

Prehlásenie autora

Čestne prehlasujem, že táto diplomová práca je mojim pôvodným autorským dielom, na ktorom som pracoval samostatne na základe vlastných teoretických a praktických poznatkov získaných počas štúdia a informácií z dostupnej literatúry. Všetky zdroje, literatúru a pramene, ktoré som pri vypracovaní použil, riadne citujem s uvedením plného odkazu na príslušný zdroj. Uvedenú prácu som vypracoval pod vedením Ing. Stanislava Marčeka.

Bratislava, dňa 20. 05. 2016

.....

Bc. Matúš Mačák

Pod'akovanie

Ďakujem rodine, priateľke Anne, vedúcemu Ing. Stanislavovi Marčekovi a môjmu trénerovi, Mgr. Patrikovi Kurilovi, ktorý ma o tréningu naučil všetko, čo som mohol využiť pri návrhu algoritmu.

Abstrakt

Univerzita: Slovenská technická univerzita v Bratislave

Fakulta: Fakulta elektrotechniky a informatiky

Študijný program: Aplikovaná informatika

Školiace pracovisko: Ústav informatiky a matematiky

Autor: Bc. Matúš Mačák

Názov práce: Aproximácia wattového výkonu športovca

Vedúci práce: Ing. Stanislav Marček

Bratislava 2016

Diplomová práca sa zaoberá hľadaním algoritmu pre odhadnutie wattového výkonu športovca, cyklistu. V úvode práce je vysvetlená dôležitosť sledovania výkonu ako hlavného ukazovateľa v tréningu. Popisuje zariadenia, aplikácie a štandardy, ktoré sú vyvinuté za účelom sledovania tréningových údajov. Jadro práce pojednáva o problematike odhadovania výkonu z rôznych údajov a prezentuje riešenie, ktoré vďaka strojovému učeniu umožňuje určiť približné hodnoty výkonu bez jeho priameho merania. Záver práce je venovaný rozboru a analýze presnosti údajov odhadnutých algoritmom.

Kľúčové slová: šport, cyklistika, výkon, ANT+, umelá neurónová sieť

Abstract

University: Slovak Technical University in Bratislava

Faculty: Faculty of Electrical Engineering and Information Technology

Study programme: Applied Informatics

Training workplace: Institute of Computer Science and Mathematics

Author: Matúš Mačák

Title: Approximation of athlete power output

Thesis supervisor: Ing. Stanislav Marček

Bratislava 2016

This master thesis describes the research of algorithm for estimating the power effort of an athlete, cyclist. First part explains the importance of monitoring the power as the main indicator in training. It describes devices, software and standards that are developed to track sport activity. The main part deals with common problems in power estimation from various data and presents the solution based on machine learning that allows approximating the value of athlete's power without a direct measurement. The conclusion is devoted to review and analyze the accuracy of the algorithm.

Keywords: sport, cycling, power, ANT+, artificial neural network

Obsah

0 Úvod	8
1 Analýza problému	9
1.1 Význam veličiny výkon v tréningu cyklistu	9
1.2 Štandardy pre prenos tréningových dát	11
1.2.1 GPX	11
1.2.2 TCX	11
1.2.3 ANT	12
1.2.4 FIT	12
1.3 Meranie výkonu	14
1.4 Systémy pre vyhodnotenie tréningu	15
1.4.1 Strava	15
1.4.2 Bike Calculator	18
2 Pôsobenie síl	20
2.1 Gravitačná sila	20
2.2 Aerodynamický odpor	21
2.3 Valivý odpor	24
2.4 Podiel jednotlivých zložiek na celkovom odpore	25
3 Opis riešenia	27
3.1 Strojové učenie	28
3.1.1 Rozhodovacie regresné stromy	29
3.1.2 Umelé neurónové siete	30
3.2 Použité technológie a riešenia	32
3.3 Prípady použitia	33
3.4 Diagram aktivít	35
3.5 Implementácia	36
3.5.1 Nahranie a konverzia súboru	37
3.5.2 Detekcia stúpaní	38
3.5.3 Zobrazenie tréningu	41
3.5.4 Implementácia neurónovej siete	45
3.5.5 Trénovanie neurónovej siete	47
3.6 Porovnanie výsledkov	49

