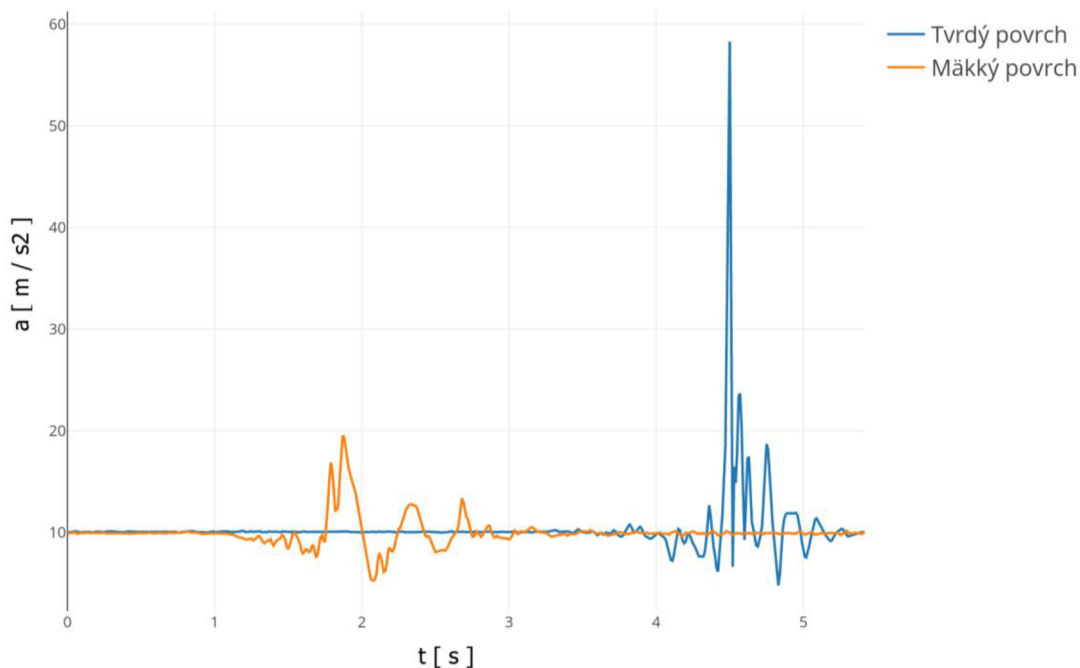
Signály zobrazené na grafoch v tejto časti sú nasnímané mobilom Huawei P9 uloženým vo vrecku nohavíc. Podľa najjednoduchšieho úsudku môžeme prísť na to, že najviac sa budú podobáť pádu človeka a pádu samotného mobilu. Na Obr. 1.2 môžeme vidieť pád človeka bokom na tvrdý povrch a pád mobilu približne zo 40 cm tiež na tvrdý povrch.



Obr. 1.2 Zmena magnitúdy akcelerácie počas pádu človeka a mobilu

Ako môžeme vidieť na Obr. 1.2, obidva priebehy zdieľajú istú podobnosť, keď zohľadníme silové pôsobenie na človeka a mobil a ich nepomer hmotností. Preto pre model je potrebné zohľadniť hmotnosť človeka, ktorá ako jeden z parametrov môže prispieť k lepšej detekcii, pretože veľkosť zrýchlenia pri páde je závislá od hmotnosti.

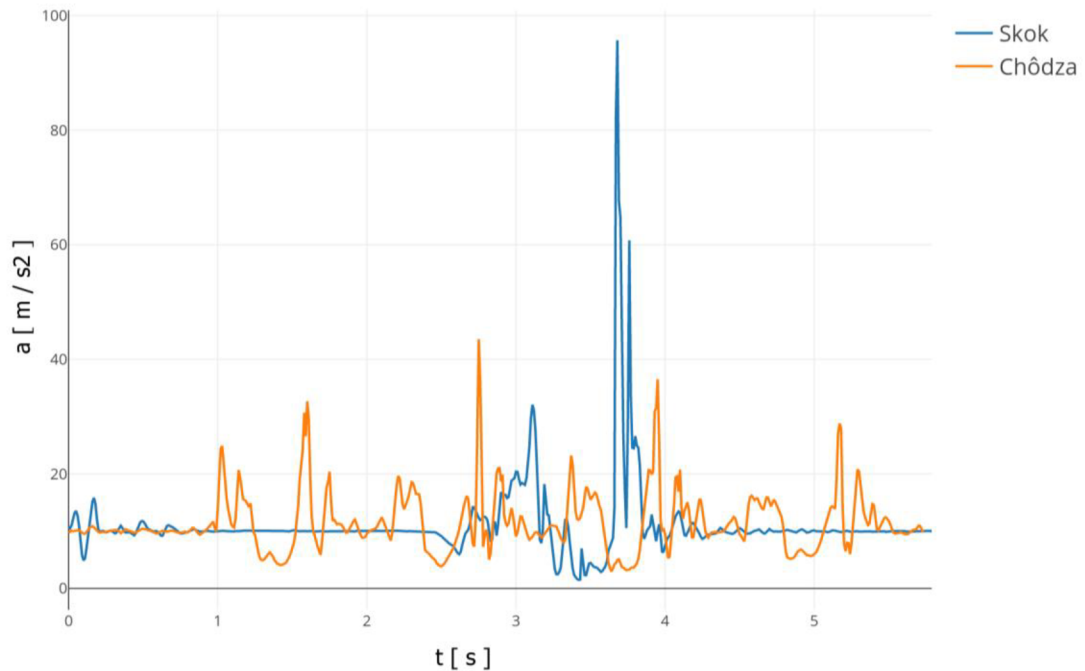
Ako ďalší znak si môžeme všimnúť odlišnosť priebehu poklesu gravitačného zrýchlenia počas padania na zem. Pri páde človeka nie je táto pasáž až tak výrazná, ale je prítomná. Rušenie je spôsobené voľným umiestnením mobilu vo vrecku. Pri páde môže dôjsť k interakcii s človekom a ďalším následným iným silovým pôsobeniam, ktoré skreslia meranie. Pri voľnom páde samotného mobilu je táto pasáž výrazná a dosahuje takmer nulu. Tento faktor využijeme pre odlišenie týchto dvoch dejov, ale toto nemusí stačiť. Pri pohľade na ostatok krivky už veľa odlišností nájsť nemusíme. Tieto dva pády sú len demonštračné, ale môžu mať absolútne iný charakter za iných podmienok, teda viac píkov, vyššiu amplitúdu a iný pokles gravitačného zrýchlenia počas pádu.



Obr. 1.3 Zmena magnitúdy akcelerácie pri sadnutí na mäkký/tvrdý povrch

Asi najčastejšia aktivita, ktorá tiež môže pripomínať pád z pohľadu hodnôt zrýchlenia je sadnutie, pri ktorom veľkosť akcelerácie môže presiahnuť 3g, ale so zásadnými rozdielmi, ako môžeme vidieť na Obr 1.3. Pri mäkkom/pružnom povrchu zrýchlenie veľmi ťažko presahuje 2g, pričom pokles gravitačného zrýchlenia prakticky ani nemusí existovať, pretože ide o pomalý kontrolovaný pohyb osoby. Hlavne dochádza k plynulej zmene, ktorá sa po usadení môže ešte ďalej vlniť ako keď sa povrch pohybuje s osobou. Tento dej môžeme neskôr premietnuť do parametra v podobe strmosti alebo mieru zvlňenia okolo hlavného píku. Pri tvrdom povrchu nedochádza k plynulému

pohybu, ale napriek tomu strata gravitačného zrýchlenia a aj rozptylu hodnôt v okolí píku sú jednoznačne menšie ako pri páde, čo môže dopomôcť k budúcej detekcii [6].



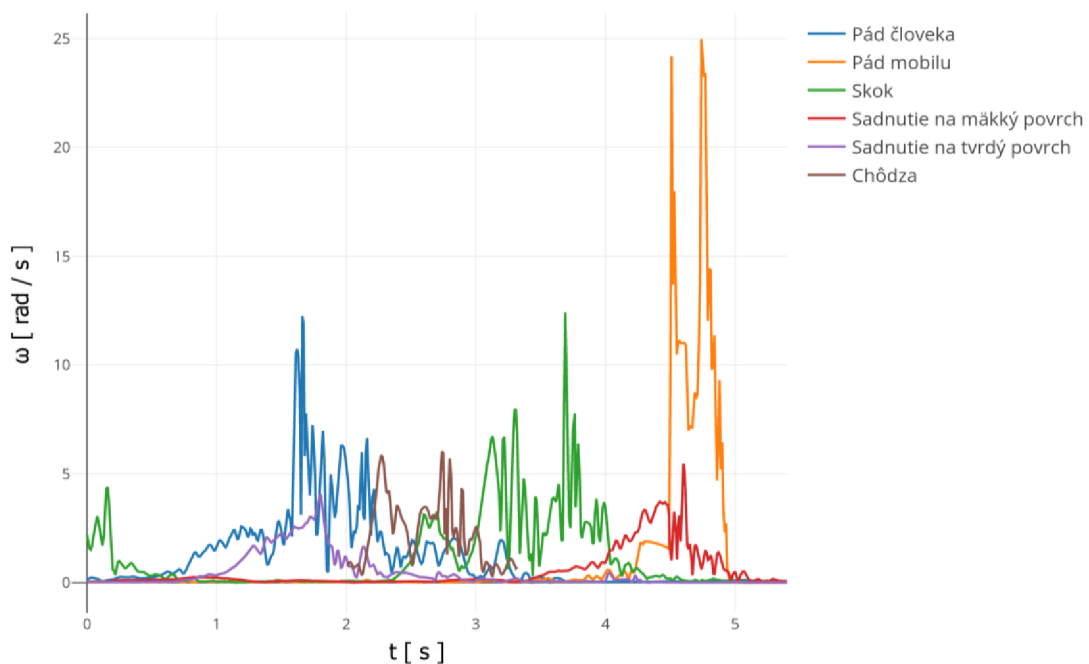
Obr. 1.4 Zmena magnitúdy akcelerácie skoku a chôdze

Ďalšie pohyby, ktoré by mohli napodobňovať priebeh pádu sú skok a samotná chôdza človeka, čo približuje zobrazenie na Obr 1.4. Pri chôdzi dochádza k vzniku periodického priebehu signálu. Mobil bol umiestnený na ľavej nohe, preto môžeme vidieť striedanie menšieho a vyššieho píku pri došliapnutí nohy na zem. Každému píku spôsobenému nárazom na zem predchádza rázny pokles gravitačného zrýchlenia pri uvoľnení nohy. Následne tiež sa objavuje ďalší sekundárny pík, ktorý naznačuje pohyb druhej nohy vpred. Pri tvorbe algoritmu na detekciu pádu je potrebné ošetriť vznik situácie, kedy periodický pohyb presahujúci istý prah zrýchlenia by pravidelne vyvolával klasifikáciu pohybu, hlavne z dôvodu šetrenia batérie v mobile. Pád môžeme chápať ako jednorazovú situáciu, nie ako periodicky opakujúci sa pohyb ako je chôdza.

Skok sa skladá z fáze náprahu nôh, ktorý môžeme vidieť ako samostatný pík v signáli a následného ich uvoľnenia, kedy sa človek dostáva do vzduchu a mobil nie je schopný zaznamenať zrýchlenie, kým nedôjde k dopadu na zem. Dopad je kontrolovaný, a z tohto dôvodu dochádza síce k ráznemu nárastu zrýchlenia, ale tento pík potom klesá takmer rovnomerne na rozdiel od pádu, kedy po píku nastáva až chaotické striedanie akcelerácie z rôznych strán.

1.3 Popis iných veličín

V reálnom svete môžu nastať desiatky rôznych situácií, ktoré môžu pripomínať pád. Pri klasifikácii týchto priebehov signálu budú tvoriť hlavnú väčšinu a akcelerometer nemusí stačiť. V nie ďalekej minulosti väčšina mobilov obsahovala len akcelerometer slúžiaci správnej orientácie zobrazovania plochy na obrazovke. Dnes výrobné metódy pokročili a výrobná cena iných senzorov klesla. Obsahujú ich zvyčajne zariadenia strednej-nížšej triedy, ale stávajú sa čím ďalej dostupnejšie aj pre lacnejšie zariadenia. Dáta z akcelerometru poskytnú najväčšiu oporu pri klasifikácii pohybu, ale ďalšie senzory nám ponúkajú ďalšie informácie o tom, ako sa mobil chová pri rôznych pohyboch a zároveň prispievajú k lepšej klasifikácii.



Obr. 1.5 Zmena uhlovej rýchlosti pri rôznych situáciách

V Android mobiloch sa bežne nachádza aj gyroskop, ktorý slúži na meranie uhlovej rýchlosti, ktorá je stále prítomná, keď sa mobil pohybuje. Na Obr 1.5 si môžeme všimnúť plynulosť zmeny na gyroskope počas takmer všetkých údajov až na pád mobilu. Dôvodom je, že bol pustený priamo bez nejakej rotácie, preto až pri dopade dochádza k vzniku píku, ktorý tvorí niekoľkonásobok ostatných meraní. Pre iné meranie by sme mohli predpokladať podobné správanie ako je vidno pri páde človeka, pri ktorom dochádza najskôr k jemnému navýšeniu. Keby sme pozreli na čas hlavného dopadu, tak je takmer identický s píkcom hodnôt z gyroskopu a dochádza k zotrvaniu nerovnovážneho stavu mobilu po dopade, kým sa telo neupokojí.

Pri sadnutí, skoku, chôdzi a pravdepodobne aj pri páde mobilu dochádza k postupnému poklesu uhlovej rýchlosti. Preto by sme sa pre budúcu kvalifikáciu nemuseli opierať len o výpočet nejakej formy prahu, ale zobrať do úvahy dĺžku intervalu,

kedy je uhlová rýchlosť prítomná nad istú hodnotu. Napriek tomu, že uhlová rýchlosť je stále prítomná pri pohybe, nemôžeme jej prisudzovať maximálnu dôležitosť, ale iba ju posudzovať ako dodatkovú informáciu, pretože pomerne vysoké zrýchlenie nemusí evokovať vysokú uhlovú rýchlosť. Môžeme si to všimnúť napríklad na zrýchlení pri sadnutí na tvrdý povrch. Pík pri zrýchlení dosahuje takmer 6g, ale uhlová rýchlosť je takmer najnižšia zo všetkých meraní. Pri páde človeka si ale môžeme byť istí, že uhlová rýchlosť bude vyššia a to z jednoduchého dôvodu. Dochádza k postupnej zmene polohy celého tela, čo má za následok neustály pohyb mobilu, ktorý nie je pevne pripevnený. Následne pri dopade má snahu zmeniť polohu pri presune síl, ale nemá priestor, čo má za následok neustále zmeny uhlovej rýchlosti [7].

Akcelerometer a gyroskop patria medzi pohybové senzory. V Androide nachádzame aj úplne inú triedu dostupných senzorov. Polohové senzory majú na starosti zisťovanie orientácie mobilu ako je napríklad kompas, ktorý za pomoci toku dát z akcelerometru vie zistiť orientáciu mobilu vzhľadom na severný pól. Podobná obdoba je tiež rotačný vektor, ktorý poskytuje informácie o rotácii zariadenia voči všetkým osám x, y, z [8].

1.4 Problematika strojového učenia

Metodiky detekcie pádu sú rôzne. Niektoré koncepty používajú iba sériu prahov, alebo nájdeme koncepty, ktoré využívajú modely umelej inteligencie na odlišenie druhu aktivity. Ako sme vyššie mohli vidieť, výstupy zo senzorov sú rôzne v závislosti od prebiehajúcej udalosti. Modely sú založené na matematických základoch, preto tieto výstupy zo senzorov je nutné kvantifikovať nejakým parametrom, ktorý je ľahko dosiahnuteľný a zároveň dostatočne reprezentuje obsiahnutú informáciu z daného výstupu senzoru a danej udalosti, ktorú chceme klasifikovať. Toto nazývame „feature extracting“ alebo extrakcia znakov. V tejto práci sme sa zaoberali so „supervised machine learning“, teda učením s učiteľom. Vypočítané parametre boli doplnené o cieľovú klasifikáciu, ktorá zjednoduší tvorbu modelu.

Pre splnenie úlohy máme na výber rôzne modely, ktoré ponúkajú rôzne výhody a nevýhody, napr.: support vector machine - SVM (metóda podporných vektorov), clustering (zhlukovanie), Naive Bayes, decision tree (rozhodovacie stromy), neurónové siete - NN a mnoho ďalších. Existuje vysoké počet prác, ktoré porovnávajú postupy prahovania a strojového učenia alebo ich kombináciu. Prahovanie je veľmi úspešné v kontrolovaných podmienkach, ale vyvoláva falošné detekcie pri pádom podobných situáciách. Či sa jedná o detekciu v laboratóriu alebo detekciu pomocou mobilu, pri prahovaní je v reálnom prostredí úspešnosť správnej detekcie 70 – 90 %, pričom pri niektorých existujúcich aplikáciách je falošná detekcia pri prahovaní vyvolaná aj 2 – 3 krát za deň. Klasifikácia pomocou strojového učenia bola úspešnejšia a dosahovala úspešnosti 80 – 95 % [9, 10].

Z dôvodu všeobecne vyššej úspešnosti algoritmov využívajúcich strojové učenie, Android aplikácia využíva jeden z týchto modelov. S tým prichádza niekoľko prekážok. Problémom niektorých postupov spomenutých v [9, 10] je nedostatok spracovaných dát, nazbierané dáta sú príliš idealistické, namerané v laboratóriu pomocou špecializovaných senzorov a hlavne nezahŕňajú obširnejšie testovanie počas bežnej dennej aktivity alebo vyžadujú špeciálne umiestnenie mobilu na hrudníku, trupe alebo na nohe. Preto je cieľom výber parametrov, ktoré sú nezávislé od uloženia mobilu, a zároveň sú dostatočne všeobecné pre rôzne druhy pádu. Zber dát obsahuje čo najvyšší počet ľudí pre vyššiu variabilitu priebehov signálu. Každý človek vykoná čo najviac meraní jednotlivých udalostí, čo pripomínajú pád a sú pádmi. Každý návrh detekcie je potom potrebné pretestovať na reálnych mobiloch rôznych výrobcov a u rôznych osôb, pre získanie najlepšieho prehľadu úspešnosti návrhu. V reálnom svete nie je nič ideálne, preto sa stretáme so šumom v dátach a ešte vyššou variabilitou dát, s ktorou sa model popasuje a zachová si svoju špecifickosť.

Strojové učenie je už implementované v niektorých mobilných aplikáciách napr. v [11]. V tejto práci použili dva prístupy pri voľbe atribútov, ktoré poslúžia na učenie modelu. Vytvorili sadu jednoduchých znakov ako priemer, rozptyl, rozpätie hodnôt, a následne potom tento počet obohacovali o ďalšie znaky až do 45 znakov. Zároveň pomocou nich otestovali SVM, zhukovanie a Naive Bayes klasifikátor. Všetky vynikali pri inej konštalácii znakov, ale každý si viedol pomerne dobre pri 10 znakoch. Hlavná pointa je v tom, že vysoký dôraz musí byť zvolený na výber atribútov, z ktorých sa daný model bude učiť.

Bola vypracovaná aj práca [12] zaoberajúca sa detekciou pádu a porovnaním jednotlivých algoritmov strojového učenia, kde na báze 50 dobrovoľníkov vytvorili dataset s 1000 záznamami. Porovnávala SVM, rozhodovací strom, Naive Bayes, NN a ďalšie 2 jednoduché modely, ktorými sa zapodievať nebudeme, kvôli ich pomerne jednoduchšej forme len pre demonštračné účely v práci. Pri SVM modeli je snahou učenia vytvoriť hranicu i v multidimenzionálnom priestore medzi jednotlivými kategóriami dát a vytvoriť pritom, čo najmenšiu chybovosť. SVM dokáže byť rýchla pre výpočet a dokáže sa prispôbiť i nelineárnym anomáliám v dátach. Do istej miery ide aj o nevýhodu, pretože má vyššiu šancu sa preučiť, a tým vypovedať istú chybovosť. Pri predikcii na základe SVM dostávame iba finálnu klasifikáciu bez vyjadrenej pravdepodobnosti daného rozhodnutia. Rozhodovací strom je zložený z rozhodovacích pravidiel, uzlov, ktoré sa ďalej vetvia do vetví až do finálnych listov, kde môžeme vidieť pravdepodobnosť rozhodnutia. Výhodu majú vo svojom samotnom princípe, a to že vykonávajú sériu jednoduchých rozhodnutí, ktoré si dokážeme predstaviť a vizualizovať, pričom sa vedú vysporiadať i s nečíselnou podobou údajov. Problém nastáva pri veľkých dátach, pretože náročnosť a komplexnosť stromu narastá exponenciálne. Naive Bayes kvalifikátor je založený na Bayesovom teoréme, kde je každý atribút na začiatku výpočtu považovaný ako nezávislý od ostatných. Vypočítava sa podmienená pravdepodobnosť,

čo umožňuje rýchly výpočet i s niekoľkými atribútmi, ale v reálnom svete sú tieto parametre vždy v nejakom zmysle prepojené a nie nezávislé, a preto sa vo všeobecnosti jeho výsledky považujú skôr len ako odhad, než reálna predpoveď.

Neurónové siete sú v poslednej dobe hlavnou témou pri umelej inteligencii, pretože poskytujú vysokú flexibilitu pri detekcii obrazu, reči a iných javov. Ide o zosieťované neuróny, ktorých váhy a prahy aktivácie sú upravované optimalizačným algoritmom rôzneho rázu. Ich interpretácia je náročná na rozdiel od napr. rozhodovacích stromov, preto sa maximálne reprezentujú vrstvy, z ktorých je sieť skonštruovaná. Trénovanie siete je výpočtovo náročné a ich upravovanie pre lepšie výsledky ešte viac, lebo pri pohľade na model nevieme určiť zdroj problémov pri učení. Preto je nutné neurónové siete zásobovať veľkým počtom údajov, pričom by sa mali správne kvantifikovať a reprezentovať cieľ klasifikácie. Po tréovaní je výpočet neurónovej siete jednoduchý a rýchly.

Tab. 1.1 Porovnanie presnosti modelov detekcie pádu prebrané z [12]

Algoritmus	Správne klasifikácie [%]	Priemer	Smerodajná odchýlka σ	Čas stavby modelu [s]
Viacvrstvý perceptron	90,15	89,07	0,879	28,35
Naive Bayes	88,57	87,11	1,151	<1
SVM	86,68	86,18	0,388	3,52
Rozhodovací strom	84,28	83,68	0,572	4,5

Ako môžeme vidieť na Tab 1.1, zostavenie neurónovej siete trvá mimoriadne dlho na rozdiel od iných metód, ale vykazuje najvyššiu presnosť, najmenšiu chybovosť zároveň, a to i napriek vyššej variabilite výsledkov. Na dosiahnutie takej úspešnosti bolo potrebných až 45 neurónov v počítačovom prostredí [12]. Dnes algoritmy, hardware, programovacie knižnice pokročili o veľký kus v umelej inteligencii, v poslednom roku hlavne pri adaptácii na smartphony, a preto dostanú hlavnú pozornosť pri stavbe modelu pre detekciu pádu v ďalšej kapitole.

1.5 Popis a detekcia autohavárie

Autonehodu definujeme ako zrážku vozidla s ďalším vozidlom alebo inou prekážkou v podobe statického objektu ako je budova, strom, lampa alebo v podobe človeka, zvierat'a, trosiek na ceste a iné. V tejto práci sa venujeme tvorbe mobilnej aplikácie, ktorá vie identifikovať prudké zastavenie, ktoré je charakteristické len pre kolíziu s veľkými objektmi. Bez priamej interakcie s autom je nemožné inak rozpoznať zrážku s menšou divou zverou alebo chodcom, pretože daná interakcia nie je prenesená v takej sile na to, aby evokovala detekciu. Na rozdiel od detekcie pádu, táto detekcia nevyužíva strojové učenie, a to z jednoduchého dôvodu – takéto testovanie by bolo dosť nákladné a zároveň by šlo o príliš komplexnú záležitosť, ktorá by si vyžadovala nedostupné zdroje. Preto detekcia je založená na splnení istých podmienok v istom časovom slede, a prahovaním.

Pri náraze auta pri vysokých rýchlostiach dochádza k drastickému spomaleniu aj z 130 km/h na nulu, pričom pripútaný pasažier vo vozidle môže absorbovať zrýchlenie presahujúce aj 60 g v priebehu 0,1 – 0,5 s [13]. Pre takomto preťaženi dochádza k niekoľkonásobným poraneniam, pričom silové pôsobenie sa najviac vzťahuje na hlavu, čo zapríčiňuje ťažký otras mozgu, vnútorné krvácanie, ktoré často vedie k smrti. Kolízia sa nemusí udiať vo vysokých rýchlostiach, pričom jeden z účastníkov môže stáť aj na mieste. Pri náhlej zmene rýchlosti len o 8 km/h v priebehu 0,1 s medzi dvomi vozidlami dochádza k pôsobeniu zrýchlenia s hodnotou 10 g na oblasť hlavy, 8 g na hrudníku a v priemere 4 g na celé vozidlo [14]. Podobné koncepty tiež využívajú túto vlastnosť v zmysle, že pre detekciu autohavárie je nutná prítomnosť zrýchlenia s hodnotou vyššou ako 4 g [15, 16]. Táto hodnota výrazne prevyšuje hodnoty, s ktorými sa mobil môže stretnúť v rámci jeho bežného používania, preto potláčanie nežiadúcich aktivít nie je potrebné.

Problém sa ale nedá vyriešiť len na báze akcelerometra. Pre správnu detekciu je nutné snímať aj pohyb človeka, respektíve vozidla, v ktorom sedí. Prvotným cieľom sa stane najskôr detekcia toho, že sa človek nachádza vo vozidle. Motorovým vozidlom vieme jednoducho dosiahnuť rýchlosti, ktoré človek nedosiahne a tieto rýchlosti vieme získať prostredníctvom GPS v mobile, čo je jednoznačným indikátorom, že sa človek nachádza v nejakom druhu vozidla. Vrcholoví športovci sú schopní pri maratone bežať v priemere aj 20 km/h [17], preto keď rýchlosť vozidla bude podľa GPS vyššia ako 20 km/h, môžeme takmer s istotou určiť, že človek sedí v nejakom vozidle a pri dopravnej nehode môžeme očakávať presah preťaženia nad 4 g. Preto sa GPS stáva ďalším parametrom, ktorý rozhoduje o autonehode. Pri získavaní polohy v Androide dochádza k vytvoreniu požiadavky na systém na pravidelnú aktualizáciu polohy pre danú aplikáciu v istých časových intervaloch alebo za istú prejdenú vzdialenosť od poslednej aktualizácie [8], čo je ešte podrobnejšie rozobrané v ďalšej kapitole. Tým, že nedostaneme aktualizáciu po istom čase znamená, že nedošlo k zmene polohy, pričom keby sme stratili signál, tiež by sme dostali správu o nedostupnosti lokalizácie. Teda keď

v tomto okamihu registrujeme vysoké preťaženie a GPS nevykazuje výraznú zmenu polohy, môžeme registrovať autonehodu.