4 Záver**55**

Zoznam obrázkov

1	Porovnanie výkonu a tepovej frekvencie pri záťaži	10
2	Zjednodušená štruktúra FIT súboru [18]	13
3	CSV súbor po konverzii z FIT [18]	13
4	Strava - Analýza tréningu	16
5	Strava - Graf výkonu	17
6	Strava - Graf odhadnutého výkonu podľa GPS	17
7	Strava - Graf odhadnutého výkonu podľa Assisted GPS	17
8	Bike Calculator - Kalkulačka výkonu cyklistu	18
9	Teleso na naklonenej rovine	20
10	Závislosť potrebného výkonu od nadmorskej výšky	23
11	Závislosť potrebného výkonu od teploty vzduchu	23
12	Valivý odpor [7]	24
13	Závislosť potrebného výkonu od koeficientu odporu pneumatiky	25
14	Podiel silových zložiek v závislosti od stúpajúceho sklonu vozovky	26
15	Rozhodovací strom	29
16	Rozdelenie množiny hodnôt	29
17	Neurónová sieť	30
18	Neurón	31
19	Architektúra webovej aplikácie	32
20	Diagram prípadov použitia	35
21	Diagram aktivít	36
22	Detekcia segmentu	40
23	Zobrazenie mapy v detailnom náhľade tréningu	42
24	Zvýraznenie výberu pred priblížením konkrétnej časti tréningu	44
25	Porovnanie aproximácie výkonu - úsek č.1 - meranie č.1	50
26	Porovnanie aproximácie výkonu - úsek č.1 - meranie č.2	51
27	Porovnanie aproximácie výkonu - úsek č.2 - meranie č.3	52
28	Porovnanie aproximácie výkonu - úsek č.3 - meranie č.4	53
29	Porovnanie aproximácie výkonu - úsek č.3 - meranie č.5	54

0 Úvod

Technologický pokrok je bezpochyby dôvodom, prečo sa rôzne inteligentné zariadenia stávajú naším najdôležitejším nástrojom pri každodennej rutine, ale aj v krízových situáciách. Inak tomu nie je ani v športe, najmä ak sa jedná o vytrvalostné disciplíny. Na pulte takmer každej športovej predajne sú k dispozícii merače srdcového tepu, športové hodinky, cyklopočítače s GPS, a podobne. Nielen tréning podľa vlastného pocitu, ale podriadenie tréningovej záťaže údajom na meracom zariadení sa stáva kľúčom k postupnému zvyšovaniu výkonnosti. Vysoká dostupnosť technológií a *know-how* vyrovnáva schopnosti konkurencie. Víťazia športové družstvá, ktoré okrem mimoriadneho talentu dokážu tréning najlepšie zmerať, vyhodnotiť a opätovne naplánovať.

V práci sa zameriavame na analýzu vytrvalostného športu - cyklistiky, ktorú môžeme v globále uskutočniť z údajov, ako je dĺžka tréningu v hodinách, tzv. tréningový objem, vzdialenosť, rýchlosť, tepová frekvencia. Tréning podľa tepovej frekvencie môže byť pre športovca obrovským prínosom za predpokladu, že má správne stanovené tréningové zóny, ktoré by mali odzrkadľovať zaťaženie organizmu [5]. Tepovú frekvenciu už môžeme v dnešnej dobe považovať za pomerne nenákladný spôsob usmerenia tréningu, no má však jednu nevýhodu. Tep srdca reaguje na stúpajúce zaťaženie s miernym oneskorením a môže byť skreslený pri výkyve vnútornej rovnováhy organizmu. V cyklistike, kde na reálny výkon vyjadriteľný fyzikálnou veličinou vplýva veľké množstvo faktorov, je aktuálna rýchlosť síce okamžitým, ale veľmi nepresným ukazovateľom. Preto je merač rýchlosti aj tepovej frekvencie postupne nahrádzaný meračom výkonu.

Meranie výkonu v cyklistike nám poskytuje okamžitú odpoveď na otázky: ako veľmi sa športovec namáha, akou silou pôsobí na pedále. Subjektívny pocit námahy nahrádza konkrétny číselný údaj, štandardne vyjadrený v jednotkách *Watt [W]*. Problémom je však cenová dostupnosť takéhoto merania. U profesionálnych športovcov možno cenová náročnosť nehrá rolu, avšak sú aj športovní nadšenci, amatérski pretekári, ktorí si chcú zmerať svoj progres, prípadne sa porovnať s najlepšími. Výkon je miera toho ako rýchlo vykonáva určitá sila prácu [8]. Je to fyzikálna veličina, a teda by malo byť možné ho vypočítať na základe údajov, ako je stúpanie, rýchlosť, vzdialenosť. Práve týmto výpočtom sa snažíme odhadnúť výkon športovca.