Komplikácie by mohli nastať, keď sa rýchlosť vozidla sa občas pohybuje pod 20 km/h. Tento problém ešte viac komplikuje fakt, že GPS sa stáva pri malých rýchlostiach nepresná, pretože presnosť určenia polohy sa pohybuje v priemere od 5 – 10 m okolo miesta merania na holej planine [18]. Pri kombinácii nepresností spôsobených okolím (tienenie signálu objektami, odrazom od budov, idealizácia zemského povrchu,...) a nastavení satelitov sa pri výpočte rýchlosti v GPS senzore skreslia hodnoty o aktuálnej pozícii a môžu vyjsť skreslené hodnoty. V horšom prípade nemusíme dostávať žiadne aktualizácie polohy, pretože žiaden signál nemusí byť dostupný. Riešenia tohto problému sú rôzneho charakteru. V jednom koncepte je tento problém je riešený výpočtom smerodajnej odchýlky rýchlosti, ktorá sa podľa testov odlišuje od pri jazde autom pri nízkych rýchlostiach pod 24 km/h (1,056 – 1,88 pri chôdzi, okolo 2 pri behu, 2,9 – 7,7 pri jazde autom) [15]. Prekážku je možno vidieť v tom, že nepredpokladá neprítomnosť GPS signálu. Pri jazde autom môžeme predpokladať vo väčšine prípadov prítomnosť GPS signálu, ale nie vždy.

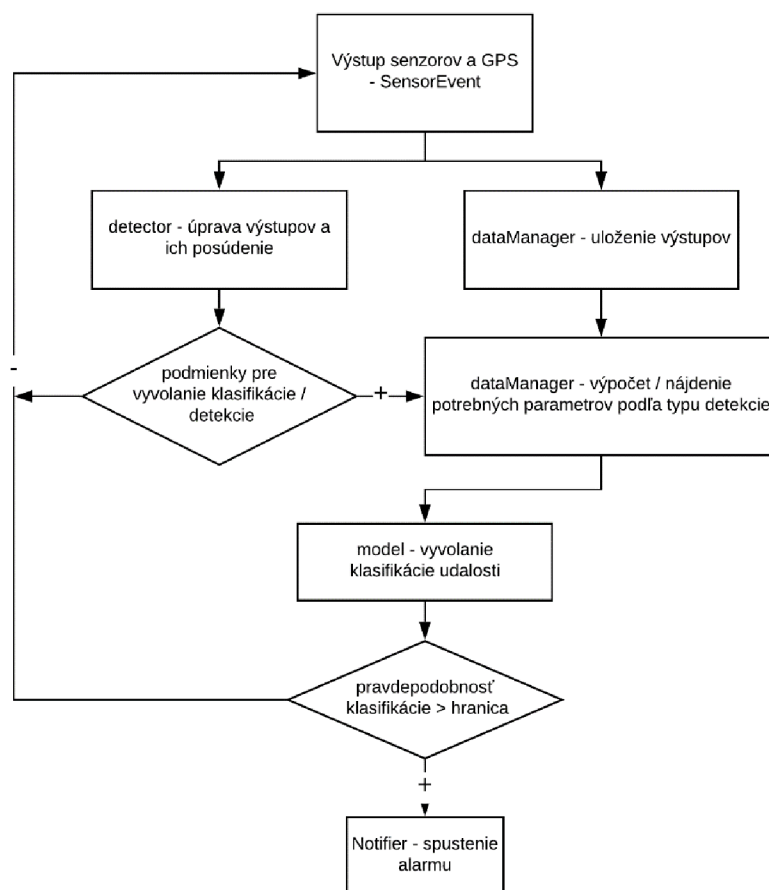
Vhodnejšia by bola implementácia pedometra/krokomera, ktorý je prítomný v Android mobiloch buď so špecifickým senzorom alebo ide o programové ošetrenie na základe akcelerometru [8]. Pomocou tohto senzoru, by sme mohli odlišiť chôdzu a sedenie vo vozidle, a tým posunúť hranicu pre jazdu autom aj na 10 km/h. Teda pri absencii detekcii z krokomera a rýchlosti nad istý prah by aplikácia zmenila svoj stav na detekciu autohavárie a naopak, pri detekcii chôdze a absencii vysokej rýchlosti by zmenila svoj stav na detekciu pádu. Falošné detekcie môžu byť tvorené pádom mobilu v aute, ale ako sme mohli vidieť pri detekcii pádu, pri názornom páde zo 40 cm na tvrdú zem, akcelerácia sa ledva vyšplhala nad hodnotu 4 g. V aute mobil padá veľmi komplikovanou trajektóriou po palubnej doske, mäkkých poťahoch alebo pogumovaných rohožiach, ktoré utlmia dopad. Senzor na kroky by tiež mohol reagovať na prácu nôh pri preradzovaní alebo brzdení, pokiaľ by bol umiestnený vo vrecku. Túto anomáliu, i pri jej absencii, je možné potlačiť podmienkou pre istý počet krokov v istom časovom intervale.

Detekciu autonehody pomocou GPS a akcelerometru môžeme ešte podporiť menej spoľahlivým, ale stále podporujúcim parametrom, a to intenzitou zvuku pri autonehode. Pri dopravnej nehode dochádza k mohutnému hluku, ktorý dosahuje hodnôt aj nad 140 dB [16, 19]. Problémom je, že mikrofóny v mobiloch sú stavané na snímanie ľudskej reči, teda reagujú hlavne na úrovni 60 dB a hlasitosť nad 130 dB pri dlhšom trvaní by ich mohla poškodiť alebo až nenávratne zničiť [20], preto tento údaj musí byť váhovo voči ostatným dvom znížený a má skôr podpornú funkciu.

2. NÁVRH MODELU PRE ANDROID

V tejto kapitole analyzujeme základný priebeh algoritmu, ako bude detekcia fungovať, a stavbu modelu pre klasifikáciu. Preto zatiaľ budeme brať za samozrejmé, že existujú nejaké výstupy zo senzorov mobilu a GPS, ktoré sú ukladané do dočasnej pamäte, posielané na overenie prítomnosti podozrivej aktivity a pripravené na využitie pri detekcii.

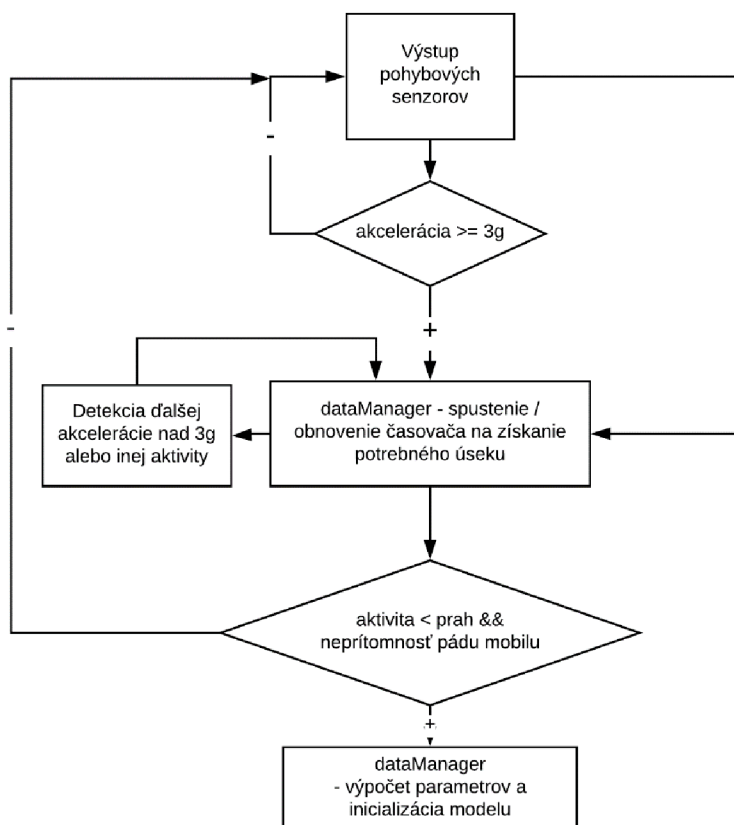
Pred vykonaním sekvencie rozhodnutí pre klasifikáciu aktuálnej činnosti, je nutné rozhodnúť, či sa človek práve nachádza v aute alebo mimo auta. Android má v sebe zabudovaný systém rozpoznania aktuálnej aktivity, pričom vie rozpoznať či sa mobil nachádza vo vozidle, bicykli alebo či človek beží, chodí, alebo len mobil zmenil svoju polohu ku smeru gravitácie, či leží stacionárne na nejakej ploche. Preto pri počiatku spustenia aplikácie, ak je povolená automatická detekcia, dochádza k registrácii aplikácie k odberu informácií o aktivite. Ak dôjde k zmene pohybového stavu, aplikácia dostane túto informáciu a podnikne potrebné kroky na zmenu detektora [8].



Obr. 2.1 Diagram všeobecného algoritmu detekcie

Princíp algoritmu je ukázaný v diagrame na Obr. 2.1. Vystupujú v ňom už objekty reprezentujúce jednotlivé celky programu. Celý algoritmus obsahuje jeden objekt v podobe „DataManager“, teda objekt, do ktorého sa budú dočasne ukladať výstupy zo senzorov a GPS. Zároveň pri vyvolaní podozrivej udalosti bude pokračovať v zázname obmedzenú časovú dobu, kým nedôjde k narušeniu záznamu, napr. človek ďalej pokračuje vo svojej činnosti a klasifikácia nie je nutná alebo nenastala žiadna aktivita a záznam je poslaný na ďalšie posúdenie. V prípade pádu môžeme overiť prítomnosť beztiažového stavu a neprítomnosť inej aktivity človeka. Údaje zároveň budú posielané do objektu „Detector“, ktorý bude detekovať podozrivú aktivitu, pričom vyvolá potrebné akcie na vykonanie prípadného merania alebo navráti ostatné objekty do pôvodného stavu. V priebehu detekcie bude aplikácia kontrolovať aj stav človeka, či je v aute alebo mimo auta. Pre klasifikáciu bude vyvolaný znova objekt „detector“, ktorý bude slúžiť na klasifikáciu udalosti. Pri pozitívnej klasifikácii pre pád alebo autohaváriu, dochádza k vyvolaniu alarmu a odoslania informácii cez objekt „Notifier“.

2.1 Detekcia pádu



Obr. 2.2 Diagram algoritmu detekcie pádu

Algoritmus detekcie pádu je zobrazený na Obr. 2.2. Základným predpokladom pre detekovanie pádu je rázne zvýšenie akcelerácie nad 3g, ktoré naznačuje pád alebo udalosť jemu podobnú. Z výstupu senzorov je vypočítaná amplitúda akcelerácie a položená podmienka, či je vyššia akcelerácia ako 3 g (Táto hodnota bude ešte upravená v závislosti od vytvoreného datasetu). Ak sa táto detekcia potvrdí, nahrávanie ešte pokračuje istú časovú dobu na nasnímanie celej udalosti. Následne po udalosti dôjde k snímaniu úseku po udalosti, ktorá je kľúčová pre zvýšenie presnosti algoritmu.

Keďže sa snažíme pomôcť ľuďom, ktorí sa môžu ocitnúť bez blízkej pomoci, je dôležité klasifikovať, či sa človek po páde snaží postaviť alebo predsa len vykonáva inú aktivitu. Keď sa po nadetekovanej udalosti nachádza aktivita, nie je nutné vyvolávať klasifikáciu pomocou modelu, čím sa vyhneme falošnej detekcii. Pri nečinnom ležaní, nedochádza k zmenám na akcelerometri, teda až potom môžeme využiť model na klasifikáciu predošlej udalosti a podľa výstupnej pravdepodobnosti môžeme určiť, či mobil vyvolá nejakú akciu na upozornenie alebo nie.

Zároveň ako ďalšiu podmienku môžeme pridať prítomnosť zmenšenia akcelerácie pod 1 g, pretože pre obdobie voľného pádu tiež je nutná charakteristika pádu. Nejde o povinný parameter pri páde, z dôvodu interferencií z okolia, preto sa do klasifikácie nepridáva ako záväzný prvok. Keďže mobil, keď spadne jeho užívateľovi, sníma takmer nulovú akceleráciu v rámci desiatín sekúnd ako je zobrazené na Obr. 1.2, tak môžeme vyhľadať tento priebeh a indikáciu pádu zamietnuť. Preto keď priebeh obsahuje časť menšiu ako 2 m/s^2 , pričom je dlhá najmenej 50 ms, tak sa indikácia zamieta.

Tento postup môžeme chápať ako filter pred klasifikáciou modelom, ktorý zabezpečuje zredukovanie množstva vyvolaní modelov v systéme. Spomenuté podmienky sú kľúčové z pohľadu smartphonu, pretože nemáme k dispozícii neobmedzené množstvo kapacity batérie. Spotreba pri využití zopár podmienok a časovačov v Androide bude menšia ako pri inicializácii a klasifikácii pomocou samotného modelu i napriek vysokej miere optimalizácie, takže vo výsledku dostaneme i dlhšiu životnosť batérie.

2.2 Výpočet parametrov pre detekciu pádu

2.2.1 Parametre špecifické pre akceleráciu

Keď sa potvrdí amplitúda akcelerácie nad 3g, prítomnosť zložky voľného pádu a žiadna aktivita po detekcii danej udalosti, vyvolávame výpočet parametrov zo zaznamenaného priebehu a klasifikáciu pomocou modelu. Ak predošlé podmienky nie sú splnené alebo model vyhodnotil priebeh ako udalosť podobnú pádu, tak dôjde k návratu do pôvodného stavu. K dispozícii máme viac senzorov, teda môžeme vytvoriť väčšiu sadu parametrov popisujúcich danú udalosť. Najviac zastúpené sú parametre vypočítané z akcelerometru, pretože tieto údaje sú najviac špecifické pri popise pádu.

Výpočet parametrov je prispôsobený tak, aby v ňom bolo zaznamenané obdobie voľného pádu, kontaktu a následne aby sa okno ukončilo po odznení hlavnej aktivity. V [21] odporúčajú pre výpočet zobrať 640 ms pred a 540 ms po hlavnom píku, s čím sú v zhode s prácou [5]. V [6] zase odporúčajú prahovať okno na hodnote akcelerácie 1,5g pred a po hlavnom píku za podmienky prítomnosti zložky voľného pádu. Pri nasledujúcich parametroch je volený podľa potreby jeden alebo druhý spôsob.

Základný rozdiel medzi priebehmi pádu a inými udalosťami je strmosť priebehu akcelerácie. Pri páde mobilu nachádzame jeden okamžitý vysoký pík, zatiaľ čo pri sede ide o dlhú vlnu s postupným stúpaním. Tento poznatok využili aj v [21], kde ako hlavný parameter vypočítali **priemernú odchýlku amplitúdy akcelerácie**, po anglicky „Average acceleration magnitude variation“ so skratkou **AAMV**.

$$AAMV = \frac{1}{N_w} \sum_{i \in W} |acc_{i+1} - acc_i| \quad (1)$$

V rovnici vystupuje acc_i ako vzorka celkovej amplitúdy akcelerácie, N_w ako počet vzoriek v okne a W ako definované okno, z ktorého prebieha výpočet. Ide o sumu odčítaných vzoriek v naslednom poradí, ktorá definuje mieru stúpania daného signálu. Zároveň je normalizovaná počtom vzoriek v danom okne, teda je vhodný na porovnanie rôznych signálov. Tento postup je ekvivalentný výpočtu Waveletovou dĺžkou, ktorý aplikujeme na všetky signály.

Ako sme už spomínali, obdobie voľného pádu je charakteristické pri páde, pričom pri pohľade na priebeh pádu mobilu a človeka je zásadný rozdiel. Pri páde mobilu dosahujeme takmer nulových hodnôt akcelerácie, pričom pri páde človeka mobil nepadá priamo na zem, ale vplývajú na neho aj iné elementy, preto je táto hodnota vyššia a obsahuje viac interferencií. Preto jednoduchým priemerom hodnôt v danom intervale nám posluží ako ďalší parameter a môžeme ho nazvať **priemerná hodnota amplitúdy akcelerácie voľného pádu**, alebo FFAAM z anglického prekladu [21]. Tento parameter je vypočítaný z úseku nachádzajúceho sa pred hlavným píkom, zvyčajne do 200 ms [6], s hodnotami pod 0,8g. Pri náhodnej absencii tohto úseku, dochádza nastaveniu tohto parametru na 1g.

Pre spresnenie popisu deja, musíme zaviesť aj **celkovú aktivitu akcelerácie, CAA**, na odlíšenie jednotlivých impulzov v akcelerácii a postupného pádu. K tomuto si zavedieme presne definované okno 500 ms naľavo a napravo od dominantného píku. Z tohto okna následne môžeme viacerými spôsobmi definovať aktivitu akcelerácie. Môžeme použiť prostý rozptyl, ale v rámci ľahšej normalizácie dáť môžeme dať do pomeru počet vzoriek akcelerácie s amplitúdou nad 1,5g a všetky ostatné v zvolenom okne. Tento pomer by reprezentoval pretrvávanie zmeny akcelerácie pri postupnom páde, zatiaľ čo pri páde mobilu by tento pomer bol výrazne menší.

Ďalším parametrom, ktorým by sme mohli vylúčiť pád, je **změna uhla**, change in angle **CA**. Dokážeme odfiltrovať všetky aktivity a získať len čisté pôsobenie zeme na mobil a vektor pôsobenia. Android mobilny zvyčajne obsahujú aj dedikovaný senzor na záznam pôsobenia gravitácie. Z tohto môžeme porovnať dominantnú zložku gravitácie pred a po danej udalosti. Pri páde môžeme predpokladať změnu náklonu o 70 až 90° a následne dôjde k změne dominantnej zložky gravitácie, teda změne smerovania. Tento parameter testovali v zjednodušenej forme v [5], kde do úvahy zobrali len 2 osi mobilu, pretože translačný pohyb po osi y môžeme zanedbať pri detekcii pádu.

$$CA = \frac{1}{N_w} \sum_{i \in W} \sqrt{a_{x_i}^2 - a_{z_i}^2} \quad (2)$$

V rovnici (2) sa teda vyskytuje a , ktoré reprezentuje zrýchlenie na danej osi. V [7] mali tiež podobnú myšlienku, ale vybrali iný postup.

$$CA = \cos^{-1} \left(\frac{\overline{a_a} \cdot \overline{a_b}}{\overline{acc_a} \cdot \overline{acc_b}} \right) \quad (3)$$

Vytvorili dva vektory a_a a a_b , z ktorých jeden bol tvorený priemerom hodnôt vektorov v intervale jednej sekundy, sekundu pred pádom a druhý vektor sekundu po páde s rovnakým intervalom. Následne bol vypočítaný uhol týchto dvoch vektorov z podielu ich skalárneho súčinu a súčinu ich amplitúd, ktoré sú reprezentované symbolom acc .

V [5, 7] tiež riešili priebežnú změnu smerovania vektoru zrýchlenia, pričom chceli využiť všetky osi a vytvoriť tým všeobecnejší parameter. Toto snaženie odhalilo **uhlovú odchýlku** alebo angle deviation **AD**. Pri páde sa táto hodnota pohybuje okolo 30°, zatiaľ čo pri postavení zo stoličky len maximálne 7°.

$$AD = \frac{1}{N_w} \sum_{i \in W} \cos^{-1} \left(\frac{a_n \cdot a_{n+1}}{\overline{acc_n} \cdot \overline{acc_{n+1}}} \right) \quad (4)$$

V (4) ide o skalárny súčin dvoch nasledujúcich vektorov, so skratkou a , v podiele s ich amplitúdami, so skratkou acc . Vypočítaná hodnota sa následne vloží do inverznej funkcie kosínusu, prípadne sa prepočíta na stupne.

Jedným z parametrov, ktorý sa vyskytuje v rámci spracovania reči alebo iného audia, je prechod cez nulovú hodnotu. Väčšina mobilov má akcelerometer bez filtrácie gravitácie. Preto naša implementácia obsahuje variantu, kedy magnitúda akcelerácie prechádza cez hodnotu 1g. Ďalšie parametre by sme mohli odvodiť od času intervalu istých stavov pádu, ako napr. dĺžka stavu voľného pádu alebo dĺžka intervalu od 1,5g do 1,5g v okolí hlavného píku. Podobne môžeme vybrať z daného úseku minimum a maximum, alebo ich reprezentovať ich rozdielom. V [6] ešte uvádzajú „krokový index“, teda snahu nájsť vzor predchádzajúceho kroku, t.j. detekovaná udalosť by bola len ukončenie chôdze. Tento aspekt potlačujeme zisťovaním aktivity nasmerovania

gravitácie v predošlom parametri. Pri umiestnení mobilu vo vrecku a bežnej chôdzi nedochádza k zmene smerovania vektoru zrýchlenia gravitácie. I pri zmene tohto vektoru by ostatné parametre mali byť dostatočné, aby nedošlo k vyvolaniu falošnej detekcie.

2.2.2 Všeobecné parametre

K parametrom uvedeným vyššie pribudnú ešte ďalšie, ktoré uplatníme nie len pre akceleráciu, ale aj pre hodnoty z gyroskopu a pre uhlové zrýchlenie vypočítané z hodnôt gyroskopu, ktorú pre presné určenie pádu využili v [22]. Na vybranú časť pádu aplikujeme rôzne štatistické parametre, ako je napr. priemer, smerodajná odchýlka, rozptyl, strmosť, špicatosť, aproximáciu entropie, minimum, maximum, rozdiel maxima a minima, a podobne.

K týmto priebehom môžeme pristupovať ako ku každému inému signálu. Preto môžeme využiť aj niektoré znalosti z iných druhov spracovania. K získaniu ďalších parametrov môžeme použiť Hjortove parametre [23], medzi ktoré patrí aktivita, mobilita, zložitosť. Obdobne môžeme definovať parametre pre energiu a výkon. Ďalej môžeme použiť stredný výkon signálu, ktorý je normalizovaný počtom vzoriek. Síce netradične, ale pokúsime sa použiť Teager-Kaiserov energetický operátor [24], ktorý zosumujeme a vydělíme počtom použitých vzoriek z dôvodu zovšeobecnenia hodnoty pre rôzne druhy mobilov.

Pre naše testovanie môžeme nájsť veľa odlišných parametrov. Problém ale zostáva, že sme v prostredí mobilu a nemáme neobmedzenú kapacitu batérie. Snahou je maximalizovať presnosť detekcie, ale musíme sa taktiež snažiť o minimalizáciu počtu krokov pre dosiahnutie správnej detekcie. Všetky parametre sú spracované do potrebných vizualizácií, distribúcií a zredukované na minimálny potrebný počet v nasledujúcej kapitole.

2.3 Trénovanie neurónovej siete

Na Google I/O 2017, konferencii určenej pre prezentáciu nových developerských produktov, bola po prvýkrát odprezentovaná oficiálna adaptácia modelov strojového učenia pre Android v podobe Tensorflow Lite. Tensorflow je open source knižnica, tvorená hlavne inžiniermi z Google Brain team, ktorá slúži na stavbu modelov strojového učenia, ktorá vie využiť pri učení aj hardwareovú akceleráciu na urýchlenie procesu učenia, zároveň aj pri aktuálnom využívaní modelu. Tensorflow je najviac rozvinutý pre využitie Pythonu, Javascriptu, C jazykov, ale aktívne sa vyvíjajú aj adaptácie pre Java, Go, Swift a iné jazyky [25]. Modely Tensorflowu nájdete takmer v každej populárnej webovej službe, serveroch a už vďaka Google aj v Android aplikáciách ako je Gmail, Google Assist, Photos a iné. Svojomu rozšíreniu vďaka hlavne svojej flexibilitě pri adaptácii do systémov s rôznymi jazykmi, teda model nie je problém v prípade potreby adaptovať na iné prostredie ako je napr. iOS od Apple.

Ako je uvedené v [26], písanie kódu priamo pomocou Tensorflow knižnice je veľmi náchylné na vytvorenie veľkého množstva chýb v modeli, ktoré človek nemusí po dlhšej dobe odhaliť, a navyše je potrebné stráviť nad tým veľa času. Preto existujú ďalšie knižnice, ktoré využívajú jadro Tensorflow, ale ich rozhranie je značne jednoduchšie. Cieľom je odhaliť tú najlepšiu konfiguráciu modelu pre riešenie nášho problému a prípadne až potom sa zapodievať jej technickou optimalizáciou. Medzi najpopulárnejšie knižnice využívajúce Tensorflow patrí Keras [27], ktorý využíva ako základ tiež Python. Množstvo vývojárov ho zároveň aj odporúča pre prvotnú tvorbu modelu, pretože Keras ponúka vysokú flexibilitu a modularitu v podobe predom vytvorených blokov, ktoré programátor môže pospájať dohromady, zvolí ich parametre optimalizácie a váhovania, a zvolí druh tréningu. Modely vytvorené Kerasom sú zároveň plne kompatibilné s pôvodným kódom z Tensorflow, čo umožňuje ich konverziu na Tensorflow lite verziu, použiteľnú pre Android a iOS.

Stále ale platí, že je nutné veľkosť modelu minimalizovať, preto boli tréningové pomery malé počty neurónov. V Keras existujú dva typy návrhov siete a to sekvenčný a model API. Sekvenčný slúži na poskladanie jednotlivých druhov vrstiev, zatiaľ čo model poskytuje pokročilejšie konfigurácie, vlastné vetvenie sietí a spájanie odlišných modelov do jedného [27]. V našom prípade si vystačíme s klasickým sekvenčným modelom, ktorý bude obsahovať vstupnú a výstupnú aktivačnú vrstvu s jednou až tromi skrytými vrstvami. Pomocou Kerasu je tento cieľ veľmi ľahko dosiahnuteľný, teda môžeme interpretovať neurónovú sieť s jednou skrytou vrstvou. Stačí vytvoriť inštanciu sekvenčného modelu, do ktorého pridáme jednotlivé vrstvy, ktoré sú navzájom plne prepojené. V Keras tieto vrstvy reprezentuje objekt „Dense()“, v ktorom si špecifikujeme koľko neurónov bude obsahovať a zároveň pri prvej vrstve je potrebné definovať aj počet vstupov. Obdobne pridáme aj skrytú vrstvu aj výstupnú vrstvu. Pri aktivačnej vrstve sú možné dva spôsoby, kedy môžeme aplikovať aktivačnú vrstvu priamo na Dense alebo pridáme ďalšiu samostatnú vrstvu s „Activation()“ objektom, pričom obidva spôsoby sú ekvivalentné.

Už je len potrebné zvoliť optimalizačný algoritmus, spôsob hodnotenia a napasovať potrebné údaje, na ktorých sa má sieť tréningovať. Ako je vidieť, celú sieť je možné vyskladať z jednotlivých objektov, a to umožňuje zostrojenie skriptu, ktorý pretestuje rôzne konfigurácie siete, teda môžeme otestovať rôzne počty neurónov a vrstiev s rôznymi optimalizačnými algoritmami.

Sieť bude obsahovať rovnaký počet vstupných neurónov ako je počet parametrov a 1 výstupný neurón, ktorý bude vypovedať o šanci detekcie pádu. Vstupné neuróny budú mať aktivačnú funkciu usmernenej lineárnej jednotky, alebo „ReLU“, podľa odporúčania v dokumentácii a výstupný bude mať funkciu sigmoidy [27]. Od tohto bodu, už je ťažké definovať smer, aká sieť by bola vhodná pre riešenie nášho problému a vyžaduje si to obširné testovanie na nájdenie najoptimálnejšej konfigurácie neurónovej siete. Preto bol vytvorený skript, ktorý vytvorí rôzne variácie siete za pomoci rôznych prostriedkov

a porovná ich úspešnosť. Najjednoduchším nápadom je meniť počty neurónov a počty skrytých vrstiev. Ďalej môžeme meniť druh optimalizačného algoritmu, zároveň meniť ich rýchlosť učenia a dĺžku učenia v epochách.

V rámci optimalizačných algoritmov použijeme dva populárne algoritmy. To sú SGD, alebo stochastický gradientný spád, a „Adam“, alebo adaptívny predpoklad momentu. SGD patrí medzi staršie algoritmy, ktoré vychádzajú z výpočtu gradientu chybovej funkcie popisujúcej neurónovú sieť, ktorá by mala viesť k čo najvyššiemu minimu. Hlavnou nevýhodou tohto postupu je jeho pomalosť, pretože pre vypočítanie predpokladu gradientu je použitá len frakcia dostupných údajov, čo môže viesť aj k prípadom, kedy sa sieť nemusí učiť a sekne sa v jednom stave alebo sa ešte viac vzdiali ideálnemu riešeniu, pretože pri použití len frakcie údajov môže smerovať odhad zlým smerom. Tomuto predchádzajú ďalšie variácie tohto algoritmu, kedy sú zohľadnené aj predošlé výpočty a ich momenty alebo sú zavádzané ďalšie aproximácie, ktoré posúvajú stav siete k lepšej konfigurácii. Ďalším problémom je aj, že všetky rýchlosti učenia pre všetky parametre ostávajú počas učenia vždy konštantné, čo vedie k ďalším zbytočným krokom v optimalizácii.

Adam má podobné riešenie. Na rozdiel od SGD, ale dochádza k zmene jednotlivých rýchlosti učenia pri všetkých parametroch, teda keď parameter nemal až tak vysoký vplyv na tvorbu chyby, tak jeho rýchlosť učenia môže byť znížená. Adam vypočítava momenty jednotlivých gradientov pre každý parameter, čo umožňuje ich osobnú úpravu. Zároveň Adam vznikol kombináciou ďalších algoritmov ako je AdaGrad a RMSProp, ktoré čiastočne implementujú túto funkciu. Vďaka tomu Adam dokáže oveľa rýchlejšie nájsť cestu k optimálnejšiemu riešeniu, ale nemusí dospieť k tomu najlepšiemu z dôvodu poklesu miery učenia.

Preto napríklad v [28] navrhujú používať obidva algoritmy zároveň. Adam je použitý na začiatku pre nájdenie približnej cesty k riešeniu a SGD na zdokonalenie výsledku. Pre vyššiu variáciu modelov v našom testovaní budeme používať aj rôzne konštanty učenia, pretože základné nastavenie nám nemusí vyhovovať. Algoritmy ponúkajú aj ďalšie parametre, ale ich zmeny by vyvolali veľké nekonzistencie [27].

Vyššie spomenuté parametre stačí už len precykliť do jednotlivých modelov, podľa potreby zvoliť počet epoch a trénovať ich. Pre dohľad nad ich trénovaním sme použili pomôcku z Tensorflow, Tensorboard, ktorá zobrazuje základnú architektúru neurónovej siete ale aj priebeh učenia a porovnanie odlišných sietí. Podľa okolností, môžeme zmeniť metodiku výpočtu parametrov, pretože i s najideálnejšími nastaveniami siete, dáta vytvoria danú sieť, ktorých zber a spracovanie preberieme v ďalšej kapitole.

2.4 Detekcia autohavárie

Algoritmus detekcie autohavárie využíva akcelerometer a GPS, ktorý sa nastaví na vyššiu frekvenciu aktualizácii. Dôvodom je pravidelné zisťovanie rýchlosti, ktorá je jedna hlavných parametrov tejto detekcie. Autonehodu nemôžeme nijakým spôsobom otestovať v realite očividne, ale existujú záznamy z akcelerometru zo simulácii autonehody.

Napríklad pri prehlíadaní záznamov zo Štátnej administrácie pre bezpečnosť na cestách v USA, je možné nájsť záznamy zo simulácii autonehôd. Nárazy boli monitorované pomocou vysokofrekvenčných akcelerometrov s rozsahom desiatok násobkov gravitačného zrýchlenia. Tieto akcelerometer boli osadené na hlave figuríny, sedačke alebo inej časti vozidla [29]. Problémom analýzy týchto dát spočíva vo vlastnostiach jednotlivých akcelerometrov v mobile a tých používaných pri simuláciách. Najlepšie mobily majú rozsah len 8 až 16g, s pomerne malou vzorkovacou frekvenciou, ktorá môže klesnúť aj na menej než 5 Hz. Okamžitá zmena akcelerácie pri náraze môže byť prítomná len niekoľko milisekúnd. Preto pri nehode v zázname je k dispozícii len saturovaná hodnota v rozsahu daného akcelerometru, ktorá by sa len veľmi ťažko prirovnávala k akcelerometrom použitých pri simuláciách.

Preto bola zvolená cesta ako zistiť aktuálny pohybový stav vozidla v závislosti od dostupnej akcelerácie. Podľa dostupnej teórie bola zvolená hranica 4g, ktorá je dostatočne vysoká na potlačenie bežných aktivít a vedia ju bežné mobily zachytiť. Za prítomnosti GPS signálu sa skontroluje, dochádza k vyvolaniu 30 sekundového čakacieho intervalu. Počas tohto intervalu dochádza k vyhľadávaniu 3 podmienok, podľa ktorých sa určí prítomnosť havárie. Prvou z nich je, keď sa vozidlo pohybuje ďalej, tak vyvolá aktualizácie niekoľkých GPS súradníc a indikáciu autonehody môžeme zamietnuť.

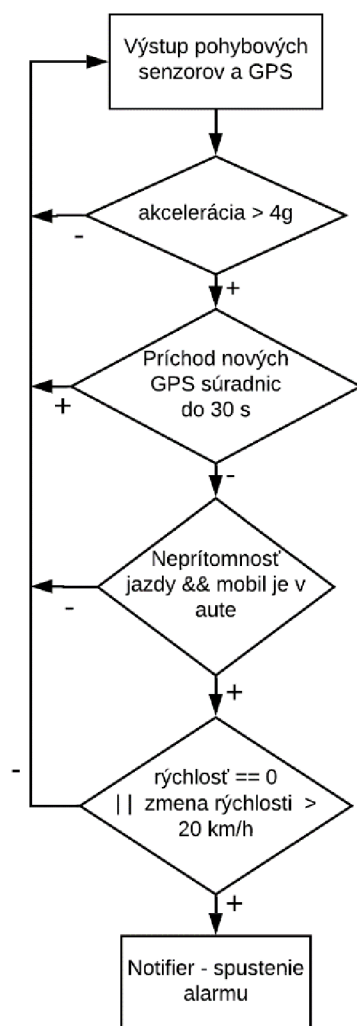
Ďalej počkáme na aktualizáciu rozpoznania aktuálnej aktivity zo systému, či sa mobil neustále nachádza v aute. Môže dôjsť k prípadu, že systém detekuje chôdzu, pričom automaticky sa prepne na detekciu pádu alebo dôjde k zamietnutiu autonehody a prejdeniu do režimu spánku. Táto podmienka hlavne vylučuje fakt, že užívateľ má natrvalo zapnutú detekciu autonehody a zároveň nie je v aute. Takto predídeme zbytočnému vyčerpávaniu batérie a falošným detekciám. Ak sa užívateľ nachádza vo vozidle, môžeme taktiež vyhľadať zvýšený rozptyl akcelerácie, ktorý je spôsobený otrasmi vo vozidle a motorom. Prah bol určený podľa meraní na hodnotu 0,03, ktorý keď je prekročený, prechádza sa na ďalšiu podmienku.

Následne ak je aktivita vozidla vylúčená, dôjde ešte ku kontrole zmeny hodnoty rýchlosti, ak je dostupný GPS signál. Pokiaľ rýchlosť nárazovo klesla o 20 km/h alebo je rovná nule, tak dôjde k vyvolaniu alarmu. Vyššia hodnota tohto prahu je tak nastavená z dôvodu možnosti nepresnosti GPS, ktorá sa môže posúvať z miesta na miesto bez pohybu mobilu. Dôvod je tiež výpočet samotnej rýchlosti. Súradnice GPS zvyčajne prichádzajú aj s aktuálnou rýchlosťou, ale v Androide to nie je pravidlo. Preto keď nie je dostupná rýchlosť, je nutné ju dopočítať manuálne medzi dvoma bodmi za využitia

nadmorskej výšky a samotných súradníc. V najhoršom prípade nie je prítomná ani nadmorská výška. Pre hodnotu nadmorskej výšky sú nutné 4 satelity, a keď sú prítomné len 3, tak dôjde k vytvoreniu súradnice bez nadmorskej výšky. V tomto prípade je vypočítaná rýchlosť medzi dvoma bodmi s nulovou výškou. Z tohto dôvodu je nutná táto podmienka, pretože môžeme prejsť do roviny väčšej aproximácie. Celý algoritmus je zobrazený v diagrame na Obr. 2.3 .

Pre overenie spoľahlivosti tohto algoritmu boli vyzbierané dáta dlhších ciest automobilom, ktoré boli prejdené skriptovanou verziou v Pythone. Tento program bude dodržiavať časovú následnosť každej udalosti, pričom kópia tohto algoritmu sa bude nachádzať aj v mobile. Pre overenie správnosti, ak sa v dátach nebude nachádzať pík nad 4g, bude do priebehu vložená aj umelá časť, ktorá bude obsahovať navýšenie akcelerácie nad 4g na pár milisekúnd a následne sa signál ustáli akoby sa nič nedialo.

Je nutné ešte poznamenať, že táto detekcia je plne závislá od prítomnosti GPS signálu. Preto keď nie je prítomný, aplikácia automaticky prechádza do spánkového režimu. Pri návrate signálu detekcia pokračuje ďalej.



Obr. 2.3 Diagram algoritmu detekcie autonehody

3. ZBER, SPRACOVANIE A ANALÝZA DÁT

3.1 Zber dát

Pre dosiahnutie najlepších výsledkov je potrebné vytvoriť čo najväčší dataset priebehov všetkých udalostí, čo sa týka detekcie pádu aj autohavárie. Na to poslúži aplikácia vlastnej výroby SensorBox voľne dostupná na Google Play. Jej zdrojový kód nie je predmetom tejto práce, ale jej vytvorenie bolo založené na získaní skúseností tvorby kompletnej Android aplikácie, senzormi a uverejnenia na Google Play. Ďalším dôvodom bolo aj zaistenie presnej interpretácie výsledkov produkovaných z mobilu, pretože odlišné aplikácie nezapisujú každý údaj, inak skresľujú meranie alebo neponúkajú možnosti potrebné pre vývoj našich modelov. Preto je SensorBox je vytvorený na zápis všetkých údajov produkovaných mobilom v surovej forme, pričom parametre užívateľa sú pevne dané v nastaveniach. Je možné meniť aktivitu GPS, teda interval aktualizácie a vzdialenosti potrebnej pre vyvolanie aktualizácie. V Androide sa nedá nastaviť presná vzorkovacia frekvencia pohybových senzorov, pretože ju riadi sám systém. Jediné, čo je možné zmeniť, je urgentnosť potreby vzoriek zo senzoru pre danú aplikáciu. Teda aktualizáciu zo vzorkovania môžeme čakať najneskôr v 4 rôznych intervaloch 0, 20, 66, 200 ms, pričom i pri nastavení 200 ms systém môže vyvolať aktualizáciu aj v oveľa menších intervaloch v závislosti od vyťaženia mobilu [8].

SensorBox zároveň ponúka meranie všetkých senzorov, ktoré sa v mobile nachádzajú, pričom sa v aplikácii nachádza kompatibilita so senzormi z Wear Os alebo Android Wear v minulosti. Výsledkom merania sú csv súbory v samostatnom priečinku, v ktorých je obsiahnutý UNIX čas v nanosekundách alebo milisekundách podľa senzoru, s hodnotami jednotlivých os merania a presnosti merania so stupnicou 0 až 3, kde 3 je najpresnejšie. Zároveň má aplikácia možnosť vytvorenia anotácií v priebehu merania, ktoré sú modifikovateľné v nastaveniach. V základnej verzii sa nachádzajú druhy nosenia mobilu, aktivity človeka a zaujímavých udalostí. Tieto anotácie sa pripnú v samostatnom textovom súbore, kde sa nachádza UNIX čas zaznamenania anotácie a jej kódové označenie, ktorých kódová zbierka je tiež zahrnutá v súbore. V tomto súbore sa tiež nachádzajú maximálne výstupné hodnoty senzorov a prípadná absencia niektorého zo senzorov.

Android v sebe obsahuje už spomenuté rozpoznanie aktuálnej aktivity. SensorBox tiež v sebe obsahuje tohto klienta a vytvára ďalšie dva súbory, v ktorých sú zachytené ich jednotlivé výsledky. Prvý obsahuje pravidelné aktualizácie jednotlivých pravdepodobností každej aktivity, ktorú je klient schopný rozpoznať. V druhej sú len zapísané časové známky, kedy sa systém rozhodol pre zmenu aktivity.

Zber dát rozdelíme na niekoľko druhov, kvôli potrebe stavby modelu, testovaniu detekcie autohavárie a aplikácie ako celku. Pre detekciu pádu bude potrebné vykonať jednotlivé merania pádu a udalostí podobných pádu. Meranie bude tvorené záznamami

pádov, chôdze, chôdze po schodoch, behu, skoku, sadnutia, pádu mobilu a obdobne, pričom pôjde o záznamy o dĺžke 10 s. Sensorbox obsahuje položku na časovo obmedzené záznamy a ich rýchle opakovanie. Vo väčšine štúdií spomenutých v predošlých kapitolách boli datasety vytvorené pomocou 10 až 20 ľudí, pričom každý vytvoril v priemere 10 opakovaní každej činnosti.

Problém nie je tvorba záznamov udalostí podobných pádu, pretože ide o činnosti, ktoré sú bežné a nepredstavujú žiadne riziko. Pri páde to bude opačne, pretože ich opakovanie môže byť bolestivé a potenciálne nebezpečné. Pri pádoch bola použitá vždy nejaká forma matrace alebo žinenky, čo môže ale viesť k skresleniu výsledkov. Nie vždy, keď človek padá, tak padá do mäkkého. Tento fakt ale môžeme potlačiť tým, že keď budeme schopní detekovať tieto pády, tak budeme schopní detekovať aj pády za vyšších amplitúd, pretože priebehy by sa mali stále podobat'. Ideálna bude ich kombinácia za ochoty dobrovoľníkov.

Zberu dát sa zúčastnilo 23 ľudí, z čoho bolo 15 vo veku 20 až 30 rokov, 7 vo veku 40-50 rokov a 1 osoba s vekom nad 70 rokov, ktorá ale nevykonávala pády. Pri meraní boli hlavne použité Sony Xperia Z5 a Huawei P9. Pri prítomnosti ďalšieho mobilu bol taktiež použitý. Spomenuté dva mobily sa pomerne líšia svojou charakteristikou senzorov. Sony zvyčajne pracuje s dvojnásobnou vzorkovacou frekvenciou oproti Huawei, ale je obmedzený len na rozsah 4 g, zatiaľ čo Huawei disponuje rozsahom až 8 g. Preto pri meraní boli obidva mobily použité zároveň, aby modelu zabezpečili variabilitu aj v tejto oblasti. Zároveň boli mobily uložené v predných vreckách nohavíc alebo vo vreckách bundy. Detekcia teda nie je uspošobená na detekovanie pádu, keď sa mobil nachádza v taške alebo batohu, z dôvodu nekonzistencie zmeny akcelerácie tašky a ľudského tela. Pridaním takýchto meraní by sme mohli znížiť celkovú úspešnosť modelu.

Detekcia autohavárie je testovaná na kolekcii záznamov vytvorených počas jazdy autom. Mobily sú uložené u samotného pasažiera alebo v nejakom úchyte, v zmysle puzdra v dverách alebo nosiči na mobil, ktorý sa používa pri navigácii. Keďže neočakávame vyvolanie autohavárie, cieľom je zistiť bežné zrýchlenia a zmeny rýchlosti vozidla a odfiltrovať bežné situácie.

Pre testovanie modelu môžeme využiť meranie aktivity v priebehu celého dňa. SensorBox umožňuje obmedzenie intervalu merania na užívateľom zvolený čas. Tým pádom môžeme vytvoriť kolekciu záznamov aktivity v priebehu celého dňa a vytvoriť skript na prejdenie týchto záznamov s našim modelom. Tieto záznamy delíme na bežné aktivity človeka a aktivity spojené s dlhšou jazdou v aute.

3.2 Spracovanie dát detekcie pádu

Nasledujúca Tab. 3.1 zhrňuje aktuálny počet dobrovoľníkov a počet meraní jednotlivých udalostí. Udalosti sú klasifikované na 4 triedy pre jednoduchšiu interpretáciu na báze podobnosti pohybu. V závislosti od distribúcie parametrov sa môže táto klasifikácia zmeniť napr. na 2 triedy, teda pád a všetko ostatné.

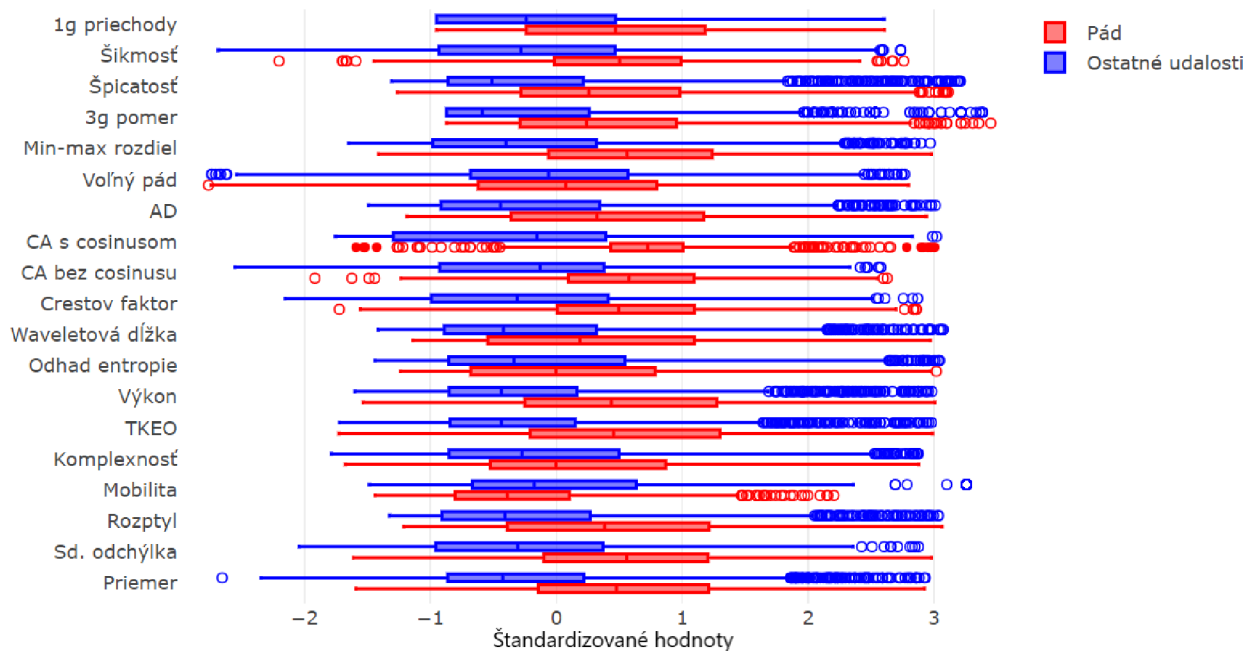
Tab. 3.1 Počet dobrovoľníkov/udalostí

Počet dobrovoľníkov	23
Celkový počet záznamov	3260
Počet vylúčených záznamov	131
Počet pádov	840
Počet behov/chôdze/skokov/potknutí	667
Počet sadov	1071
Počet ľahov	551

Záznamy sú 10 s dlhé a obsahujú jednu zo spomínaných udalostí. Udalosti sú vopred definované v názve priečinku vytvorenom pomocou SensorBox. V každom zázname je obsiahnutá len konkrétna daná udalosť alebo sa pred udalosťou nachádza jemné rušenie v podobe chôdze. Pred spracovaním sú vypustené merania, v ktorých sa nenachádza pík o veľkosti minimálne 3g, pretože menšie hodnoty nebudú vecou detekcie. Merania, u ktorých bol pík nájdený v prvej alebo poslednej sekunde, boli tiež vyradené, pretože CA s kosínusom potrebuje rezervu. Rovnako keď meranie je svojou dĺžkou mimo rozsahu 9 – 11 s alebo medzi vzorkami sa nachádza medzera dlhšia ako 200 ms, meranie je vylúčené. Takéto merania sú spôsobené okamžitou zvýšenou aktivitou mobilu.

Z celého záznamu je cieľom vybrať okamih pádu v rozmedzí približne 1 s, preto budeme postupovať nasledovne. V prvom rade dôjde k určeniu maxima v danom priebehu akcelerácie, ktoré jednoznačne určuje polohu klasifikovanej udalosti, pretože nebudú prítomné výraznejšie rušenia a chôdza bude mať vždy menšiu hodnotu ako ostatné udalosti. Následne sa od konca merania hľadá prvá hodnota vyššia ako 1,5g, ktorá je maximálne vzdialená 0,7 s od píku, a tento bod bude uznaný ako koniec udalosti. Od píku na opačnú stranu sa hľadá prvá hodnota voľného pádu pod 0,8g. Tieto hodnoty môžu vykazovať šum, čo je ošetrené istou toleranciou voči vyšším hodnotám, pokiaľ sa za nimi sa nachádza hodnota menšia ako 0,8g. Pri absencii zložky pod 0,8g sa ako začiatok označí vzorka, ktorá je od píku vzdialená 0,3 s.

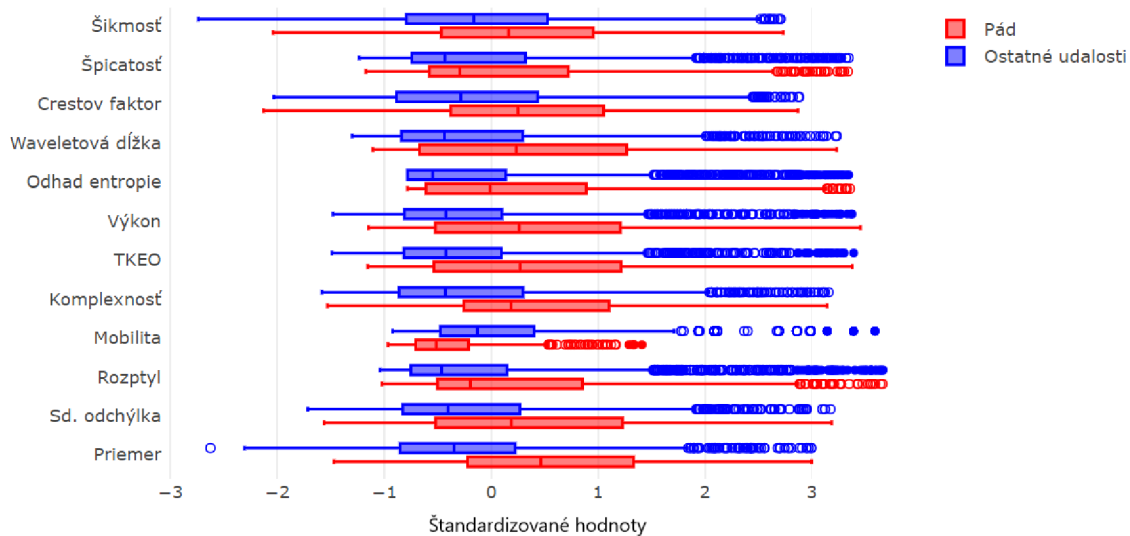
Tieto body počiatku a konca sú aproximované aj do merania z gyroskopu. Meranie uhlovej rýchlosti z gyroskopu je prepočítané aj na uhlové zrýchlenie. Vypočítané parametre sú následne uložené s príslušnou hlavičkou do csv súboru, ktorý použijeme na vizualizáciu a selekciu parametrov vhodných na klasifikáciu pádu. Dataset tvorí 43 parametrov, z toho najväčší podiel patrí parametrom z akcelerometru. Všetky dáta boli štandardizované a zobrazené do grafov.



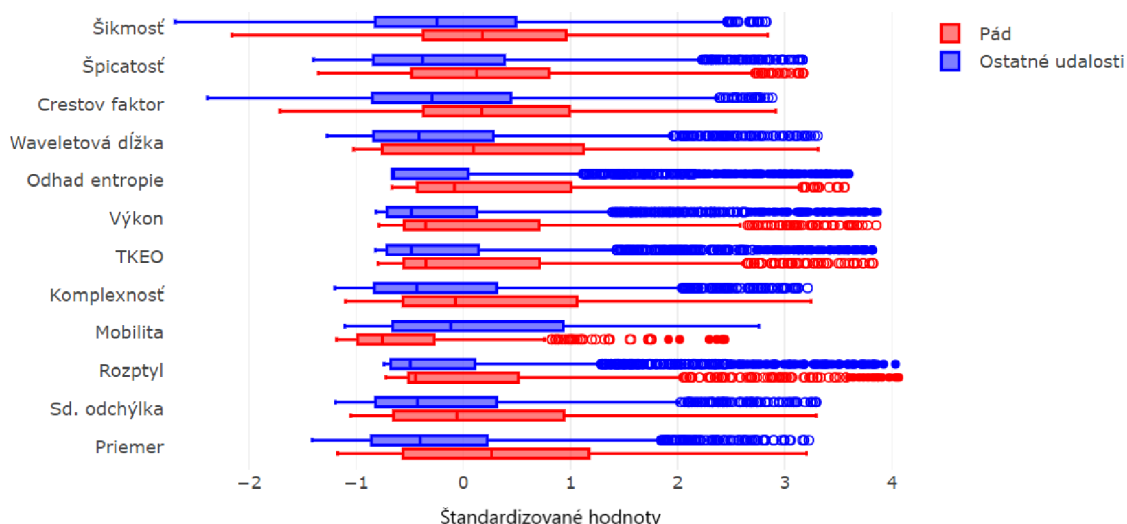
Obr. 3.1 Vizualizácia štandardizovaných parametrov akcelerácie

Pre odstránenie odľahlých hodnôt bolo zvolené rozšírené interquartilné pravidlo, ktoré popisujú aj samotné obrázky. V rámci každého parametra boli nájdené odľahlé hodnoty, ktoré boli následne spriemerované s 5 najbližšími hodnotami, pre zachovanie variability v dátach, zachovaniu počtu dát, a hlavne aby sme nespôsobili nerovnováhu v distribúcii dát.

Cieľom je určiť parametre, ktoré sú značne odlišné pri páde. Existuje množstvo metód na výpočet relevantnosti jednotlivých parametrov, ale kvôli obmedzeným možnostiam mobilu z pohľadu dostupnej batérie, sme nútení vystačiť si s jednoduchými parametrami na výpočet. Pre začiatok tieto dáta môžeme rozanalyzovať vizuálne po vykreslení štandardizovaných hodnôt na krabicový graf. Na prvý pohľad môžeme vidieť, že dochádza k odlišeniu hodnôt väčšiny parametrov, a to hlavne pri akcelerácii na Obr. 3.1. Môžeme vyzdvihnúť napríklad výpočet výkonu a TKEO, ktoré značne popisujú vysoké hodnoty piku v priebehu pádu. Zároveň aj priemer, smerodajná odchýlka a rozptyl deklarujú daný rozdiel. Ďalej aj Crestov faktor, Waveletová dĺžka, CA koeficient s cosinusom a AD koeficient dokázali vytvoriť značnú hranicu medzi pádom a inými udalosťami. Ďalšie údaje tiež majú túto schopnosť, ale trpia už so zvýšeným šumom od iných udalostí ako je pád, preto nebudú použité v modeli. Napr. mobilita a komplexnosť nevykazujú oddelenie kategórií, preto pravdepodobne nebudú zahrnuté do modelu. Zaujímavým prípadom je CA koeficient bez cosinusu, kde len niektoré prípady ľahu zapadli medzi pády, pričom pri ich ignorancii ide asi o najjednoduchší parameter.