1 Analýza problému

1.1 Význam veličiny výkon v tréningu cyklistu

Ako sme spomenuli v úvode, obrovská dostupnosť informácií a prostriedkov umožňuje vykonávať kontrolovaný tréning takmer každému záujemcovi. V tejto podkapitole sa opierame o poznatky zo štúdií literatúry [5]. Pokiaľ máme záujem o perspektívne zlepšovanie, prvým krokom by malo byť absolvovanie výkonnostného testu u športového lekára, tzv. funkčného testu. Lekár by na základe anamnézy, aktuálnej výkonnosti a predchádzajúcich skúseností so športom mal odporučiť primerané tréningové zaťaženie. Čo je však najdôležitejšie, mal by stanoviť aj tréningové zóny, v ktorých by sa športovec pri tréningu mal pohybovať a v akom trvaní, teda odporučiť tréningový plán. Tréning cyklistu nemôže byť postavený iba na odjazdení určitej vzdialenosti v čo najkratšom čase, a ani tréning vytrvalosti tým, že sa trénuje čo najdlhšie. Tréning cyklistu je zložený z rôznych intenzít, úsekov, opakovaní a cvičení, kedy sa trénujúci športovec vedome pohybuje v určite tréningovej zóne aby docielil ideálny tréningový impulz.

Zámerne používame slovné spojenie tréningová zóna a nielen tepová zóna, hoci sa tieto slovné spojenia často zamieňajú. Tréningová zóna je však úroveň fyzického zaťaženia, ktoré prebieha v ľudskom organizme. Najjednoduchšie odporúčané delenie je:

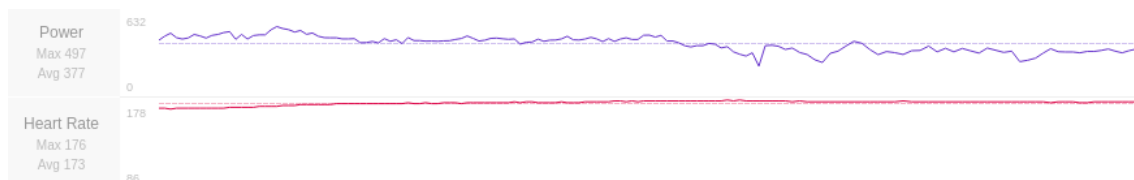
- Aeróbne pásmo
- Zmiešané pásmo
- Anaeróbny prah
- Anaeróbne pásmo

Ako sú tréningové zóny kontrolované počas tréningu je len na samotných možnostiach športovca. Najideálnejším ukazovateľom by bola hladina laktátu v krvi. Túto metódu však do každodenného tréningu aplikovať nemôžeme, nakoľko odber krvi z prsta alebo ucha a následné vyhodnotenie vzorky je zdĺhavé a nepraktické. Najdostupnejšia metóda je meranie tepovej frekvencie, ktoré však nie je vždy spoľahlivé.

Každý ľudský organizmus je unikátny komplexný biologický systém. Tlak krvi, hormonálna rovnováha, krvný obraz a iné faktory neustále ovplyvňujú tep srdca.

Mierny rozdiel v tepovej frekvencii pri rovnakej záťaži môžeme pozorovať už pri porovnaní tréningu ráno a večer. Pred prichádzajúcim ochorením môžeme zaznamenať zvýšenie tepovej frekvencie. Nižší tep športovci vykazujú po niekoľkých dňoch náročného tréningu. Taktiež sme sa stretli s prípadmi, keď športovec pociťoval únavu a oproti normálu mal mierne zníženú tepovú frekvenciu, ale jeho wattový výkon v náročnom tréningu bol porovnateľný, niekedy aj lepší ako v iné dni. Vyšší tep sa dá jasne pozorovať už pri veľmi miernej dehydratácii, napr. na konci dlhého tréningu v teplom počasí. Sledovanie faktorov, ktoré ovplyvňujú tepovú frekvenciu nie je predmetom tejto práce. Je však dôležité poznamenať, že z pohľadu fináčnej náročnosti a výpovednej hodnoty je tepová frekvencia stále veľmi dobrým kontrolným mechanizmom pre riadenie tréningu.

Okamžité hodnoty s dobrou presnosťou, ktoré reprezentujú zaťaženie organizmu, získame meraním výkonu. Intenzitu môžeme určiť pri krátkych zrýchlených úsekoch, kedy tepová frekvencia ešte nestihne vystúpiť na hodnotu zodpovedajúcu fyzickému zaťaženiu. Môžeme presnejšie analyzovať podaný výkon pri dlhších úsekoch v maximálnom úsilí, kedy zrýchlenie/spomalenie môže odzrkadľovať zmena tepová frekvencia o 1 tep, ale rozdiel vo výkone môže byť aj o 10%. Obr. 1 zobrazuje, ako sa môže meniť výkon športovca v porovnaní s tepovou frekvenciou.