Obr. 3.2 Vizualizácia štandardizovaných parametrov uhlovej rýchlosti



Obr. 3.3 Vizualizácia štandardizovaných parametrov uhlového zrýchlenia

Z hodnôt uhlovej rýchlosti na Obr. 3.2 a uhlového zrýchlenia na Obr. 3.3 z gyroskopu je týchto špecifických parametrov menej, takmer až neexistujúce, pretože hodnoty sú náhodne rozmiestnené a nevykazujú žiadne pravidlo. Jedny z mála s dobrým výsledkom boli parametre výkonu a TKEO prípadne odhad entropie, priemer, smerodajná odchýlka a rozptyl. Z tohto dôvodu bolo vyskúšané rôznymi transformáciami, či dané parametre sú schopné sa podieľať na oddelení pádu a iných udalostí.

Boli otestované tri metódy transformácii s dátami, ktoré využívajú nejakú formu lineárnej kombinácie. Prvou z nich bola PCA, alebo v preklade analýza principiálnych komponentov. PCA slúži na dekoreáciu a kompresiu dát, pričom dochádza k ich

transformácii na jednotlivé komponenty. PCA určuje komponenty na základe rozptylu, teda prvý komponent obsahuje najvyšší rozptyl obsiahnutý v danej sústave dát. Jednotlivé body oddelené pozdĺž prvého komponentu sú viac odlišné ako jednotlivé body pozdĺž druhej komponenty, pretože reprezentujú medzi sebou vyššiu mieru rozptylu. PCA nachádza svoje uplatnenie vzhľadom na dekoleráciu ako kompresný nástroj.

ICA, nezávislá analýza komponentov, je tiež obdobnou metódou ale jej hlavným cieľom je izolovať jednotlivé zdroje dát na základe minimalizácie spoločnej informácie medzi zdrojmi a snahy dosiahnuť nie normálne rozdelenie dát. Na dosiahnutie týchto predpokladov sú využité rôzne metódy v závislosti od algoritmu.

NMF, nie negatívna maticná faktorizácia, má tiež charakter lineárnej kombinácie. Pri nej je nutné zabezpečiť, aby všetky hodnoty boli kladné, čo uľahčuje niektoré predpoklady v rámci výpočtu. Dochádza k rozkladu matice dát na ďalšie dve matice, ktorých násobením je možné vytvoriť späť tú pôvodnú. Zvyčajne prvá reprezentuje základné zdroje dát a druhá jednotlivé váhy týchto zdrojov. Na výpočet sa tiež používajú rôzne numerické metódy pre dosiahnutie čo najvyššej aproximácie uvedených dát.

Všetky tieto metódy obsahujú sadu váh, podľa ktorej je možné prevádzať jednu sústavu do druhej, pričom každý bod je závislý od všetkých parametrov. Preto môžeme jednotlivé tieto váhy v rámci rôznych komponentov priemerovať a tým zistiť, ktoré príznaky sa podieľajú na zostavení danej distribúcie v najvyššej miere.

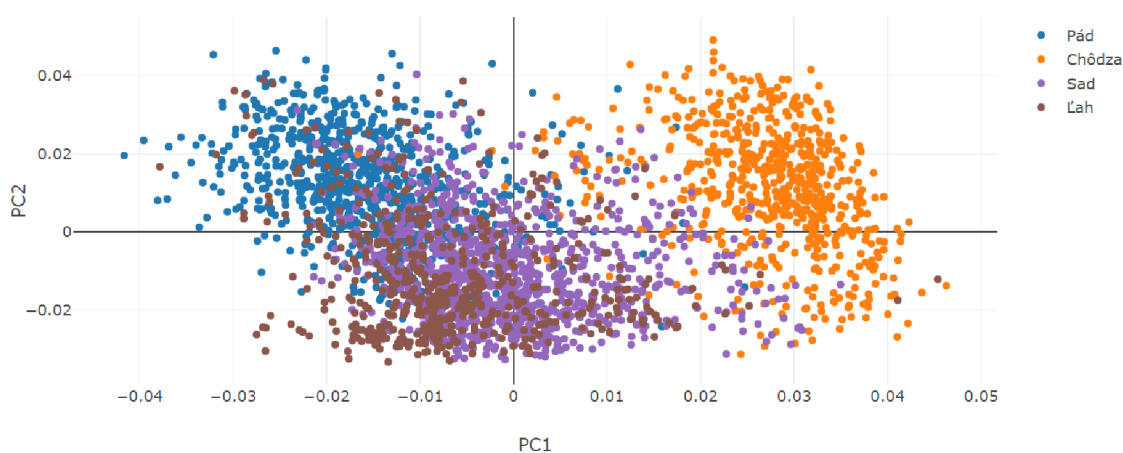
Metódy boli v prvom rade aplikované na kompletne dáta akcelerácie, uhlovej rýchlosti a uhlového zrýchlenia. V tejto reprezentácii boli najviac reprezentované dáta uhlového zrýchlenia a uhlovej rýchlosti. Je nutné podotknúť, že i po normalizácii tieto dáta obsahujú najvyššiu mieru rozptylu, preto ich tieto metódy zvyčajne uprednostňujú pri ich stavbe.

Na každý zdroj dát boli aplikované jednotlivé transformácie. Bolo zistené, že keď sa pozrieme na jednotlivé reprezentácie každého senzoru, tak parametre akcelerácie odvedli najlepšiu prácu čo sa týka odlišenia pádu a iných udalostí. Parametre uhlovej rýchlosti a zrýchlenia boli lepšie v rozpoznaní sadu a ľahu, čo iba čiastočne napomáha riešeniu. Zároveň ale dochádza k pomiešaniu jednotlivých kategórií s vyššou mierou zhlukovania sadu a ľahu. Pri akcelerácii dochádza k odlišeniu chôdze podobným činnostiam, pádu a zároveň sad a ľah sa zhlukujú dohromady. Z tohto dôvodu bolo rozhodnuté, že dôjde k vypusteniu gyroskopu. Tieto dáta možno neprispievajú k riešeniu v danej metodike, ale môžu nájsť svoje uplatnenie pri inej, ktorá zahrňuje vyššiu mieru kontextu pri pohybe človeka a mobilu. Výhodou bude šetrnejší prístup k batérii, keďže gyroskop je najnáročnejší senzor.

Podľa údajov prístupných z daných transformácií a distribúcií jednotlivých parametrov boli zvolené jednotlivé parametre. Podľa distribúcií sa najviac na oddelení pádu podieľa CA s cosinusom a bez cosinusu, min-max rozdiel, 3g pomer. Tieto parametre boli pri transformáciách označené ako najmenej prispievajúce. Tento dôvod nájdeme v tom, že dané parametre prispievajú k oddeleniu pádu, ale svojou

variabilitou/nezávislosťou nie sú vhodné pre dané transformácie, pretože sa tam nachádzajú parametre, ktoré vykazujú oveľa vyššiu mieru variability. Čo ale má vysokú variabilitu a zároveň do istej miery oddeľujú pád, je Crestov faktor, uhlová odchýlka, TKEO, výkon, waveletová dĺžka a iné.

Boli zvolené parametre, ktoré priamo oddeľujú pád, ku ktorým sme postupne pridávali rôzne ďalšie parametre podľa transformácii. Najviac vydarená kombinácia bola pri zvolení CA s kosínusom a bez kosínusu, Crestov faktor, min-max rozdiel, 3g pomer, uhlová odchýlka, špicatosť a šikmosť. Pri použití ICA len na tieto parametre vznikol Obr. 3.4. Obdobné vznikli za použitia PCA a NMF.



Obr. 3.4 ICA aplikované na vybrané parametre

Na Obr. 3.4 môžeme vidieť jednotlivé oddelenia udalostí. Všetky dostupné transformácie pri redukcii počtu parametrov vyťažili najviac z CA parametrov, ktoré poslúžili na oddelenie chôdze a iných udalostí. Mobil pri chôdzi vlastne nezmení význačne uhol ku gravitácii, preto sú tieto hodnoty zásadne iné ako pri iných činnostiach.

Pád, ľah a sad sa už na seba viac podobajú, preto tvoria pomerne jednoliaty zhluk, ktorý ale v sebe obsahuje nejasnú hranicu pádu a ostatných udalostí. Pri pridávaní jednotlivých parametrov a ich odoberaní sa výsledok už nezlepšoval, ale s danou reprezentáciou je silný predpoklad, že pri tréningu NN dôjde k dosiahnutiu dobrého výsledku.

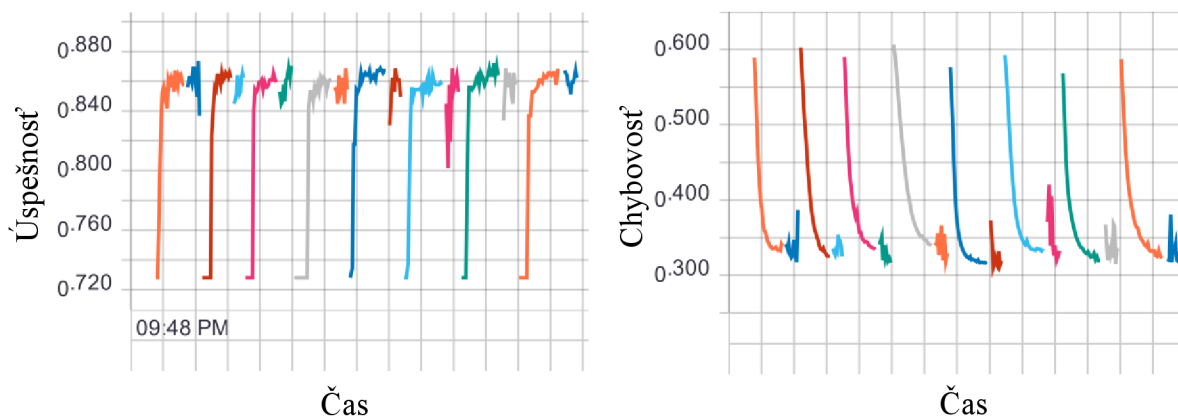
3.3 Výsledky tréovania neurónovej siete

Pri tréovaní neurónovej siete bolo testované, aký minimálny počet neurónov je potrebný na dosiahnutie vhodného riešenia, aký vplyv má algoritmus učenia na výsledok, rýchlosť učenia a taktiež vplyv rôzneho počtu epoch pre preverenie všetkých druhov riešení. Pri testovaní boli použité rôzne podiely tréovanej časti a validačnej časti, pretože dáta sa postupom času pridávali. Jednotlivé kategórie boli prekódované do dvoch skupín, kde 0 značila pád a 1 značila inú udalosť. Príznaky boli normalizované podľa minimálnych maximálnych hodnôt do rozmedzia 0 až 1.

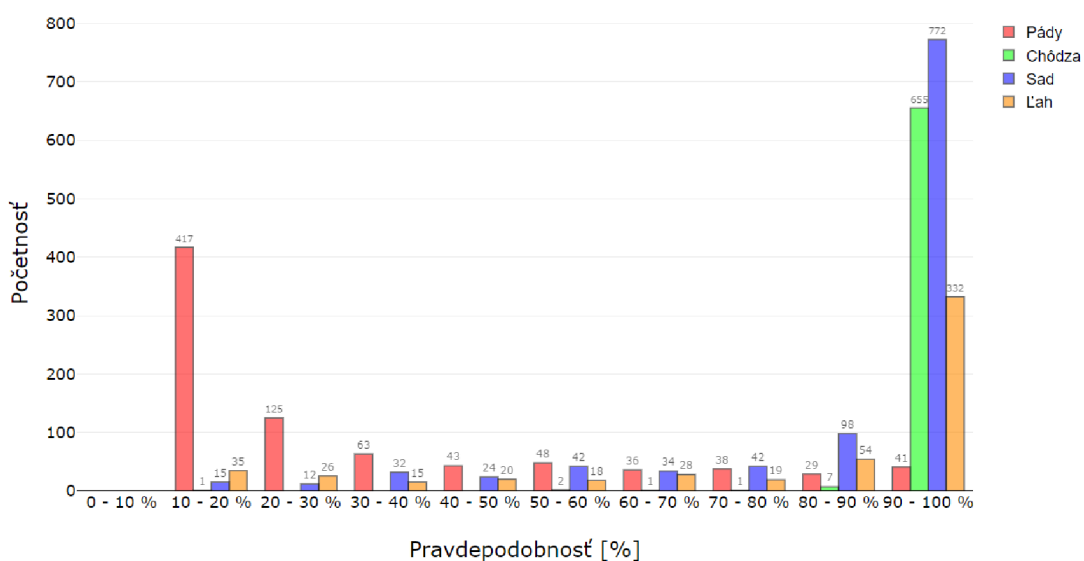
V prvom rade bolo otestované, či počet neurónov a počet skrytých vrstiev bude mať vplyv na výsledné riešenie. Na vstupe bol konštantný počet neurónov odpovedajúci počtu parametrov, pričom sa menil počet skrytých vrstiev a ich počty neurónov. Boli vyskúšané kombinácie, kedy nebola prítomná žiadna skrytá vrstva až po 3 vrstvy o 40 neurónoch. Vrstvy sa pridávali postupne a pridávalo sa po 5 neurónoch pri každej vrstve. Boli vyskúšané algoritmy aj ako aj SGD, pričom mali rovnaký počet 75 epoch. Algoritmom sa zároveň menili rýchlosti učenia, ktoré boli posunuté o desatinné miesto vpred, dozadu alebo na pôvodnú rýchlosť.

Výsledkom tohto tréovania bolo 160 neurónových sietí, ktoré dosiahli úspešnosť medzi 86 % až tesne nad 90 % pri využití 900 meraní. Len niektoré väčšie siete s SGD spomaleným algoritmom neboli schopné vyjsť z počiatku na 70 %. Úspešnosť sa začala dvíhať až na konci celkového učenia, čo popisujeme ako underfitting. Väčšine stačilo 10 epoch na to, aby tréovanie skonvergovalo k riešeniu. Dobrou správou je, že tak nedochádza ani k preučeniu siete, overfittingu, pretože pri validácii i pri prehnanom počte epoch, nedochádza k navýšeniu chyby. Z tohto môžeme usúdiť, že štruktúra nemá vplyv na riešenie úlohy a rýchlosť učenia môžeme nechať pôvodnú, pričom môžeme využiť akýkoľvek algoritmus. Najlepší výsledok prezentovala sieť s 93 % úspešnosťou.

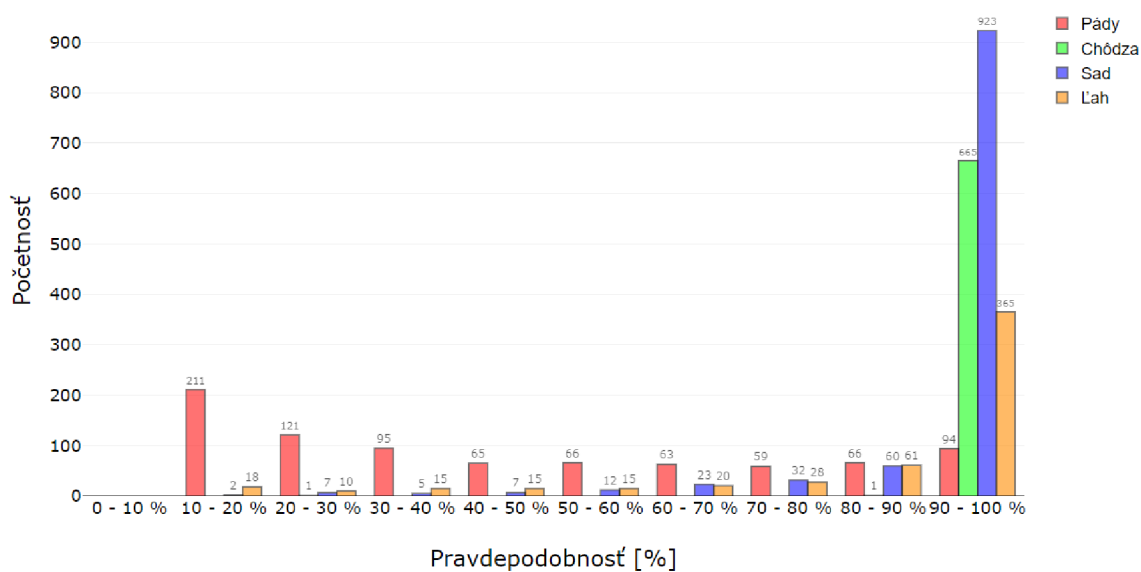
Najlepšou konfiguráciou je teda využitie len jednej vstupnej vrstvy a jedného výstupného neurónu za využitia SGD alebo adama. S nárastom dát ale počet potrebných epoch narástol z 10 až na 25, pričom klesla aj úspešnosť, ktorá dosiahla maximálne 89%. Štruktúru siete môžeme ovplyvniť už len jediným spôsobom a to zmenou počtu vstupných neurónov. Počty neurónov boli zmenené od 3 do 10 neurónov, pričom ako počiatočný optimalizačný algoritmus bol použitý adam na 25 epoch. Následne bol na ďalších 10 epoch bol použitý SGD s menšou rýchlosťou o štvrtinu učenia na doladenie a ustálenie daného riešenia. Týmto spôsobom bolo vytvorených ďalších 8 neurónových sietí, ktoré dosiahli v priemere úspešnosť 86 %, pričom bol použitý plný počet vzoriek, z čoho 20 % bolo použitých na validáciu siete. Jednotlivé priebehy sú zobrazené na Obr. 3.5, pričom medzera v danom priebehu symbolizuje zmenu algoritmu na SGD. Obr. 3.6 a 3.7 reprezentujú klasifikáciu databázy. Tým, že pád je kódovaný ako 0, tak zobrazená pravdepodobnosť blížiaci sa nule reprezentuje pád a opačne inú udalosť.



Obr. 3.5 Úspešnosť/chybovosť neurónových sietí v postupnosti od 3 do 10 neurónov



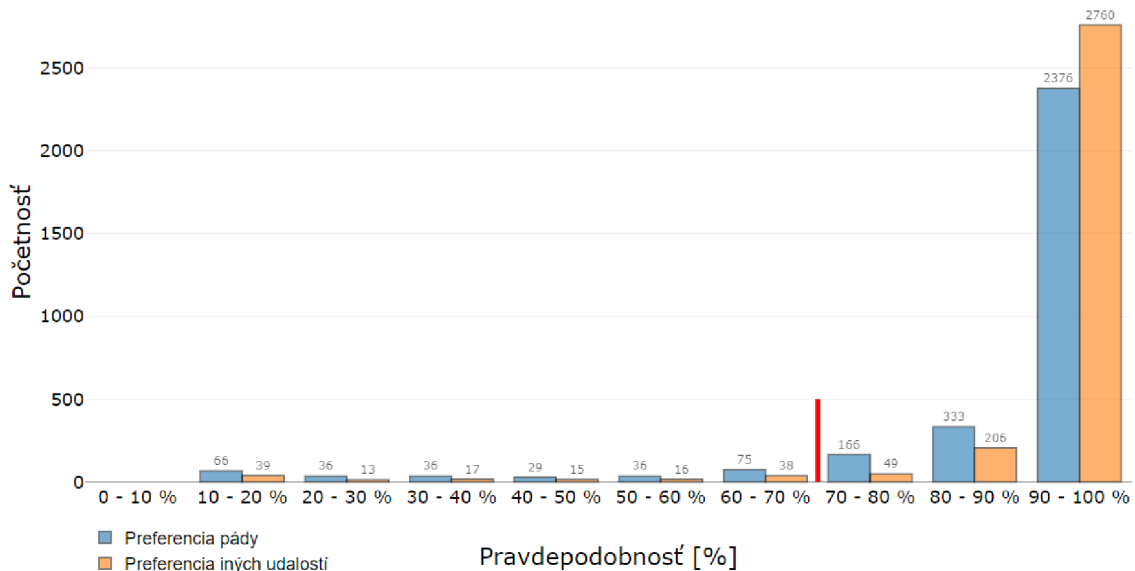
Obr. 3.6 Distribúcia menej špecifického modelu voči iným udalostiam



Obr. 3.7 Distribúcia menej špecifického modelu voči pádom

Pri pohľade na jednotlivé priebehy tréovania NN vidíme, že sú takmer identické. SGD spôsobilo, že niektoré siete boli schopné dosiahnuť aj 87 % v niektorom okamihu. Problémom ale je, že táto úspešnosť nereprezentuje schopnosť siete plne odlišovať pád a iné udalosti. Pri pohľade na rôzne distribúcie jednotlivých kategórii pri využití rôznych NN tak narazíme na dva hlavné vzory, ktoré môžeme vidieť na Obr. 3.6 a 3.7 na predošlej strane.

Jedna distribúcia preferuje detekciu pádu na úkor falošne pozitívnych detekcií iných udalostí, zatiaľ čo druhá distribúcia presne naopak, preferuje detekciu iných udalostí na úkor falošne negatívnych detekcií pádov. Je ťažko povedať, ktorá distribúcia je vhodnejšia. Z tohto dôvodu boli zozbierané ďalšie priebehy účastníkov počas dňa. Tieto priebehy boli prejdené skriptom na výber zaujímavých výsekov o dĺžke 10 s, ktoré odpovedajú našej charakteristike detekcie pádu. Celkovo bolo vyzbieraných 300 h záznamov z akcelerometru, z čoho bolo vybraných 3153 úsekov na analýzu pre NN. Bol zvolený model so 6 neurónmi, ktorý mal vyššiu preferenciu voči iným udalostiam, a model s 9 neurónmi s vyššou preferenciou pre pády.



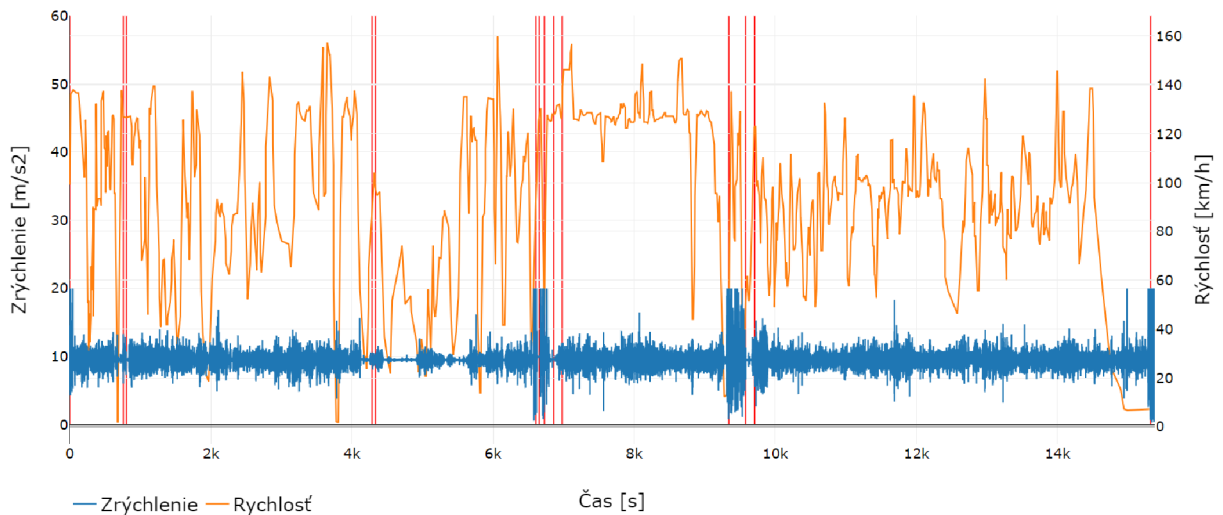
Obr. 3.8 Test NN na reálnych dátach - úspešnosť modelov

Ako môžeme vidieť na Obr. 3.8, model s preferenciou pre iné udalosti odviezol lepšiu prácu v rámci denných aktivít a vyvolal o 140 menej falošných alarmov ako druhý model. Vzhľadom na dostupnú distribúciu je možné ale usúdiť, že pri páde bude model schopný reagovať správne. Ako hraničná hodnota pre NN bola zvolená 0,7, pretože za touto hodnotou sa nachádza väčšina detekcií pádov a minimum iných udalostí. Na Obr. 3.8 je zvýraznená hranica červenou líniou. Táto hodnota je meniteľná v nastaveniach aplikácie.

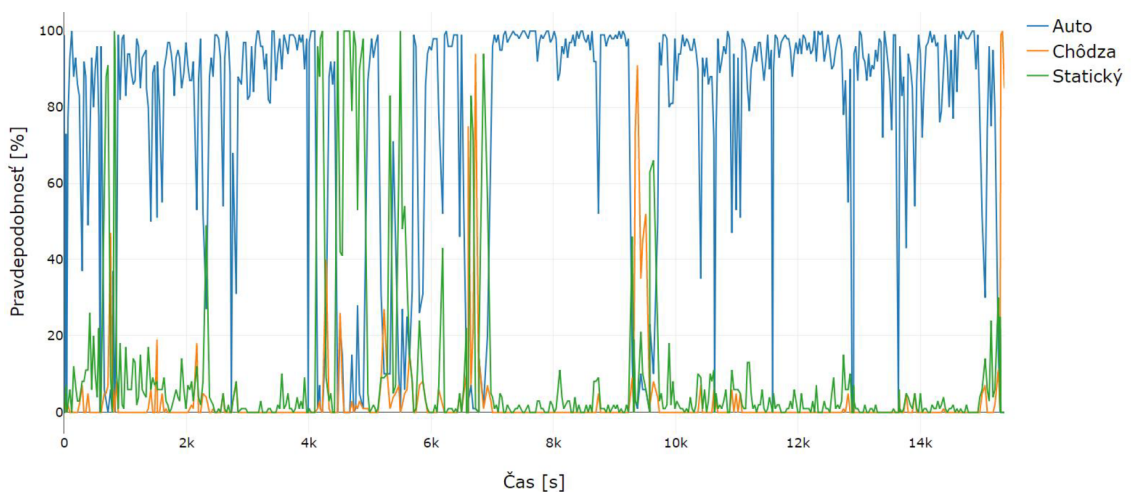
3.4 Zber a analýza dát pre detekciu autohavárie

Zber dát prebiehal počas bežnej jazdy, kedy boli mobily uložené na rôznych miestach, napr. vo vrecku, v bočnom priečinku dverí, na palubovej doske auta, a podobne. Cieľom bolo zistiť, aký priebeh má akcelerácia počas rôznych dopravných situácií a aká je pravdepodobnosť, že akcelerácia náhodne presiahne našu stanovenú hranicu 4g.

Celkovo bolo zozbieraných 28 h jazdy, pričom bolo prejdené približne 2500 km po komunikáciách všetkých tried, či mestskom alebo inom prostredí. Vybierané boli údaje z akcelerometru, GPS a údaje pochádzajúce z klienta na rozpoznanie činnosti, ktoré sa rozdeľujú na dva súbory. Prvý súbor obsahuje jednotlivé pravdepodobnosti, či sa mobil nachádza v aute, je stacionárny alebo je v statickej pozícii. Druhý súbor obsahuje časové známky, kedy sa klient rozhodol zmeniť pohybový stav mobilu. Na nasledujúcich obrázkoch môžeme vidieť priebeh cesty z Popradu (SVK) do Brna, čo je približne 380 km dlhá cesta s dobou cestovania približne 5 h. Cesta je tvorená cestami rôznej kvality, dopravnými situáciami a dvomi prestávkami. Mobil bol uložený vo vrecku.



Obr. 3.9 Popis akcelerácie, rýchlosti a zmeny pohybového stavu (červené línie)

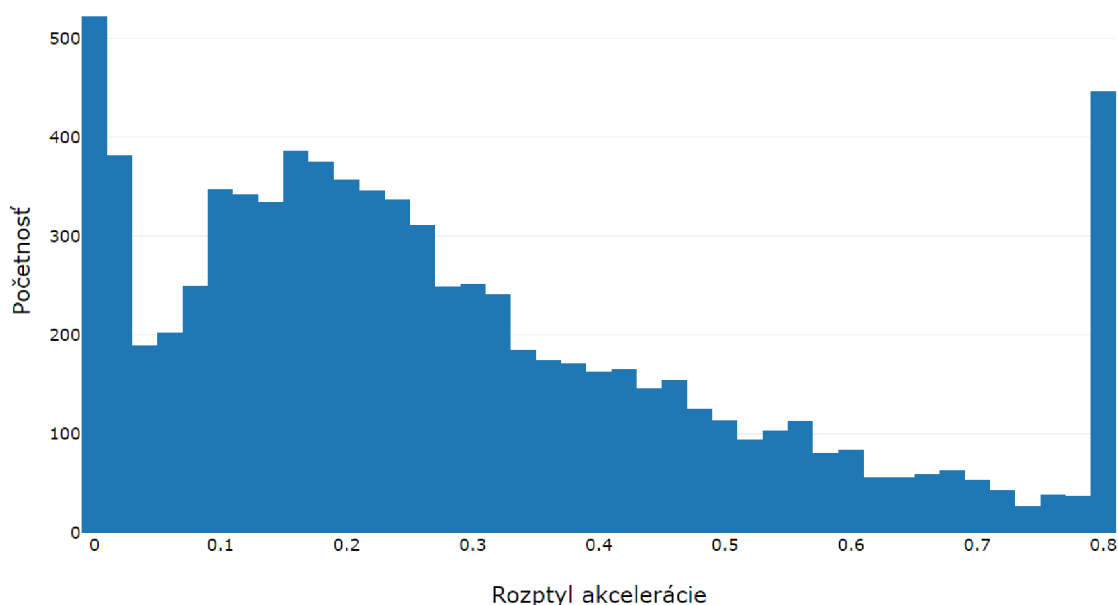


Obr. 3.10 Zmena pravdepodobnosti jednotlivých pohybových stavov počas jazdy

Ako môžeme vidieť na Obr. 3.9, zrýchlenie len veľmi ťažko vyjde nad 4g. Jediným prípadom dosiahnutia tohto prahu je, ak človek vyjde z auta a začne kráčať. Toto rušenie vieme veľmi ľahko eliminovať pomocou klienta na rozpoznanie činnosti. Na základe pravdepodobnosti vieme určiť, čo pravdepodobne daný človek vykonáva.

Na Obr. 3.10 je zobrazené ako sa menia jednotlivé pravdepodobnosti. Približne v momente, keď jedna z udalostí presiahne 60 % hranicu, dochádza v rámci Androidu k zmene pohybového stavu, čo je popísané červenými líniami na Obr. 3.9. Na základe tejto zistenej pravdepodobnosti, môžeme ignorovať hodnoty akcelerometru, ak hodnoty danej akcelerácie predchádzala zvýšená pravdepodobnosť chôdze nad 60 %. Zároveň je nutné podotknúť, že keď sa mobil nachádza v statickom postavení, neznamená to, že človek vyšiel von z vozidla. Zvýšená pravdepodobnosť statickej polohy popisuje aj zastavené vozidlo, alebo vozidlo pohybujúce sa v kolóne, čo môžeme vidieť medzi 4000 – 6000 s medzi Obr. 3.9 a 3.10.

Pre vytvorenie detekcie je nutné vytvoriť ale prah pre detekciu prítomnosti jazdy v aute. Aktualizácia pohybového stavu mobilu nemusí byť dostupná z dôvodu spánku mobilu. Preto ako ďalšiu poistku zavedieme detekciu jazdy prostredníctvom rozptylu hodnôt v akcelerácii. Pri jazde je symbolické neustále chvenie spôsobené jazdou auta, či vibráciami iných častí vozidla. Toto je zdroj detekcie aj pre klienta na rozpoznanie jazdy vo vozidle. V príkladnej situácii dôjde k zvýšeniu zrýchlenia nad 4g. Keď neprídu nové súradnice z GPS alebo aktualizácia o zmene pohybového stavu, ktorá spôsobí zmenu polohy užívateľa mimo auto, tak dôjde ku kontrole rozptylu akcelerácie po tejto dobe. Z vyzbieraných údajov bol pomocou 10 s okien vypočítaný histogram na Obr. 3.11. Okná, v ktorých boli hodnoty vyššie ako 14 m/s^2 , boli vylúčené, a okná, ktoré presahovali hodnotu nad 0,8, boli presunuté do tohto jedného poľa pre lepšiu vizualizáciu.



Obr. 3.11 Histogram rozptylu akcelerácie pri jazde auta

Na Obr. 3.11 môžeme vidieť oddelenia voľnej línie a rozptylov, ktoré sú spôsobené jazdou vozidla. Rozptyl pri vozidle v relatívnom klúde sa pohybuje najvyššie po hodnotu 0,03, ktorú použijeme pri našej detekcii. Vyššie hodnoty skôr indikujú priebeh jazdy. Táto podmienka je tu hlavne z dôvodu, keby náhodou došlo ku strate signálu priamo pri autonehode, teda podmienka pre prítomnosť GPS by nebola využiteľná. Rozptyl je použitý taktiež z dôvodu posunu stabilnej hodnoty akcelerometru v mobile, ktorý časom môže vykazovať nepresnosti pri meraní.

Pre otestovanie bežných dopravných situácií boli otestované aj podmienky ako prudké brzdenie z 50 na 0 km/h, vytočenie 90° zákruty pri 40 km/h alebo prejdenie podvalom pri rýchlosti 40 km/h. Ani jeden z týchto menších pokusov nepreukázal schopnosť vyskočiť s hodnotou zrýchlenia nad 4g. Tieto priebehy sú zobrazené v prílohe č. 3.

Pre kontrolu naprogramovaného skriptu bola vytvorená trapa algoritmu, ktorý sa nachádza v mobile ako python skript. Výsledkom prešetrenia všetkých dostupných záznamov bolo 61 indikácií na autonehodu, z toho ale ani jedna algoritmom nebola potvrdená. Najčastejším aspektom, ktorý zruší danú indikáciu je dostupnosť ďalších GPS známk, ktoré naznačujú pohyb vozidla. Inak v tomto smere prispieva klient na rozpoznanie činnosti, ktorý dokáže odfiltrovať situácie, kedy užívateľ len vyjde z vozidla a začne chodiť.

4. ZLOŽENIE ANDROID APLIKÁCIE

Kapitola popisuje základné stavebné objekty aplikácie, ich funkcionálnosť a zároveň isté limitácie. Aplikácia je naprogramovaná pomocou jazyka Java prostredníctvom Android studio. Spoločnosť JetBrains vydala svoj jazyk, Kotlin, ktorý rieši niektoré nevhodné uplatnenia Javy. Samotné Android studio je vyvíjané Googlom v kooperácii so spoločnosťou JetBrains. Kotlin už bol v roku 2017 implementovaný do editora a v máji 2019 sa stal ako preferovaný jazyk pre tvorbu Android aplikácií. Taktiež tieto jazyky môžu existovať v rámci jednej aplikácie bez problémov. Nanešťastie ale niektoré potrebné knižnice ešte nie sú plne vyvinuté alebo sú až neprítomné. Navyše táto aplikácia nadobúda aj experimentálne knižnice, ktoré by bez Javy neboli funkčné, preto je celý kód aplikácie výhradne napísaný v Jave.

4.1 Základne rozhranie aplikácie

Pri tvorbe aplikácie sa stretávame s dvoma hlavnými objektami, a to s aktivitou, Activity, a službou, Service, ktoré dedia kontext. Vďaka nim tieto objekty dostávajú plnú paletu metód, ktorá umožňuje interakciu s celým systémom. Zásadný rozdiel je, že aktivita obsahuje grafiku, ktorá je prezentovaná užívateľovi, zatiaľ čo služba je len proces na pozadí, ktorý je maximálne reprezentovaný notifikáciou a používa sa napríklad na synchronizáciu, sťahovanie súborov a podobne. Samotná domovská plocha je tiež istou formou aktivity.

Aplikácia obsahuje 6 aktivít, konkrétne: uvítacie snímky pri prvotnom spustení, hlavnú obrazovku, z ktorej sa dá následne dostať do nastavení, nastavenia rozvrhu detektorov a aktivitu zobrazujúcu licencie knižníc tretích strán, a nakoniec aktivitu alarmu. Celá detekcia sa odohráva na jednej službe na pozadí, ktorá je schopná komunikovať na diaľku s hlavnou aktivitou. Pre zistenie vlastností senzorov je pri prvom behu aplikácie použitá dočasná služba.

Uvítacia plocha obsahuje posúvaciu plochu s popisom aplikácie a jej fungovania, pričom zároveň požaduje potvrdenie od užívateľa, že využívanie aplikácie je na vlastnú zodpovednosť. Nakoniec je ešte užívateľ požiadaný na prihlásenie cez svoj Google účet. Toto prihlásenie je potrebné pre využitie určitých služieb v Androide na kontrolu správneho fungovania aplikácie.

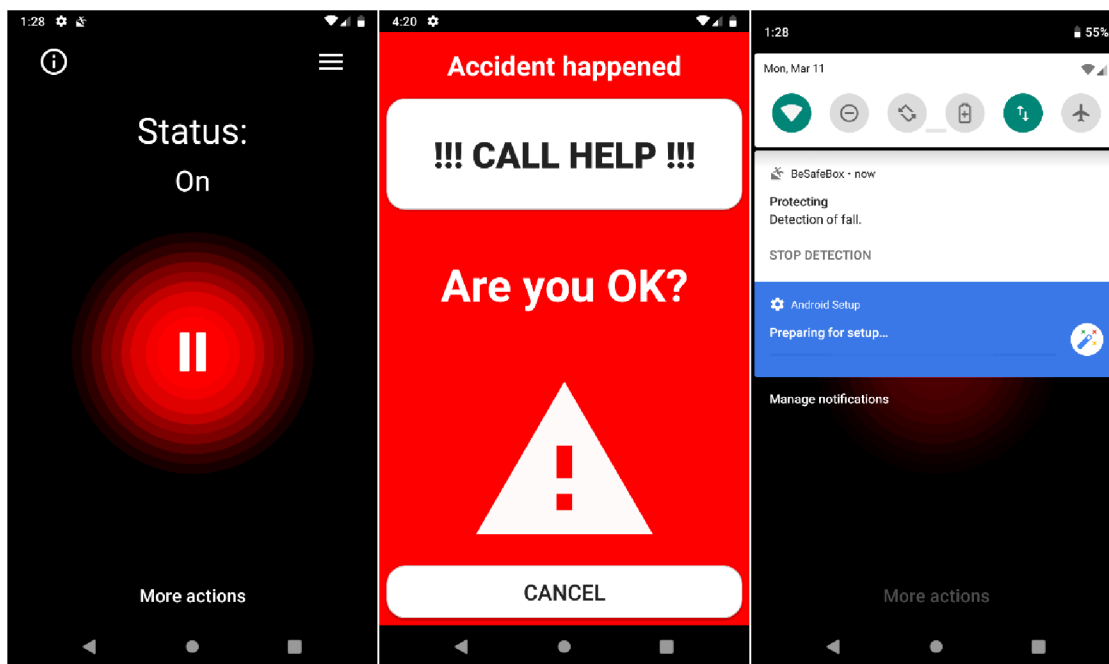
Hlavná aktivita obsahuje jedno dominantné tlačidlo, ktoré spúšťa a vypína službu na pozadí prostredníctvom dlhého stlačenia, aby užívateľ náhodne pri odchode z aplikácie nezapríčinil jej vypnutie. V dolnej časti aplikácie sa nachádza ďalšie tlačidlo, ktoré vyvolá vytvorenie dialógového okna s tlačidlami na rýchlu výmenu detektora, prepnutie automatického režimu pre voľbu detektora. Táto aktivita je spomenutá pri popise spotreby batérie. Ostatné tlačidlá priamo ovplyvňujú prebiehajúcu službu alebo ich voľba je uložená pre budúci proces.

Ďalej hlavná aktivita obsahuje tlačidlo pre prístup popisu licencií knižníc tretích strán. V tejto aktivite sú spomenutí autori knižníc, ktorí nie sú natívne v bežnom Android prostredí alebo materiály/grafiky, ktoré neboli vytvorené nami. Tieto licencie je povinné zverejniť v danej aplikácii, pretože je to zvyčajne jediné pravidlo, ktoré pri splnení umožňuje danú knižnicu voľne využívať. Patrí tu napr. knižnica umožňujúca kontrolu telefónnych čísel pre národnosti celého sveta s voľbou predvoľby, modul na odosielanie emailov bez prítomnosti užívateľa, pokročilejší systém ukladania nastavení aplikácie a množstvo grafických zdrojov. Zároveň sú tu obsiahnuté kontakty alebo licencie, prostredníctvom ktorých sa dajú vyhľadať jednotlivé knižnice.

V hlavnom menu je možné nájsť aktivitu s tlačidlom grafu, ktorá slúži na zber štatistiky počas detekcií. Počas aktívnej služby dochádza ku kolekcií jednotlivých dejov počas behu a zaratúvajú sa do pamäte aplikácie. Môžeme tu nájsť celkový čas služby, aktívny čas, počet indikácií na nehody, počet zamietnutých indikácií a počet detekcií.

Užívateľ si môže potom nastaviť možnosť alarmu v prípade detekcie. Sú na výber tri možnosti a to žiaden, alarm prostredníctvom notifikácie, ktorý je možné použiť na interaktívny zber údajov, a plný alarm, ktorý odosiela informácie o detekcii na dané telefónne číslo a mail. Pri voľbe tejto možnosti užívateľ zadá svoje meno, mail a telefónne číslo kontaktnej osoby. Pri nevyplnení sa táto možnosť nezapamätá a pri odchode z aktivity je užívateľ vyzvaný, či si je tým istý a je možný návrat. Po zadaní týchto možností je užívateľ schopný si vyskúšať poslať mail a SMS správu kontaktnej osobe pre overenie správnosti údajov a hlavne pre oboznámenie kontaktnej osoby s formou správy. Správa v maili a SMS obsahujú zadané meno s typom detekcie a link na Google mapy s pripnutými súradnicami, kde sa nehoda odohrala. Taktiež je možné si vyskúšať ako znie alarm. Pre správnu funkciu alarmu je potrebné, aby užívateľ dal práva aplikácii meniť nastavenia notifikácií. Aplikácii to umožňuje ignorovať prípadný tichý režim pre vyvolanie alarmu.

Pri zapnutí alarmu dochádza k zmene nastavení zvuku, pričom sa hlasitosť médií zmení na maximum daného zariadenia, aby bol zanedbaný fakt, keby užívateľ zapnutý tichý režim. Táto jediná aktivita má po vyvolaní taktiež zapnúť obrazovku a aj napriek uzamknutiu sa zobrazíť. Po zobrazení sa obrazovka rozsvieti a uzamkne sa do rozsvieteného stavu po dobu života danej aktivity. Aktivita taktiež začne blikať bleskom fotoaparátu, vibrovať a zároveň vydávať upozorňovací tón. Tento cyklus sa po istej časovej dobe zmení na nepríjemný piskľavý tón s prerušovaním. Na obrazovke sú umiestnené dve tlačidlá. Jedno slúži na vypnutie alarmu, ktoré ukončí celú aktivitu a druhé slúži na otvorenie núdzového číselníka podľa nastavenia mobilu. Kvôli experimentálnej forme aplikácie nedochádza k automatickému hovoru, aby nedošlo k náhodnému hovoru. Užívateľ má na reakciu 60 sekúnd, pričom po danej dobe sú odoslané údaje kontaktnej osobe a telefón ďalej pokračuje v upozorňovaní. V prípade nízkej kapacity batérie dôjde k prerušeniu aktivity po stanovenom časovom úseku, pre ušetrenie batérie pre prípadný núdzový hovor.



Obr. 4.1 Ukážka aplikácie – naľavo hlavné menu, v strede alarm, napravo notifikácia služby na pozadí

4.2 Fungovanie služby detekcie

Celá detekcia je sprostredkovaná cez službu, ktorá pracuje na pozadí systému a obsahuje všetky potrebné objekty pre detekciu pádu alebo autohavárie. Celý životný cyklus služby začína podnetom na štart od nejakej aktivity. V tomto prípade od hlavnej aktivity. Pri štarte služby dochádza k vytvoreniu inštancií všetkých potrebných objektov pre fungovanie detekcie. Služba môže ukončiť samú seba podľa potreby, môže ju ukončiť iná aktivita alebo ju ukončí samotný systém, keď potrebuje uvoľniť výpočtový výkon pre iné procesy a vyhodnotí službu ako najmenej potrebnú. Pri ukončení služby dochádza k zrušeniu všetkých relácií objektov, ktoré sú zahodené a uvoľnia pamäť ďalším objektom.

Služba je ďalej schopná prijímať príkazy od hlavnej aktivity ako je napríklad zmena senzoru alebo spôsobu zmeny stavu. Obdobne tomu tak je aj pri zmenách v nastaveniach. Celú detekciu riadi jeden hlavný objekt uložený v službe a to „ModelManager“, ktorý zastrešuje registráciu senzorov, dočasnú pamäť, detekciu potenciálnych úsekov a ich klasifikáciu. Každý z týchto dielov je tvorený samotným objektom, ktorý je možné dynamicky meniť vzhľadom na nastavenia a povahu merania, preto rozširovanie aplikácie by nepredstavovalo až taký veľký problém.

V prvom rade dochádza k vytvoreniu detektora vzhľadom na nastavenia. Môže byť vytvorený detektor na detekciu autohavárie alebo pádu. Dôvodom tvorby detektora sú atribúty, ktoré so sebou nosia. Každý detektor nosí v sebe informácie, aké senzory potrebuje a ako je schopný detekovať inú činnosť, než na ktorú je prispôbený. Taktiež každý detektor dedí vlastnosti nadriadeného abstraktného objektu, teda ich metódy sú

identické a v rámci programu je možné narábať s každým detektorom rovnako. Každý obsahuje metódu na posun výstupov zo senzorov, GPS, metódu na pokročilejšiu klasifikáciu a metódu na zmenu na ďalší detektor. Každý detektor má zároveň prístup k ďalšiemu výpočtovému vláknu, kde môže vytvárať časované požiadavky na vykonanie ďalších procesov ako je klasifikácia, pričom nebude narušovať proces vzorkovania.

Po inštancii senzoru, dochádza k tvorbe dočasného úložiska, „DataManagera“, na uloženie objektov vzoriek a GPS lokácie. Tento objekt v závislosti od zistených hodnôt frekvencie a potrebných senzorov vopred vykoná alokáciu dočasnej pamäte pre výstupy. Pre zistenie týchto parametrov je využitá dočasná služba pri prvom spustení aplikácie. Jej hlavnou úlohou je odhadnúť, koľko vzoriek približne sú schopné senzory vyprodukovať za plného behu, keď ešte užívateľ využíva svoje zariadenie. Frekvencia senzorov je premenlivá a závisí od stavu zariadenia, pričom najvyššia frekvencia je zvyčajne, keď je zariadenie aktuálne používané. Potrebné kapacity sú odvodené od najkratšieho času potrebného pre obnovenie hodnoty senzoru. Táto hodnota nemusí byť prítomná, preto pri tejto možnosti služba spustí senzory na 10s a približne určí ich aktivitu. „DataManager“ vie podľa požiadavky vytvoriť nemodifikovateľnú kópiu aktuálneho stavu úložiska, ktorá je využitá na klasifikáciu dejov, pričom neprerušene môže ďalej pokračovať v ukladaní ďalších vzoriek.

Po vytvorení dočasnej pamäte dochádza k registrácii senzorov a GPS senzorov. Podľa detektoru dochádza k výberu senzorov, ktoré budú zaregistrované. Po registrácii senzorov, ich výstupy sú distribuované do detektoru a dočasnej pamäte na spracovanie. Ako bolo už spomenuté, vzorkovacia frekvencia sa nedá pevne stanoviť, iba nastaviť maximálnu odozvu a systém si ju následne upravuje podľa potreby v záujme úspory batérie. Niektoré mobily zdvihnú vzorkovaciu frekvenciu len pri prudkých pohyboch, čo je možné odsledovať u rôznych výrobcov, inak ju držia na požadovanom limite.

Spomenuté objekty sú dynamické v závislosti od dostupného detektoru. Pri zmene detektoru je postačujúce tieto senzory inštancovať znova pomocou nastavení z nového detektoru počas existencie už vytvorenej služby. Toto riešenie zabezpečuje modularitu celej aplikácie a schopnosti jej jednoduchého budúceho rozširovania a údržby. Po hotovej inicializácii, jednotlivé vzorky zo senzorov prestupujú cez detektor do dočasného úložiska a sú pripravené pre klasifikáciu v prípade podozrivej aktivity.

Návrh detekcie je identický s návrhom v druhej kapitole. Pri detekcii pádu slúži ako hlavný spúšťač zvýšenie akcelerácie nad 3g. Pri prevýšení tejto magnitúdy, sa čaká nasledujúcich 5 sekúnd na ďalšiu aktivitu nad 3g. Ak je prítomná, senzor čaká na ďalší príchod a neinicializuje sa žiadna klasifikácia. Pokiaľ ale magnitúda presiahne 3g a po dobu 5 sekúnd nenastane žiadna aktivita nad 3g, dôjde ku kontrole menšej aktivity. Pri simuláciách bolo dobrovoľníkom nakázané, aby sa uvoľnili po vykonaní danej činnosti. Z úsekov, ktoré boli nasnímané po páde je možné zistiť, že priemerná aktivita po páde ak človek bezvládne leží sa pohybuje pod hodnotou $10,1 \text{ m/s}^2$. Ak sa menšia aktivita nepotvrdí, dochádza k pokročilejšej klasifikácii podľa vytvorenej neurónovej siete. Keď

neurónová sieť prekročí pravdepodobnosť detekcie inej udalosti napríklad pod 70 % dôjde k vyvolaniu aktivity alarmu. Pri falošnom alarme je nutné manuálne spustenie služby, aby sme predišli niekoľkonásobnému alarmu. Tieto parametre je možné meniť v nastaveniach aplikácie.

Detekcia autohavárie funguje na princípe vyhľadávania akcelerácie s prahom nad 4g. Pri pohybe auta, teda pri zmenách polohy, dochádza k vyčkaniu na zmenu polohy. Tento čas činí trojnásobok stanovenia aktualizácie zmeny polohy, ktorý je 10 sekúnd. Pri prítomnosti zrýchlenia a neprítomnosti zmeny polohy dochádza ku kontrole pádu mobilu a prítomnosti menšej aktivity vozidla.

V prvom rade je zistené, či sa nejednalo o pád mobilu v aute. Pád je charakteristický svojím poklesom akcelerácie takmer na nulu, pričom v extrémnom páde je dráha padajúceho mobilu maximálne 1 meter, čo približne odpovedá 0,5 s voľného pádu. Keď sa po dobu 0,5 s a menej objavuje zrýchlenie, ktoré je menšie ako 2 m/s^2 , pokračuje sa na ďalšiu podmienku. Tým je myslené jemné chvenie, ktoré je spôsobené jazdou, ktorá je symbolická rozptylom vyšším ako 0,01. Ak aktivita nie je prítomná, vyhodnotí sa to ako autonehoda a je najskôr vyvolaný tichý alarm, aby sme prípadne nevyrušili vodiča pri falošnej detekcii. Po 30 s tento alarm prechádza do plného alarmu, pričom po ďalších 30 s sú odoslané informácie o nehode danému kontaktu.

4.3 Doze režim

V prvom rade je nutné podotknúť, že Android aplikácie nie sú vhodným riešením dlhodobých detekcií. Aplikácie pracujúce na pozadí sú pomerne vysokým odoberateľom energie, pretože kvôli nim musí procesor ostať v aktívnom stave. Procesor je druhý najvyšší odoberateľ energie po displeji. Preto bol v Android 6.0 predstavený Doze režim, ktorý sa aktivuje po splnení troch podmienok : vypnutá obrazovka, mobil nie je pripojený k nabíjačke a je v stacionárnej pozícii. Tento režim nie je prítomný vo verziách Androidu nižších ako 6.0, takže tieto obmedzenia sa ich netýkajú. Pri jeho aplikácii dochádza k pozastaveniu činnosti všetkých aplikácií vrátane všetkých načasovaných operácií v danom čase, odopreniu internetového pripojenia a iných zdrojov. Pri niektorých výrobcoch dochádza k vypnutiu všetkých senzorov. Zariadenie sa ale pravidelne budí z Doze režimu v podobe krátkych maintenance window, okna na vykonanie naplánovaných prác aplikácie ako je aktualizácia mailov, notifikácií a iných upozornení [8]. Pre väčšinu užívateľov je to prospešné, pretože aplikácie sú redukované systémom samotným a užívateľ nemusí vypínať aplikácie na pozadí sám. Problém nastáva ale pre vývojárov, pretože je nutné vykonať celú sériu krokov a testovaní aplikácie na to, aby i napriek Doze režimu bola schopná aplikácia pracovať ďalej. Dokonca pri uverejnení aplikácie na Google Play, aplikácia musí splniť určité podmienky, pokiaľ chce tento režim obchádzať. Inak môže byť z ponuky Google Play odobraná, pretože aplikácia môže vyvolať nekontrolované vybíjanie batérie.

Väčšina aplikácii si nárok na obchádzanie Doze režimu stanovuje len na určitý časový limit, aby dokončila napríklad synchronizáciu dát cez internet. Táto stratégia je najviac preferovaná medzi vývojármi. Nanešťastie, vytvorená aplikácia potrebuje bežať bez zastávky, pretože nikdy nie je možné predpokladať pád človeka alebo autonehodu v daný časový rámec. Môžeme si zobrať ako príklad sediaceho človeka, ktorý sa rozhodne postaviť. Mobil je staticky s vypnutou obrazovkou bez pripojenia k elektrickej sieti, teda sa nachádza v Doze režime. Doze sa bude riadiť píkmí v akcelerácii, ktoré ho môžu zobudiť. V Androide ide o špecifický programový senzor „Significant motion sensor“ alebo senzor významného pohybu, ktorý sa zobudí pri detekcii akcelerácie, ktorá mohla vyprovokovať zmenu pohybového stavu. V zdrojovom kóde je popísané, že tolerancia je maximálne 10 sekúnd a pohyb musí byť detekovaný, ale pád pri postavení sa zo stoličky sa môže udiť aj v oveľa menšom okne.

Preto sa nemôžeme spoliehať na prebudenie aplikácie v potrebný čas a potrebujeme, aby aplikácia pracovala i v tomto režime. Zároveň aplikáciu musíme ochrániť pred inými systémovými optimalizáciami, ktoré kompletne vypínajú procesy na pozadí, pričom sa dajú obnoviť jedine užívateľom. Tieto zvyčajne prichádzajú od samotných výrobcov a niekedy sa nedajú ovplyvniť.

V prvom rade, aby služba bola ochránená pred náhodným vypnutím systémom napr. kvôli nedostatku dočasnej pamäte, je služba reprezentovaná neodstrániteľnou notifikáciou. Tento druh služby sa nazýva „ForegroundService“ a zabezpečuje najvyššiu prioritu pre systém. To však nepredstavuje absolútnu protekciu služby a môže dôjsť k vypnutiu služby, i keď je to málo pravdepodobné. Táto notifikácia je zároveň využitá na zobrazenie informácie o aktuálnom využívanom detektore. Služba navyše nie je prepojená priamo s aktivitami, pretože toto zvyšuje pravdepodobnosť zrušenia služby. Pre komunikáciu využíva systém intencií, celosystémové odosielanie malých správ, ktoré zabezpečuje ich nezávislosť.

Prítomnosť systémovej optimalizácie od výrobcu sa nedá automaticky zabezpečiť, pretože každý výrobca vyžaduje inú prístupovú sekvenciu pre otvorenie týchto nastavení. Pre neschopnosť otestovať každé zariadenie od každého výrobcu, toto zabezpečenie nebude v rámci konceptu riešené a pri testovaní bude musieť užívateľ túto výnimku nastaviť manuálne, ak je obdobný systém prítomný.

Čo je ale možné nastaviť je výnimka pre Doze režim, o ktorú je užívateľ požiadaný hneď po otvorení hlavnej aktivity s povoleniami k prístupu ku kontaktom v mobile a využitiu SMS správ. Je nutné, aby všetky tieto povolenia boli udelené, pretože aplikácia bez nich nebude schopná pracovať alebo jej funkcionality bude vysoko obmedzená. Udelením týchto povolení bude aplikácia stále podmanená Doze režimu. Toto je žiadané v rámci všetkých aktivít, pretože tie nie sú potrebné pre detekciu. Jedine služba detekcie si buduje nárok na procesor, a kým si ten nárok si bude držať, tak aplikácia bude fungovať i napriek Doze režimu. Tento nárok si udržiava vo forme objektu

„WakeLock“. Keď má aplikácia všetky povolenia a vytvorí si „WakeLock“ objekt v aktívnom stave, je vynechaná z regulácie režimu Doze a môže bežať nelimitovane.

Tu nastáva jedna veľká nevýhoda, a to nárast využitia batérie, pretože aktivácia procesoru vedie k zvýšenej aktivite senzorov, GPS lokácie a obdobne. Táto záťaž môže prevýšiť aj športové aplikácie v stave merania i za pripojenia Bluetooth zariadenia pri dlhom užívaní. Preto bude cieľom najvyššia miera optimalizácie celého spracovania výstupov zo senzorov a obmedzenie behu služby na čo najvyššiu mieru.

4.4 Optimalizácia služby detekcie

Najideálnejším spôsobom ako znížiť spotrebu je vzdať sa nároku na procesor. Podmienky, za ktorých ale by bolo vhodné sa ho vzdať, sú rôzne. Pri vypnutí senzorov a zároveň umožnení Doze režimu prevziať plnú kontrolu je možné, aby sa spotreba znížila na úroveň pod 10 mAh, čo je menšia spotreba ako iné bežne vyskytujúce aplikácie sa v mobile. Všetky nasledujúce nastavenia sú voliteľné.

Ako finálne riešenie šetrenia batérie bola implementácia Android služby pre detekciu aktuálnej aktivity človeka. Táto služba je obdobne založená na programovom senzore významného pohybu, ale taktiež je schopná odlíšiť stacionárneho, chodiaceho človeka alebo človeka nachádzajúceho v nejakom vozidle. Ide o službu v systéme, tak nedochádza k rozsiahlejšej spotrebe batérie. Zmena pohybového stavu je zistená pomocou prichádzajúcich intencii pri danej zmene. Táto intencia nosí v sebe informáciu o danom pohybe a podľa toho sme schopní vymeniť detektor, pričom keď je detekovaný pokojový stav, aplikácia prechádza do spiacieho režimu.

Ďalším spôsobom ako ušetriť batériu sa nachádza v použití senzoru vzdialenosti. Jeho hlavnou úlohou je zisťovať, či sa zariadenie nachádza vo vrecku, taške alebo v otvorenom priestore. Preto, keď sa mobil nachádza vo voľnom priestore, môžeme uvažovať, že mobil nie je v kontakte s užívateľom a nachádza sa mimo. Detekcia je následne zbytočná a detektor môže byť odstavený. Na úkor toho sa detekcia vypne, i keď mobil užívateľ má práve v rukách, teda pri páde pri aktuálnom používaní zariadenia nebude registrované detektorom. Treba brať do úvahy, že toto riešenie ponúka oveľa vyššiu mieru automatizácie a obmedzuje detekciu len na čas, keď je mobil aktuálne nosený užívateľom. Pri kombinácii s rozpoznaním aktivity človeka sa aplikácia pri vložení vrecka aktivuje na 5 min. Pokiaľ nedochádza k aktivite po 5 min, aplikácia prechádza opäť do spiacieho režimu.

Ako ďalšia možnosť bola implementovaná možnosť špecifikovať detekciu pádu pri pripojení na wi-fi. Keď sa zariadenie úspešne pripojí k wi-fi sieti, dochádza k odpojeniu automatického režimu ak je pripojený, teda vypnutiu GPS, a dochádza k pevnému pripojeniu detekcie pádu. Táto možnosť hlavne vyplýva z dnešnej techniky dynamického pripojovania a odpojovania wi-fi v závislosti od lokalizácie pripojenia. Zariadenie si pamätá, kedy sa nachádza v blízkosti známeho pripojenia a podľa toho je wi-fi modul zapnutý. Táto informácia je tiež využitá k špecifikácii danej detekcie.

Vzhľadom na povahu vysokej spotreby pri aplikácii bola taktiež pridaná možnosť vypnutia detekcie pri stave nízkej batérie. Systém vyšle signál aplikáciám, keď sa úroveň batérie zníži na kritickú úroveň, ktorá sa vzhľadom na výrobcu pohybuje pri 10 – 20 %. Pri tomto signále služba taktiež vypne všetky procesy, ale ostane pasívna na pozadí. Pri dobíí batérie začne služba znova normálne pracovať. Tieto možnosti sú dostupné v nastaveniach a užívateľ si ich môže nastaviť podľa potreby, pričom môže nastaviť aj neobmedzený chod aplikácie.

Čo sa týka optimalizácie z pohľadu kódu, je ako najväčší pokrok možné brať novú knižnicu na neurónové siete určené pre mobilné zariadenia, Tensorflow Lite. Táto knižnica sa postupne dotvára a čaká ju ešte veľa aktualizácií. Predošlú knižnicu, Tensorflow mobile, je treba chápať ako predchodcu, z ktorého bola vytvorená viac optimalizovaná, flexibilná a zjednodušená verzia. Vytvorená aplikácia s Tensorflow mobile mala po inštalácii vyše 80 MB, pričom s Tensorflow Lite je to len necelých 20 MB, čo sa prejavuje na efektívnosti celej aplikácie. Zároveň to znamená aj menšie vyžitie operačnej pamäte a menšiu spotrebu batérie. Vytvorená sieť je pomerne malá, tak hardwareová akcelerácia nie je až tak potrebná, pričom zatiaľ nie je ani dostupná. Prácu neurónovej siete odvádza plne procesor, ale testy prevedené vývojármi Tensorflow za využitia integrovanej grafickej karty v mobile ukazujú, že dochádza k zvýšeniu efektívnosti aj o 80 % oproti procesoru. Táto funkcia je dostupná len v experimentálnom móde a do plnej prevádzky sa dostane podľa predpokladov do konca jari 2019 [25].

5. VYHODNOTENIE APLIKÁCIE

Pre otestovanie aplikácie boli vybrané rôzne metódy na overenie správnej funkcionality. Aplikácia bola priebežne testovaná počas výstavby. Boli postavené rôzne testy, ktoré testujú stabilitu kódu.

Hlavným komponentom testovania sú dáta, ktoré boli zozbierané pomocou dobrovoľníkov počas ich bežného dňa. Ukázalo sa, že model preferujúci iné udalosti oproti pádom je viac odolný voči okolitým rušeniam a viac špecifický voči samotným pádom bez významného zníženia schopnosti ich detekcie. Táto neurónová sieť bola implementovaná do aplikácie. Odolnosť voči iným udalostiam je rozobraná na strane 33, kedy chybovosť dosahovala pri priebežných testoch 5 – 10 % bez prítomnosti optimalizácii a obmedzenia času fungovania aplikácie. Tieto mechanizmy, ako aktivita aplikácie len vo vrecku alebo len počas aktivity užívateľa, znižujú chybovosť.

Bolo vyzbieraných ďalších 104 pádov pre otestovanie funkcionality aplikácie. Bol vytvorený inštrumentálny test, test využívajúci algoritmy aplikácie bez potreby užívateľa, ktorý načítal všetky tieto pády a klasifikoval ich. Podľa testu, bolo správne detekovaných 79/104 pádov, čo je 76 %. Pri snímaní boli použité 4 rôzne mobily – Huawei P9, Sony Xperia Z1, Z5 a Lenovo P70 – každý po 26 pádov. Ukázalo sa, že vysoký vplyv má umiestnenie mobilu v danom odeve. Voľné oblečenie v kombinácii umiestnenia mobilu na opačnej strane od miesta dopadu, spôsobí význačný útlm priebehu, čo skreslí daný priebeh a NN to radšej klasifikuje ako inú udalosť, než pád. Je nutné zobrať do úvahy aj fakt, že všetky simulácie prebiehali na mäkkom povrchu. Tým pádom, keby sme pridali stratené spomalenie spôsobené mäkkým povrchom, daný priebeh by mohol nadobudnúť správnych hladín a NN by ho klasifikovala ako pád. Toto môže vyústiť v realite do vyššej úspešnosti algoritmu.

Test detekcie autohavárie bol odskúšaný počas reálnej jazdy autom. Inštrumentovaný test nebolo možné vykonať z dôvodu komplikovanej implementácie polohy vozidla, ktorá by si vyžadovala oddelenú tvorbu .gpx súboru alebo vytvorenie úplne novej implementácie polohy. Aplikácia bola pripojená na vývojárske prostredie, kedy sa umelo navodilo zrýchlenie siahajúce nad 4g, pričom bola odignorovaná podmienka pádu mobilu. Aplikácia následne bola otestovaná pre dva prípady. V prípade dostupnosti GPS, vždy došlo k odstráneniu požiadavky na kontrolu jazdy auta, pretože došlo k dostatočnej zmene polohy vozidla, ktorá zabránila alarmu. V druhom prípade bol mobil uložený do priechodu v dverách, kedy došlo k obmedzeniu GPS signálu a priemerná presnosť sa zvýšila až na okruh nad 100 m a došlo k pozastaveniu aktualizácii. Alarm bol vylúčený na základe rozptylu akcelerácie, pretože auto bolo v pohybe ďalej. Počas oboch testov bola pravdepodobnosť prítomnosti mobilu vo vozidle nad 80 %, teda nedošlo k interferencii zo strany detekcie inej aktivity. V prípade straty GPS signálu je detekcia pozastavená, pretože by sa špecifickosť senzoru značne narušila.

5.1 Interné testovanie pomocou Google Play

Aplikácia bola po sfinalizovaní úprav nahratá na Google Play do interného testovania, čo znamená, že len zvolení užívateľia majú schopnosť si aplikáciu stiahnuť. Aplikáciu si stiahlo 12 dobrovoľníkov. Aplikácia je kompatibilná od Android 4.4.4 KitKat po Android 9.0 Pie, čo predstavuje 96 % existujúcich Android zariadení na svete [8]. Užívateľ taktiež nepotrebuje žiadne ďalšie senzory okrem akcelerometru a GPS, ktoré obsahuje každý mobil.

Toto testovanie prináša veľa výhod, ktoré sú hlavne ľahká distribúcia samotnej aplikácie a jej opráv. Pri nahraní novej verzie je aplikácia schopná sa dostať do 24h ku všetkým užívateľom do všetkých štátov. Pri dosiahnutí stabilnej a funkčnej verzie by ju bolo možné hneď publikovať pre širokú verejnosť.

Využívanie Google Play má ale aj niekoľko úskalí v podobe ochrany užívateľských dát. Pre zverejnenie bolo potrebné zbaviť aplikáciu schopnosti interagovať s kontaktami v mobile a odosielať SMS. Tieto povolenia viedli v minulosti k veľkému množstvu podvodov, a preto, hlavne odosielanie SMS, je vysoko kontrolované. Autor aplikácie je nútený vytvoriť veľké množstvo ukážok, čo aplikácia dokáže, a garantovať či použitie týchto práv je oprávnené z pohľadu pravidiel Google Play. Obdobne bolo zrušené aj zasielanie mailu, ktoré porušuje práva o ochrane súkromia. Pre ušetrenie týchto právnych procesov boli tieto implementácie otestované vo vývojárskom prostredí, ale v balíku pre Google Play boli odstránené. Ich spätná implementácia je dosiahnuteľná zopár krokmi. Jediné povolenia, ktoré ostali, sú požiadavka na získanie polohy a oprávnenie na ignorovanie optimalizácii batérie, čo umožní aplikácii si kontrolovať svoj životný cyklus.

Jediné údaje, ktoré odchádzajú z aplikácie sú chybové hlášky v prípade chyby programu. Tieto hlášky sú zdieľané pomocou služby Firebase, ktorá je vytvorená Googlom. Firebase vykonáva širokú škálu funkcií pre dozor využívania aplikácie užívateľmi. Kontroluje koľko času kde strávili, optimalizáciu aplikácie a iné udalosti, zároveň ponúka databázové riešenia, autentifikáciu užívateľov, odosielanie správ a množstvo iných funkcií. Hlavnou funkciou, ktorá sa aktívne používa v aplikácii, je dokumentácia chýb aplikácie. Pri narazení na chybu aplikácie pred jej pádom Firebase odošle hlášku na servery, ktorá je sprostredkovaná tvorcovi s ďalšími informáciami o zariadení. Týmito riešeniami dochádza k nezávislosti testerov od vývojárov. Pri distribúcii aplikácie došlo hneď k niekoľkým chybám, ktoré boli v priebehu času opravené v jednotlivých aktualizáciách.

S Firebase prichádza už povinnosť zverejniť pravidlá o ochrane súkromia a pravidlá o používanie aplikácie. Firebase vystupuje ako nezávislý subjekt a je nutné sa voči nemu vyhradiť v týchto podmienkach. Pri absencii týchto podmienok môže byť aplikácia z Google Play odstránená za nedostatok transparentnosti. Z tohto dôvodu došlo ku generácii jednoduchého textu, ktorý popisuje ochranu súkromných dát a je v ňom popísané, ktoré konkrétne údaje sa preberajú od užívateľa. Užívateľ musí s týmito

podmienkami pri inštalácii súhlasiť. Pri nesúhlase nemôže užívateľ používať aplikáciu. Tieto ustanovenia sú dostupné vo webovej podobe na hlavnej stránke aplikácie na Google Play , ale aj v samotnej aplikácii.

Aplikácia bola následne distribuovaná medzi zvolených dobrovoľníkov. Dobrovoľníci sú zostavení hlavne z mladých ľudí, u ktorých pád alebo autonehodu nepredpokladáme, teda alarm a odosielanie údajov o polohe prostredníctvom SMS a mailu nie sú v testovacom režime potrebné. Postačíme si, ak dá aplikácia svoju detekciu najavo iba cez notifikáciu. Na túto notifikáciu sa dá kliknúť, pričom dôjde k vyvolaniu novej aktivity, kde užívateľ môže zvoliť správnosť detekcie. Pri falošnej detekcii pádu užívateľ zvolí možnosť zlej klasifikácie a dôjde k vyvolaniu dialógu, kde užívateľ môže zvoliť najpodobnejší pohyb. Všetky tieto štatistiky sú ukladané do lokálnej pamäti mobilu a nie sú odosielané cez internet. Celá štatistika je reprezentovaná samostatnou aktivitou, kde sa nachádza celkový čas používania, aktívny čas senzoru, jednotlivé časy jednotlivých senzorov a počty detekcií. Tieto údaje boli následne zozbierané od účastníkov s ich dojmami o fungovaní aplikácie. Jednotlivé štatistiky sú reprezentované v nasledujúcej tabuľke.

Tab. 5.1 Využitie aplikácie medzi 12 dobrovoľníkmi

Celkový čas	540 h
Aktívny čas	120 h
Čas detekcie pádu	95 h
Čas detekcie autohavárie	25 h
Počet indikácií na pád	13 151
Počet indikácií na autohaváriu	85
Počet volaní na NN	573
Počet kritických detekcií pádu	94
Počet kritických detekcií autonehody	0
Počet reálnych pádov/autonehôd	0

Najväčší problém vytvorila samotná detekcia činnosti človeka, ktorá sa niekedy ukázala ako nespoľahlivá. Jej hlavnou výhodou je úspora batérie, keďže sa jedná o interný proces v systéme. V niekoľkých prípadoch vznikol problém, že ani po veľmi dlhej dobe chôdze sa detekcia neohlásila. Vykonala tak až v intervale 5 min. Vo výsledku potom aplikácia ostala buď v spánkovom režime alebo v poslednej nastavenej detekcii, i keď to už nebolo potrebné. Tento problém je závislý od verzie a výrobcu mobilu, keďže aj tento samotný klient na detekciu aktivity je obmedzený šetriacimi režimami v mobile a jeho senzormi. Čo mohlo mať vplyv na toto rozhodnutie je umiestnenie mobilu v rámci užívateľa, keďže pravdepodobnosť chôdze nemusela byť dostatočne vysoká. Existuje ale riešenie v podobe odberu pravdepodobnosti jednotlivých aktivít, ako sa to deje pri detekcii autohavárie, teda by sme nemuseli čakať na rozhodnutie systému o zmene stavu.

Táto výhoda by ale stála systém oveľa viac energie ako je tomu teraz, pretože takto je systém nútený pracovať, aj keď by inak nepotreboval. Pri stacionárnej pozícii ale mobil i pri tejto voľbe riešenia po dlhšej dobe prestane odpovedať. Nové pravdepodobnosti by boli generované až keď by sa mobil nadmerne pohol, čo by prebudilo systém. Pri detekcii jazdy auta alebo stacionárneho stavu sa tento problém neprejavil.

Pri užívateľoch, ktorí nedisponujú autom alebo nechcú používať detekciu autohavárie je odporúčané, aby si vypli automatickú detekciu jazdy, pretože napríklad pri mestskej hromadnej doprave dochádza ku konfliktu výberu danej aktivity. Pri státi v danom prostriedku má aplikácia tendenciu ostať v režime detekcie pádu, ale keď užívateľ sedí, dochádza k detekcii autohavárie. Preto je to na užívateľovi, aký mód viac preferuje pre svoje potreby.

NN na pád je trébovaná na umiestnenie mobilu vo vrecku, teda kým ju dobrovoľníci mali vo vrecku, tak nedochádzalo k falošným detekciám. Poistka na umiestnenie vo vrecku pomocou senzoru na vzdialenosť bola vzhľadom na neurónovú sieť plne efektívna. Pri vypnutí tejto poistky nedochádzalo k veľkému vplyvu vzhľadom na falošné detekcie. Problematické boli len pohyby, ktoré pripomínali pohyb mobilu pri páde. Teda prudké spomalenie, ktoré je spojené so zmenou uhla mobilu, ako je to napríklad pri rýchlom zdvíhaní hovoru mobilu priamo z vrečka. Toto je jeden z mála prípadov, kedy dochádza k falošnej detekcii, čo je ale pochopiteľné, keďže na toto NN nie je trébovaná a najpodobnejšie charakteristiky má práve pád. Zle detekcie boli, keď užívateľ mal smartphone uložený v bunde, ktorú si dal do rúk, alebo v taške, ktorá bola položená na zem alebo rameno.

Detekcia autohavárie je svojim spôsobom robustné riešenie, pretože i napriek indikáciám na autonehodu nedošlo ani raz k vyvolaniu alarmu. Jednotlivé podmienky sa osvedčili v správnej funkcionalite v reálnom prípade, ale nie z dostatočných zdrojov testovania. V najhoršom prípade pri autonehode môže dôjsť k fatálnemu narušeniu funkcie mobilu a detekcia autohavárie sa nezrealizuje.

Z pohľadu stavby aplikácie dochádzalo k chybám rôzneho charakteru, ktoré prináša objektovo orientované programovanie, ako neexistujúce objekty alebo zlé indexovanie. Tieto problémy boli väčšina zachytené pomocou Firebase Crashlytics služby, ktoré pomocou aktualizácii boli vyriešené. To ale nevylučuje, že sa v budúcnosti nemôžu vyskytnúť ďalšie. Služba v prípade fatálnej chyby je nastavená ako permanentná, teda pri zlyhaní a ukončení hlavného procesu dochádza k jeho reštartu a jeho pokračovaniu. Jej ukončenie vykoná len užívateľ, ale po počiatočných opravách sa už taký druh chyby už nevyskytol. Aplikácia v priemere zaberá v dočasnej pamäti 20 MB, pričom jej fungovanie nezmenšuje interaktivitu systému.

Veľkosť spotreby batérie bola rôzna. Pri stacionárnom stave mobilu aplikácia vykazuje minimálnu spotrebu až žiadnu, čo nemá vplyv na užívateľa. Detekcia pádu a jej spotreba sa pohybovala medzi 20 – 40 mAh na hodinu prevádzky mobilu, čo môže mať dopad na dĺžku užívania mobilu. Čím dlhšie sa aplikácia udrží v spánkovom režime, tým

viac šetrí batériu. Spotreba nad 30 mAh je už kritická a môže mať vysoký vplyv na výdrž. Pre zlepšenie tejto spotreby by bolo nutné zmeniť štýl formovania služby.

Detekcia autohavárie spotrebuje ešte viac batérie. Spotreba sa pohybuje medzi 60 – 100 mAh, niekedy aj viac. Dôvod je jednoduchý, a to je vysoká miera vzorkovania GPS. Pre zníženie spotreby batérie by bolo potrebné dynamicky meniť vzorkovaciu frekvenciu GPS aby jej aktivita odpovedala aktuálnej rýchlosti. Pri vysokých rýchlostiach jej frekvencia je zbytočne vysoká a spotreba prekračuje rozumné hodnoty. Akcelerácia nepodlieha zložitej analýze, čo sa neprejavuje na spotrebe. Ďalším prispievateľom je klient na rozpoznanie aktivity, ktorý je nastavený na 30 s intervaly na zisťovanie prítomnosti chôdze, ktorá môže potlačiť falošnú detekciu. Táto spotreba sa ale dá jedine tolerovať v prípade, že je mobil pripojený a nabíjaný v aute. Zároveň, keď by mal užívateľ pustenú navigáciu, došlo by k zdieľaniu informácií o polohe, čo by nepredstavovalo navýšenie nárokov na systém.

Aplikácia z pohľadu svojho celkového dizajnu je uplatniteľná pre kohokoľvek a nie je vekovo obmedzená. Z funkcionality môžu vyťažiť ako mladá tak i staršia generácia. Pri jej tvorbe nebolo vytvorené žiadne obmedzenie pre danú skupinu ľudí. Android mobily sú rozšírené a nie každý človek má k dispozícii náramok alebo inteligentné hodinky, ktoré obsahujú detekciu pádu. Nie všetky autá obsahujú systém detekcie autonehody a automatické volanie o pomoc. Aplikácia ponúka jednoduché riešenie pre obidva problémy s istými obmedzeniami, ktoré boli rozobrané v priebehu celej práce. Veľkou výhodou je, že stačí mať bežný Android smartphome.

Dizajn aplikácie umožňuje jednoduché ovládanie len pomocou jedného tlačidla a aplikácia sa o všetko postará sama. Pre menší manuálny zásah je možné využiť vyvolanie rýchlych volieb, kde si užívateľ môže prepnúť detektor alebo vypnúť automatický výber detekcie pádu alebo autonehody. Pre pokročilejšiu konfiguráciu je nutné ísť do nastavení, kde je možné si pozmeniť senzitivitu detektoru a nastaviť šetriaci režim podľa potreby užívateľa.

V aplikácii existuje ešte veľký priestor na zlepšenie. Jej modularita umožňuje postaviť veľké množstvo rôznych detektorov a testovať ich funkčnosť v rámci systému. Hlavným úskalím sa stala občasná vysoká spotreba batérie. Pre jej zníženie by bolo potrebné zmeniť celé základy služby. Z klasifikácie by sme museli prejsť na predikciu časových okamihov, kedy by sa mohlo stať niečo podozrivé s čo najmenej náročným riešením. Pri nájdení takejto situácie, len vtedy aktivovať systém, využiť plne dostupné zdroje a znova prejsť do šetriaceho režimu. V našom prípade používame všetky zdroje, vždy počas aktivity detektoru, čo vedie niekedy k zbytočne prehnanej spotrebe. V Androide je nutné vykonávať úkony jedine vtedy, ak sú potrebné, aby sa vykonali, inak dochádza k spotrebe batérie, lebo na toto systém nie je uspôsobený.

ZÁVER

Výsledkom bakalárskej práce je funkčná stabilná aplikácia, ktorá bola otestovaná 12 dobrovoľníkmi po dobu 520 h. Aplikácia je použiteľná pre veľkú široké spektrum ľudí vďaka svojej schopnosti adaptovať svoju funkciu prostredníctvom integrovaného klienta na rozpoznanie aktivity človeka a rozpoznaníu kedy sa smartphone nachádza v odevu. Tieto vlastnosti zabezpečujú možnosť smartphonu vydržať i celý deň prevádzky.

Detekcia pádu založená na neurónovej sieti dosiahla počas tréovania úspešnosť 86 %. Bolo zozbieraných 300 hodín záznamu a ďalších 104 pádov od rôznych dobrovoľníkov. Pri záznamoch bežnej aktivity mala neurónová sieť úspešnosť 95 %, pri pádoch 76 %.

Aplikácia bola distribuovaná v rámci interného testu na Google Play, kde bolo zahrnutých 12 dobrovoľníkov. Pri používaní dobrovoľníkmi došlo z 573 inicializácií neurónovej siete, len k 94 falošným detekciám. Falošné detekcie boli spôsobené pohybmi pripomínajúce pád, keď mobil nebol v prostredí, na ktoré sa nebola neurónová sieť tréovaná. Použité bariéry pre prítomnosť aktivity človeka a detekcie pádu smartphonu sa osvedčili ako postačujúce z pohľadu filtrácie nedôležitých úsekov. Vzhľadom na absenciu reálneho pádu, nie je možné posúdiť schopnosť siete klasifikovať takúto udalosť.

Detekcia autonehody bola otestovaná lokálne pomocou 28 h záznamov, počas ktorých bolo prejdených približne 2500 km. Pri aplikácii skriptu napodobňujúceho funkciu algoritmu na detekciu autonehody, došlo k 61 indikáciám ale nepotvrdila sa ani jedna. Zároveň mobilná aplikácia bola v režime detekcie autonehody 25 h, pričom bolo vyvolaných 85 indikácií na autonehodu, ale ani jedna sa nepotvrdila.

Interným testovaním na Google Play bola taktiež otestovaná počiatočná stabilita programu a komplexnosť riešenia. Počas testovania došlo k niekoľkým nekonzistenciám spôsobenými životným cyklom aplikácie a systému, ktorý niekedy bránil plnej funkcionalite. Testovanie odhalilo aj niekoľko chýb programu, ktoré boli všetky odstránené. Výsledok testovania poukazuje, že vybraté riešenie by bolo po zopár úpravách, hlavne z pohľadu využitia zdrojov mobilu, vhodné pre bežné používanie. Aplikácia je zároveň vhodná pre kohokoľvek, bez nijakého obmedzenia. Pre plnohodnotné uverejnenie je nutné vykonať ďalší zber dát, vykonať ďalšie testy stability a spoľahlivosti daného riešenia a vyladiť optimalizáciu programu.

LITERATÚRA

- [1] WHO. *Falls* [online]. 2018 [vid. 2018-09-25]. Dostupné z: <http://www.who.int/news-room/fact-sheets/detail/falls>
- [2] OECD. *Road accidents (indicator)* [online]. 2018 [vid. 2018-11-18]. Dostupné z: [doi:10.1787/2fe1b899-en](https://doi.org/10.1787/2fe1b899-en)
- [3] KIM, Tai-hoon, Akingbehin KIUMI a Haeng-kon KIM. *Evaluating AAL Systems Through Competitive Benchmarking - Indoor Localization and Tracking* [online]. 2011. ISBN 978-3-642-17640-1. Dostupné z: [doi:10.1007/978-3-642-17641-8](https://doi.org/10.1007/978-3-642-17641-8)
- [4] CRADDOCK, Ger M. *Assistive technology : shaping the future ; AAATE '03*. B.m.: IOS Press, 2003. ISBN 9781586033736.
- [5] SANTOYO-RAMÓN, José Antonio, Eduardo CASILARI a José Manue CANO-GARCÍA. Analysis of a smartphone-based architecture with multiple mobility sensors for fall detection with supervised learning. *Sensors (Switzerland)* [online]. 2018, **18**(4). ISSN 14248220. Dostupné z: [doi:10.3390/s18041155](https://doi.org/10.3390/s18041155)
- [6] ABBATE, Stefano, Marco AVVENUTI, Francesco BONATESTA, Guglielmo COLA, Paolo CORSINI a Alessio VECCHIO. A smartphone-based fall detection system. *Pervasive and Mobile Computing* [online]. 2012, **8**(6), 883–899. ISSN 15741192. Dostupné z: [doi:10.1016/j.pmcj.2012.08.003](https://doi.org/10.1016/j.pmcj.2012.08.003)
- [7] FIGUEIREDO, Isabel N., Carlos LEAL, Luís PINTO, Jason BOLITO a André LEMOS. Exploring smartphone sensors for fall detection. *mUX: The Journal of Mobile User Experience* [online]. 2016, **5**(1), 2. Dostupné z: [doi:10.1186/s13678-016-0004-1](https://doi.org/10.1186/s13678-016-0004-1)
- [8] *Documentation | Android Developers* [online]. [vid. 2018-09-28]. Dostupné z: <https://developer.android.com/docs/>
- [9] IGUAL, Raul, Carlos MEDRANO a Inmaculada PLAZA. Challenges, issues and trends in fall detection systems. *BioMedical Engineering OnLine* [online]. 2013, **12**(1), 66 [vid. 2018-09-29]. ISSN 1475-925X. Dostupné z: [doi:10.1186/1475-925X-12-66](https://doi.org/10.1186/1475-925X-12-66)
- [10] ASHFAK HABIB, Mohammad, Mas S. MOHKOTAR, Shahrul BAHYAH KAMARUZZAMAN, Kheng SEANG LIM, Tan MAW PIN a Fatimah IBRAHIM. Smartphone-based solutions for fall detection and prevention: Challenges and open issues. *Sensors (Switzerland)* [online]. 2014, **14**(4), 7181–7208. ISSN 14248220. Dostupné z: [doi:10.3390/s140407181](https://doi.org/10.3390/s140407181)
- [11] KANSIZ, a. Oguz, M. Amac GUVENSAN a H. Irem TURKMEN. Selection of time-domain features for fall detection based on supervised learning. *2013 World Congress on Engineering and Computer Science* [online]. 2013, **2**, 796–801. ISSN 20780958. Dostupné z: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84903472852&partnerID=tZOtx3y1>

- [12] KERDEGARI, Hamideh, Khairulmizam SAMSUDIN, Abdul Rahman RAMLI a Saeid MOKARAM. Evaluation of fall detection classification approaches. *ICIAS 2012 - 2012 4th International Conference on Intelligent and Advanced Systems: A Conference of World Engineering, Science and Technology Congress (ESTCON) - Conference Proceedings* [online]. 2012, **1**, 131–136. Dostupné z: doi:10.1109/ICIAS.2012.6306174
- [13] TÖZEREN, Aydın. *Human Body Dynamics: Classical Mechanics and Human Movement* [online]. 2000. ISBN 0-387-98801-7. Dostupné z: doi:10.1007/b97432
- [14] NORDHOFF, L S. *Motor Vehicle Collision Injuries: Biomechanics, Diagnosis, and Management*. B.m.: Jones & Bartlett Learning, 2005. ISBN 9780763733353.
- [15] ALI, Haimd M a Zainab S ALWAN. Car accident detection and notification system using smartphone [online]. 2015, **4**(April), 620–635. Dostupné z: <https://www.ijcsmc.com/>
- [16] WHITE, Jules, Chris THOMPSON, Hamilton TURNER, Brian DOUGHERTY a Douglas C. SCHMIDT. WreckWatch: Automatic traffic accident detection and notification with smartphones. *Mobile Networks and Applications* [online]. 2011, **16**(3), 285–303. ISSN 1383469X. Dostupné z: doi:10.1007/s11036-011-0304-8
- [17] CHAPMAN, A E. *Biomechanical Analysis of Fundamental Human Movements-Google*. B.m.: Human Kinetics, 2008. ISBN 9781450408318.
- [18] *GPS.gov: GPS Accuracy* [online]. [vid. 2018-09-30]. Dostupné z: <https://www.gps.gov/systems/gps/performance/accuracy/>
- [19] ALDUNATE, Roberto, Oriel HERRERA a Juan Pablo CORDERO. Early Vehicle Accident Detection and Notification Based on Smartphone Technology. In: . B.m.: Springer International Publishing Switzerland, 2013, s. 358–365. ISBN 9783319031767.
- [20] KARDOUS, Chucrí A. a Peter B. SHAW. *Engineering Research Report Evaluation of Smartphone Sound Measurement* [online]. 2013 [vid. 2018-09-30]. Dostupné z: <https://www.cdc.gov/niosh/surveyreports/pdfs/349-12a.pdf>
- [21] ABBATE, Stefano, Marco AVVENUTI, Guglielmo COLA, Paolo CORSINI, Janet LIGHT a Alessio VECCHIO. Recognition of false alarms in fall detection systems. *2011 IEEE Consumer Communications and Networking Conference, CCNC'2011* [online]. 2011, (February), 23–28. ISSN 2331-9852. Dostupné z: doi:10.1109/CCNC.2011.5766464
- [22] JACOB, Jerene, Tam NGUYEN, Donald Y.C. LIE, Steven ZUPANCIC, J. BISHARA, Andrew DENTINO a Ron E. BANISTER. A fall detection study on the sensors placement location and a rule-based multi-thresholds algorithm using both accelerometer and gyroscopes. *IEEE International Conference on Fuzzy Systems* [online]. 2011, 666–671. Dostupné z: doi:10.1109/FUZZY.2011.6007744
- [23] HJORT, Bo. EEG Analysis Based On Time Domain Properties [online]. 1970,

306–310. ISSN 0013-4694. Dostupné z: doi:10.1016/0013-4694(70)90143-4

- [24] MARAGOS, P., J.F. KAISER a T.F. QUATIERI. On amplitude and frequency demodulation using energy operators. *IEEE Transactions on Signal Processing* [online]. 1993, **41**(4), 1532–1550. ISSN 1053587X. Dostupné z: doi:10.1109/78.212729
- [25] *TensorFlow Documentation* [online]. [vid. 2018-10-07]. Dostupné z: <https://www.tensorflow.org/>
- [26] TOM, Hope, Yehezkel S. RESHEFF a Itay LIEDER. *Learning TensorFlow: A Guide to Building Deep Learning Systems*. B.m.: O'Reilly Media, 2017. ISBN 978-1491978511.
- [27] *Keras Documentation* [online]. [vid. 2018-10-19]. Dostupné z: <https://keras.io/>
- [28] KESKAR, Nitish Shirish a Richard SOCHER. Improving Generalization Performance by Switching from Adam to SGD [online]. 2017, (1). Dostupné z: <http://arxiv.org/abs/1712.07628>
- [29] *NHTSA | National Highway Traffic Safety Administration* [online]. [vid. 2018-11-20]. Dostupné z: <https://www.nhtsa.gov/>

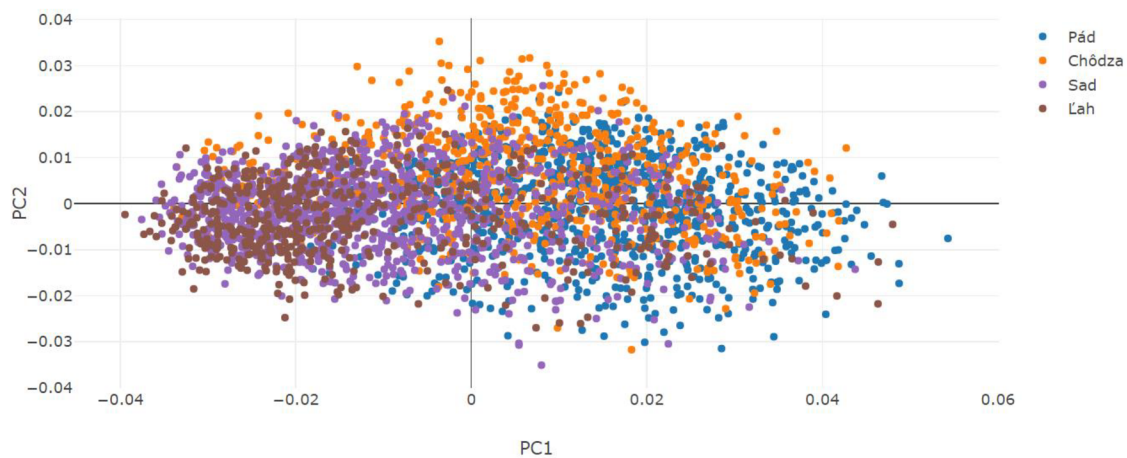
ZOZNAM SKRATIEK A ZNAČIEK

a_{x_i}	-	hodnota akcelerácie na danej osy
a	-	zrýchlenie
a_a	-	vektor akcelerácie na 3 osiach
AAMV	-	Average acceleration magnitude variation - priemernú odchýlku amplitúdy akcelerácie
acc_i	-	vzorka magnitúdy akcelerácie
AD	-	angle deviation – uhlová odchýlka
CA	-	change of angle – zmena uhla
CAA	-	celkovú aktivitu akcelerácie
FFAAM	-	Free fall average acceleration magnitude - priemerná hodnota amplitúdy akcelerácie voľného pádu
g	-	gravitačné zrýchlenie
GPS	-	globálny lokalizačný systém
ICA	-	nezávislá analýza komponent
NN	-	neurónová sieť
NMF	-	nie negatívna maticná faktorizácia
PCA	-	princiálna analýza komponent
SGD	-	stochastický gradientný spád
SVM	-	support vector machine – metóda podporných vektorov
t	-	čas
UNIX	-	názov rodiny operačných systémov
σ	-	smerodajná odchýlka

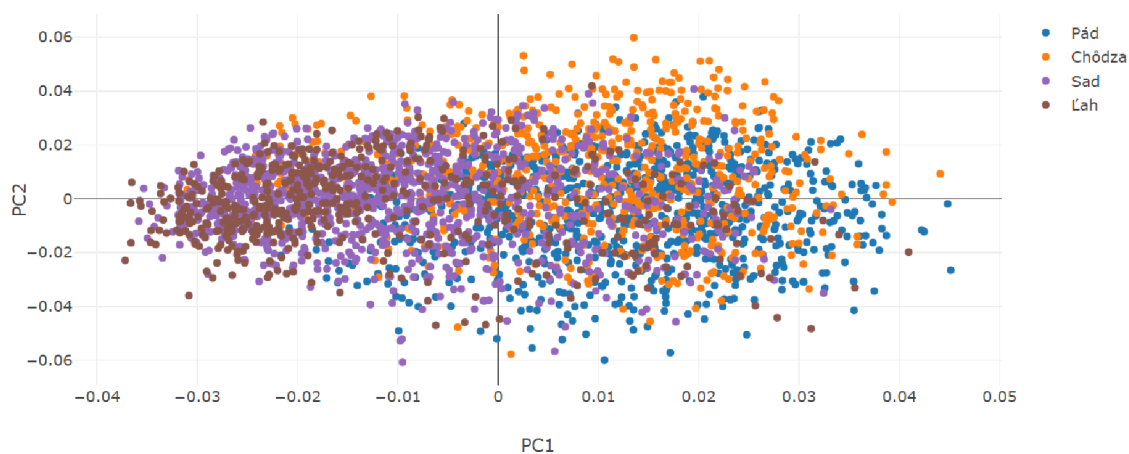
ZOZNAM PRÍLOH

- Príloha 1. Grafy transformácii parametrov pomocou PCA, ICA, NMF
- Príloha 2. Priebehy tréningových cyklov
- Príloha 3. Ukážky priebehu akcelerácie pri rôznych dopravných situáciach
- Príloha 4. CD s príslušnými kódmi

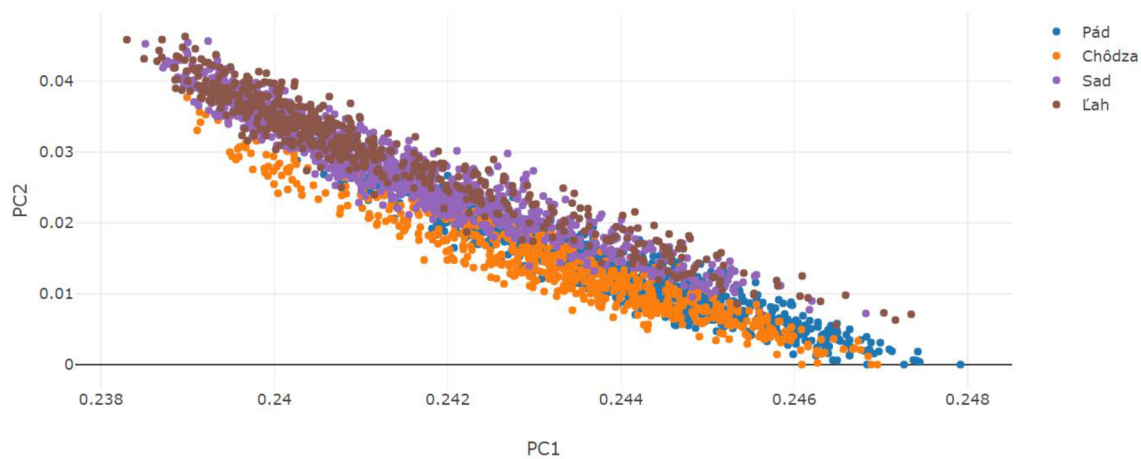
PRÍLOHA Č. 1



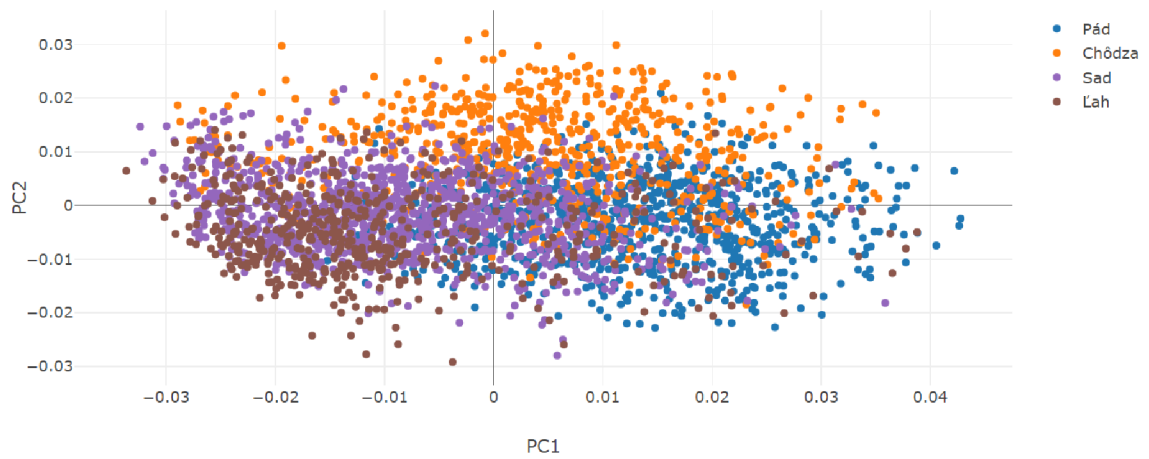
Obr. 1.1 PCA – Akcelerácia, uhlová rýchlosť, uhlové zrýchlenie



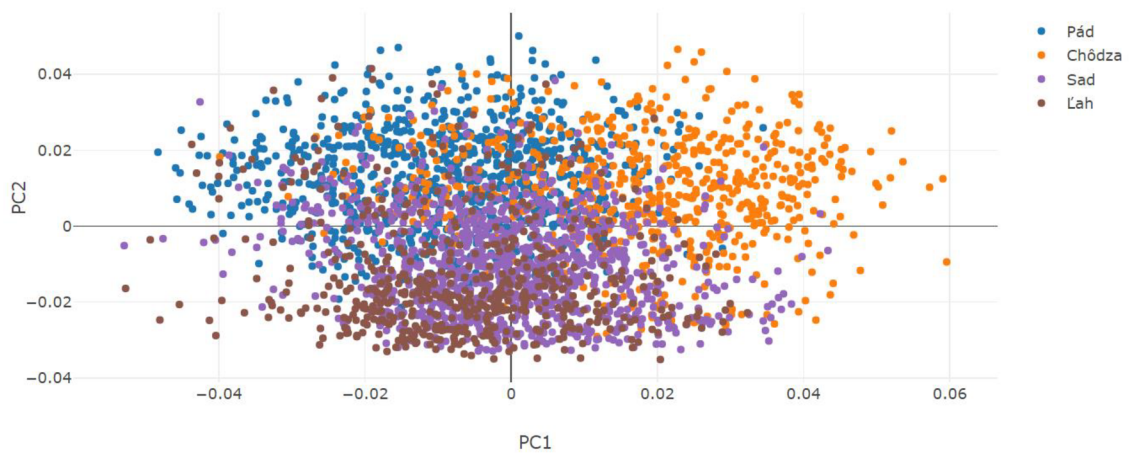
Obr. 1.2 ICA – Akcelerácia, uhlová rýchlosť, uhlové zrýchlenie



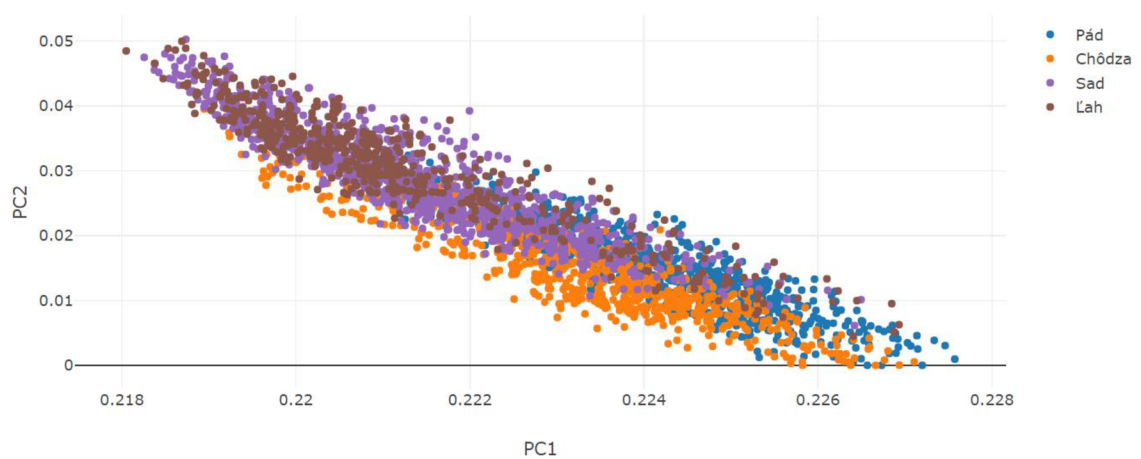
Obr. 1.3 NMF – Akcelerácia, uhlová rýchlosť, uhlové zrýchlenie



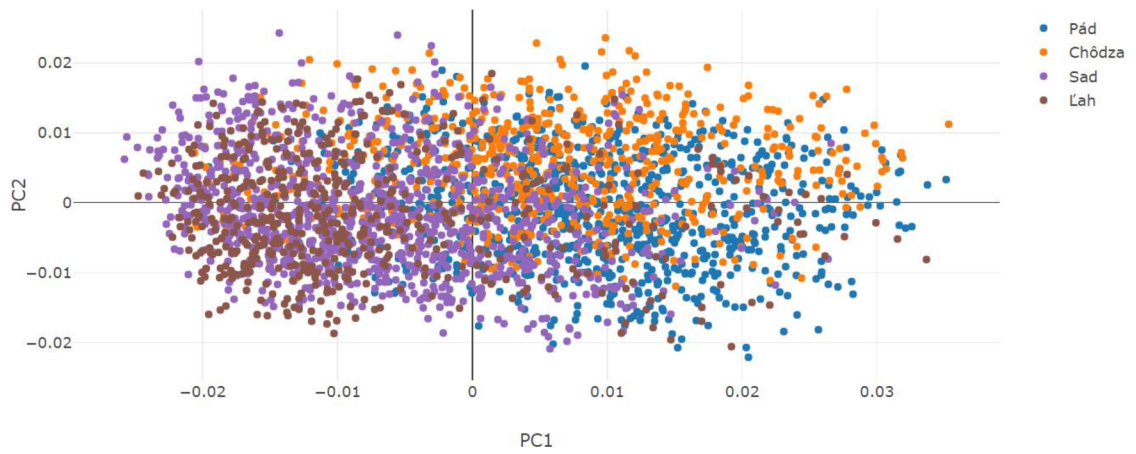
Obr. 1.4 PCA – Akcelerácia, Uhlová rýchlosť



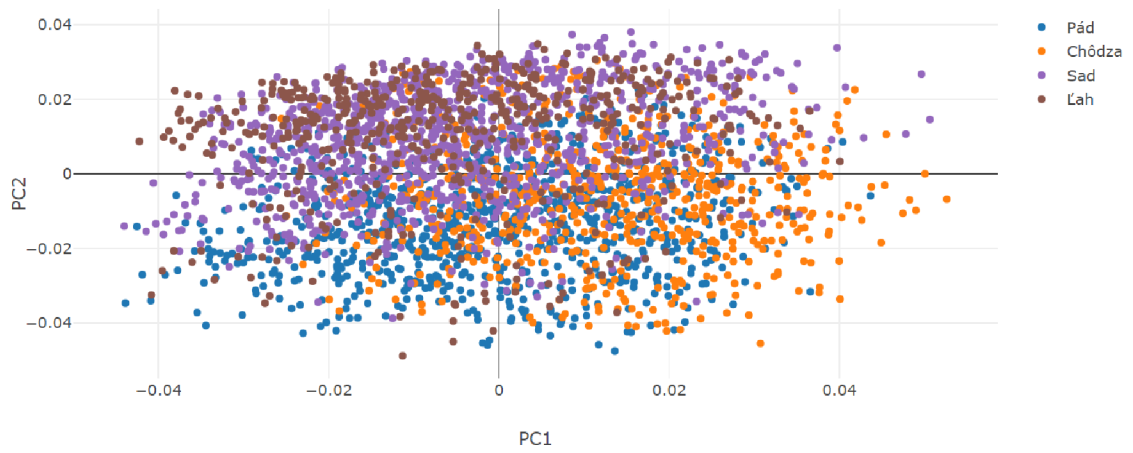
Obr. 1.5 ICA – Akcelerácia, uhlová rýchlosť



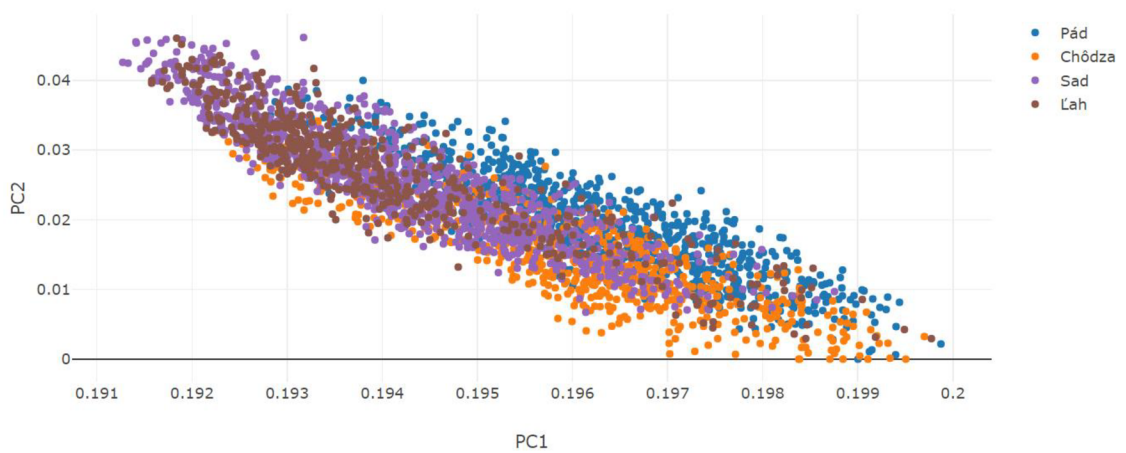
Obr. 1.6 NMF – Akcelerácia, uhlová rýchlosť



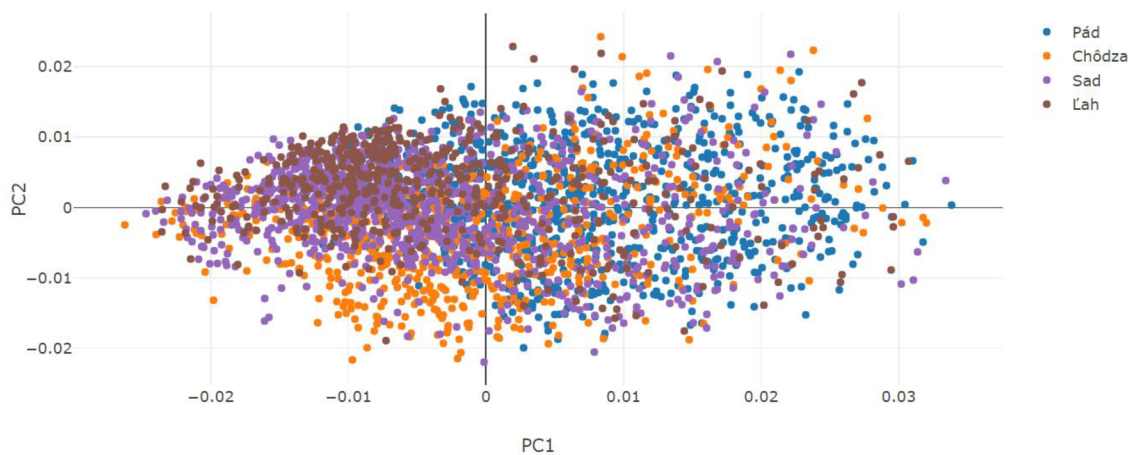
Obr. 1.7 PCA – Akcelerácia



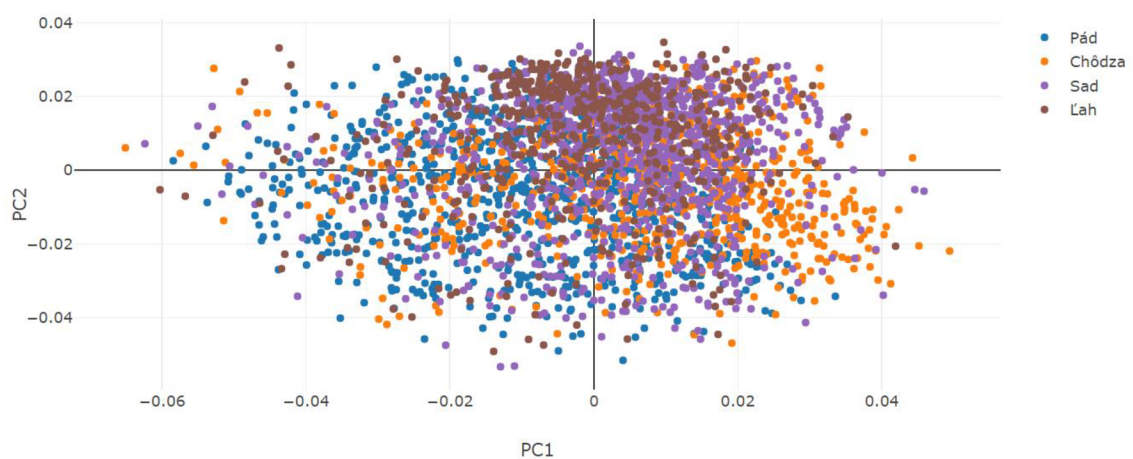
Obr. 1.8 ICA – Akcelerácia



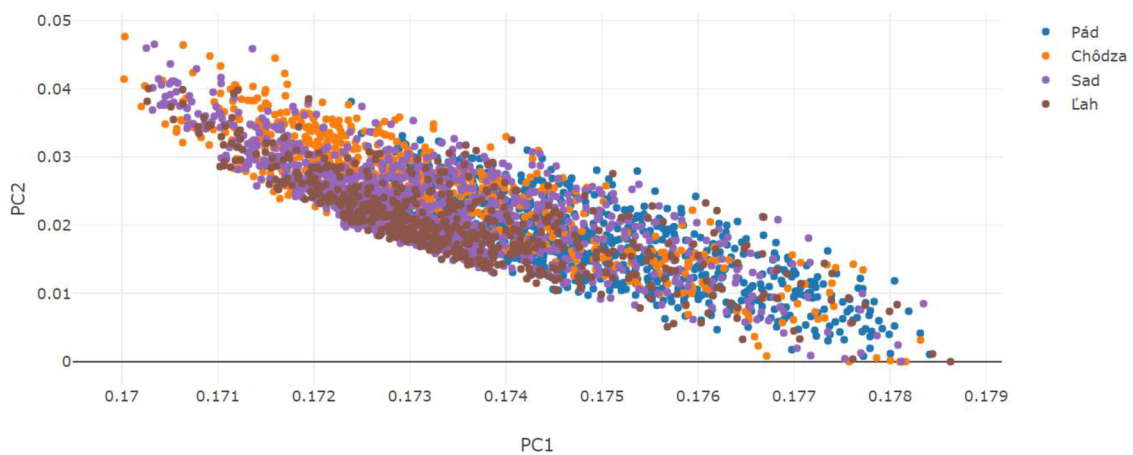
Obr. 1.9 NMF – Akcelerácia



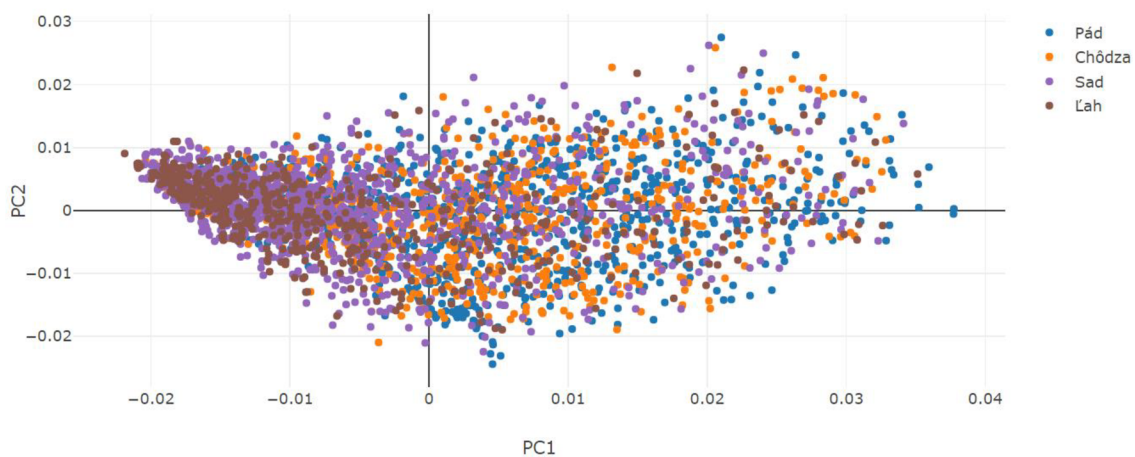
Obr. 1.10 PCA – Uhlová rýchlosť



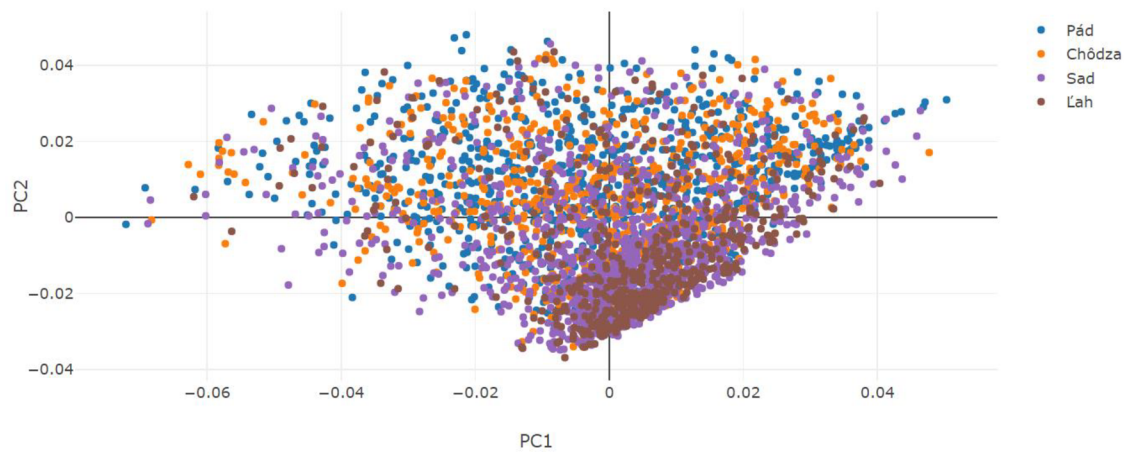
Obr. 1.11 ICA – Uhlová rýchlosť



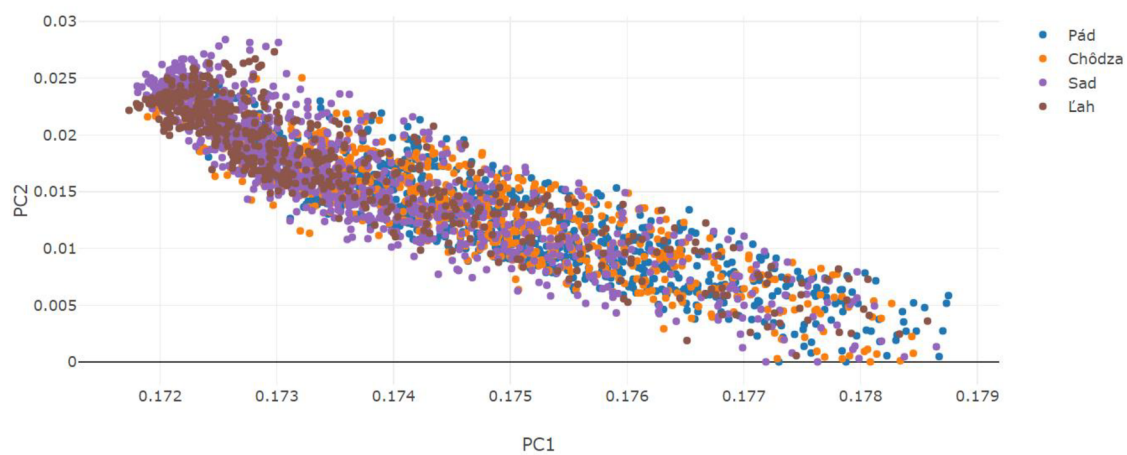
Obr. 1.12 NMF – Uhlová rýchlosť



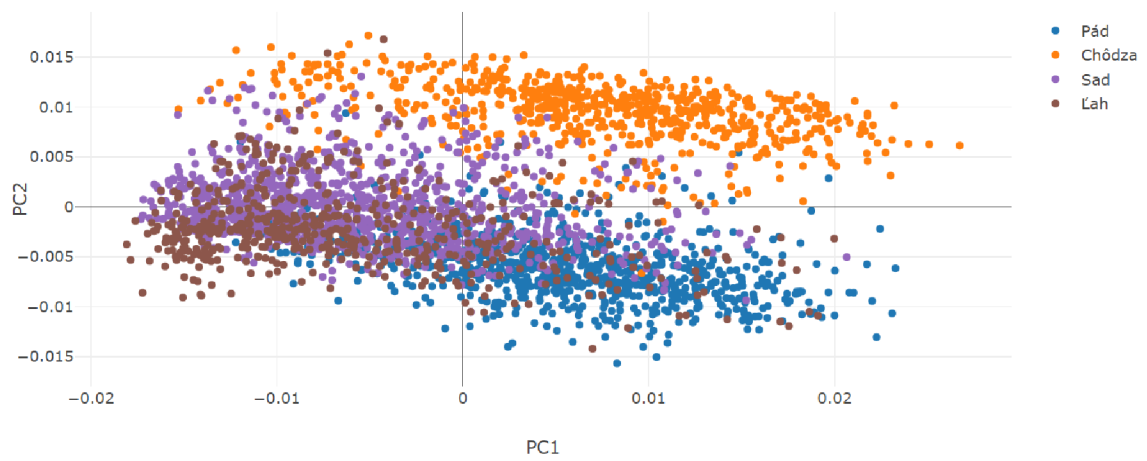
Obr. 1.13 PCA – Uhlové zrýchlenie



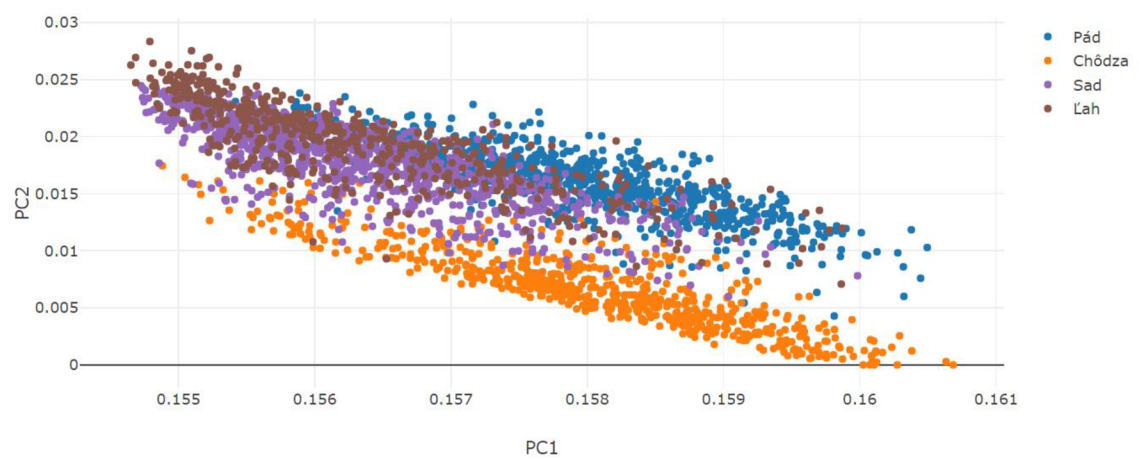
Obr. 1.14 ICA – Uhlové zrýchlenie



Obr. 1.15 NMF – Uhlové zrýchlenie

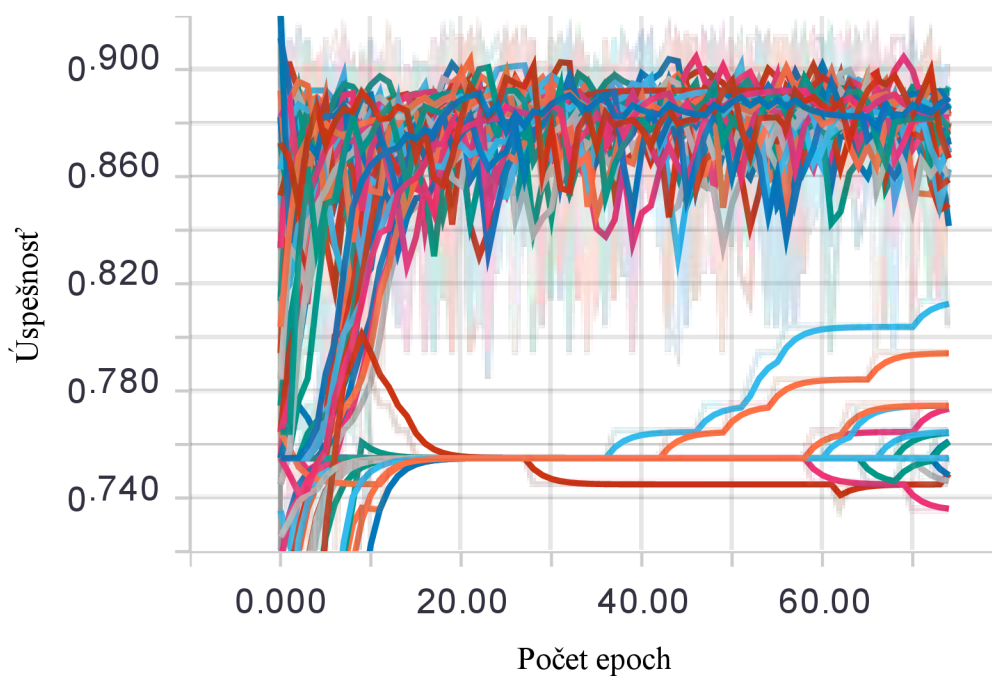


Obr. 1.16 PCA – Vybrané parametre

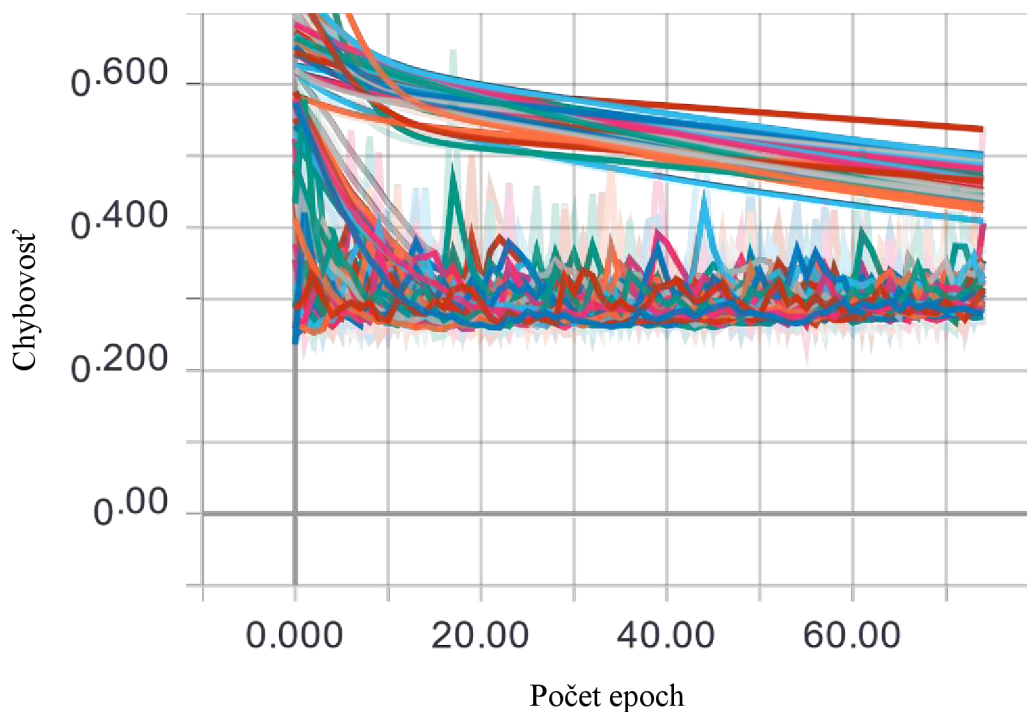


Obr. 1.17 NMF – Vybrané parametre

PRÍLOHA Č. 2

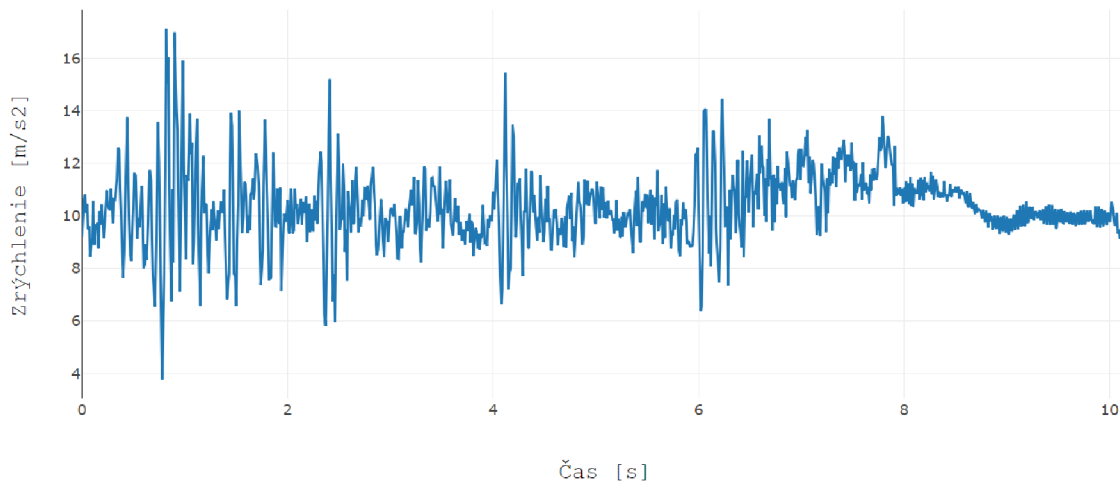


Obr. 2.1 Priebeh úspešnosti pri trénovaní neurónových sietí

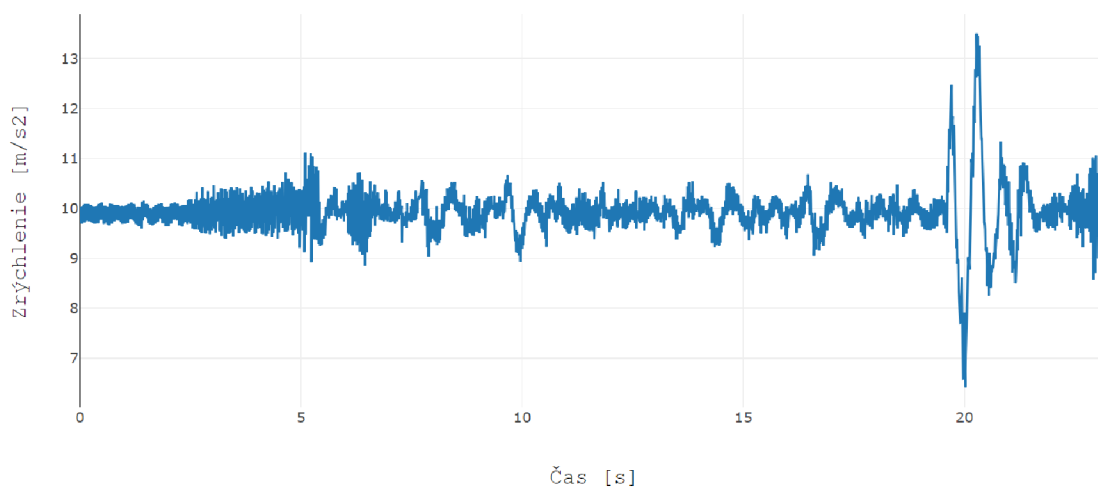


Obr. 2.2 Priebeh chybovosti neurónových sietí pri trénovaní

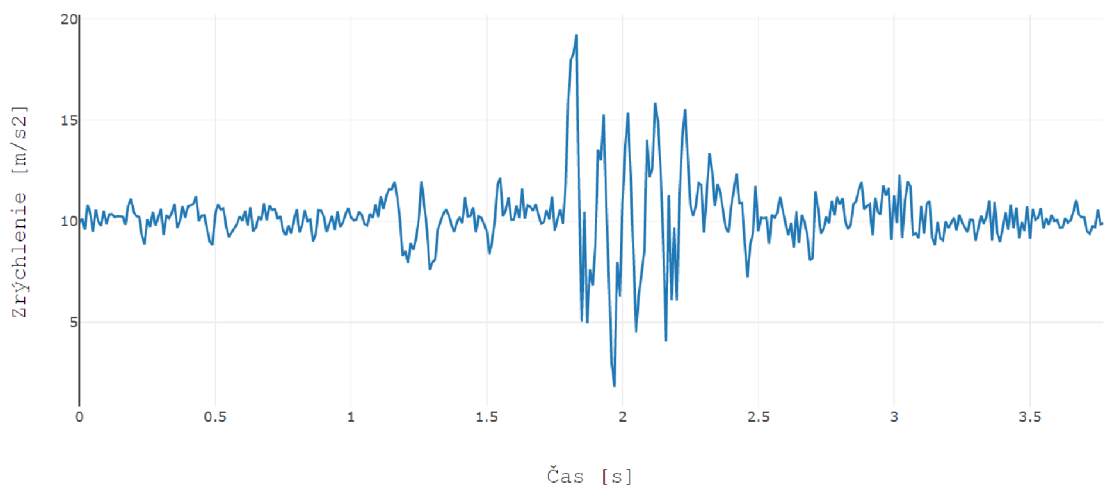
PRÍLOHA Č. 3



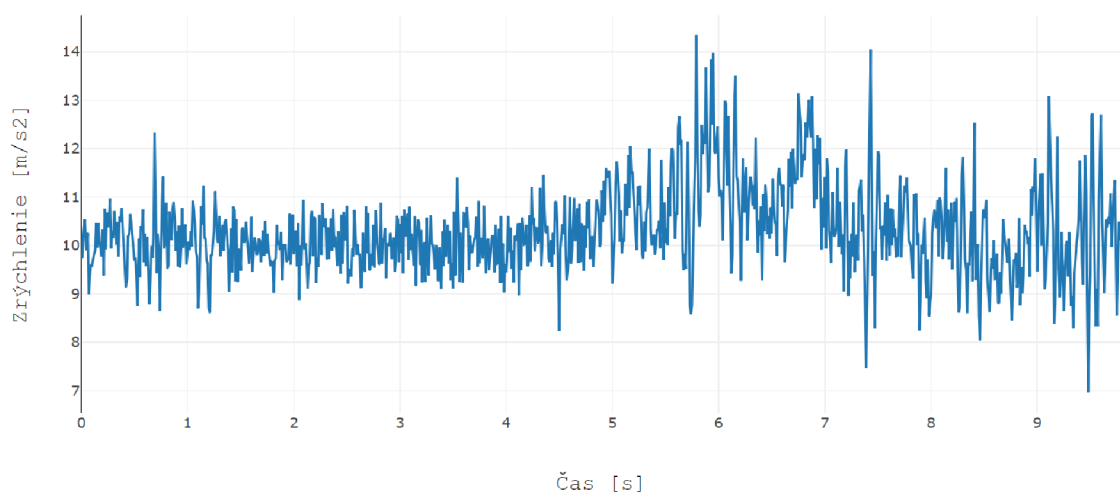
Obr. 3.1 Pribeh akcelerácie pri zastavení z 50 na 0 km/h



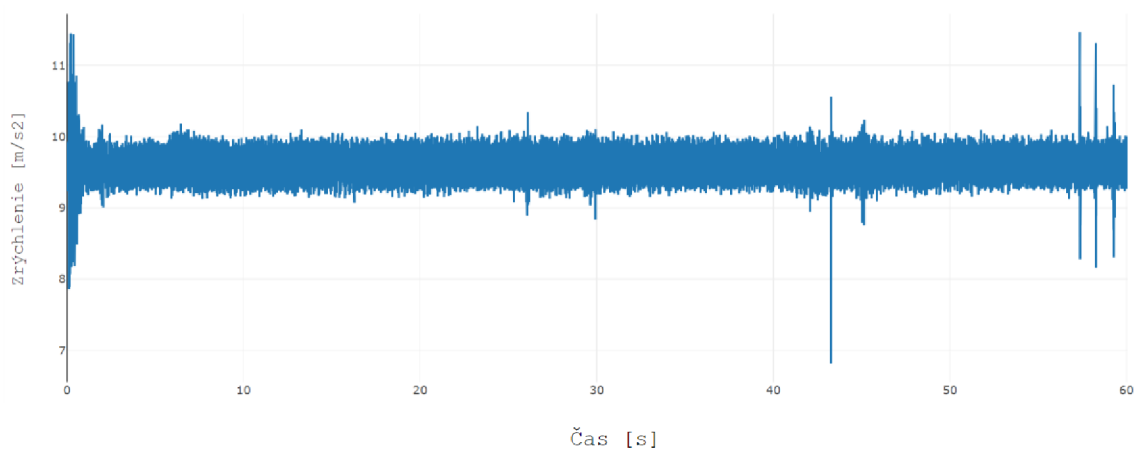
Obr. 3.2 Pribeh akcelerácie pri prechode cez podval s 10 km/h



Obr. 3.3 Pribeh akcelerácie pri prechode cez podval s 40 km/h



Obr. 3.4 Pribeh akcelerácie pri 90° zatáčke s 40 km/h



Obr. 3.5 Pribeh akcelerácie pri státi vozidla

PRÍLOHA Č. 4

Obsah CD:

- **data_processing/** – priečinok obsahuje všetky Python skripty na spracovanie a vizualizáciu dát pohybových udalostí a jazdy autom
- **mobile/** - obsahuje kód mobilnej aplikácie
 - **assets/** - úložisko pre zdrojové nekódové dáta
 - **java/** - hlavný zdrojový kód aplikácie
 - **src/** - vzhľad aplikácie napísaný pomocou XML a úložisko pre obrázky