Obr. 1: Porovnanie výkonu a tepovej frekvencie pri záťaži

Z priebehu kriviek môžeme pozorovať, ako odpočínutý športovec absolvuje úsek v tréningu. Počas desiatich minút mal udržiavať výkon vyšší ako 340 W. Vidíme, že na stúpajúce zaťaženie telo reaguje zvýšením tepovej frekvencie, zároveň sa v zaťažovaných svaloch tvorí kyselina mliečna (laktát). Prvých päť minút športovec absolvoval s priemerným výkonom 410W. Po zahľtení svalov laktátom a ustálení tepovej frekvencie už daný výkon udržiavať nedokázal a postupne záťaž zvoľňuje na priemerných 320W. Kým tepová frekvencia kolíše v rozsahu ± 1 tep, priemerný výkon v druhej polovici je o 20W nižší než požadované zaťaženie. Zmena však stále nie je dostatočná na to, aby mohlo dôjsť k čiastočnému zotaveniu svalov, hladina laktátu v krvi je stále vysoká, na čo telo reaguje vysokou tepovou frekvenciou. Hoci po skončení tréningu má športovec pocit vyčerpanosti, neabsolvoval tréning podľa plánovaného konceptu. Zdokonalenie

podobných chýb môže viesť k lepšiemu rozloženiu síl a lepšiemu vystupňovaniu výkonu, napr. v závere pretekov. Pri analýze výkonu sú tieto chyby jasne viditeľné.

1.2 Štandardy pre prenos tréningových dát

V predchádzajúcich častiach sme niekoľkokrát spomenuli športové hodinky alebo cyklopočítač. Ich spoločnou funkciou je prijímanie dát z rôznych snímačov, zobrazenie dát, a tiež aj uloženie pre neskoršie vyhodnotenie. Na trhu je veľké množstvo výrobcov, ktorý uprednostňujú vlastné formy prenosu a formu ukladania dát. Existujú však aj *open-source* protokoly, ktoré umožňujú interoperabilitu medzi zariadeniami rôznych značiek, prípadne aj inteligentnými telefónmi. Vo výskume sme sa na rozbor problematiky a následné vyhodnotenie presnosti rozhodli používať zariadenia na meranie výkonu PowerTap G3 (rok výroby 2013), Stages Rival (rok výroby 2014) a cyklopočítač Garmin Edge 500 (rok výroby 2014). Preto chceme v nasledujúcej časti popísať štandardy, ktoré tieto zariadenia využívajú, a teda sa bezprostredne dotýkajú našej práce.

1.2.1 GPX

GPX je rozšírenie formátu XML vytvorené za účelom štandardizácie prenosu GPS súradníc a trás pre navigačné systémy. Prvá verzia bola vydaná v roku 2002 a neskôr aktualizovaná v roku 2004. Neskôršie rozšírenie od firmy Garmin - schéma *GpxExtensions* umožňuje medzi GPS súradnice pridávať body záujmu¹, meniť zobrazenie trasy, a podobne. Neskôr Garmin pridal aj schému *TrackPointExtension*, ktorú už bolo možné využiť aj v zariadeniach, ktoré súvisia so športom. Okrem informácii o trase je možné do súboru uložiť aj údaje o tepovej frekvencii, teplote, rýchlosti a ďalšie [18].

1.2.2 TCX

TCX alebo plným názvom Training Center XML je dátový formát predstavený v roku 2007, opäť firmou Garmin ako súčasť ich Training Center produktov. Formát XML je veľmi podobný štandardu GPX, avšak postupnosť súradníc reprezentuje skôr ako fyzickú aktivitu než len trasu. Predstavuje štandard prenosu údajov zo športových meračov ako je tepová frekvencia, kalórie, frekvencia šliapania (bicykel) alebo kroku (beh) a mnohé ďalšie. Tiež môže obsahovať sumarizované údaje pre celú aktivitu aj jednotlivé okruhy [18].

¹z angl. Points of interest

1.2.3 ANT

Podľa [18] je ANT nízko-energetický² bezdrôtový protokol prenosu dát medzi zariadeniami. Robustný ale veľmi flexibilný protokol nachádza využitie v PAN³ alebo LAN⁴ sieťach. Je ideálny pre športové aktivity, fitness a wellness.

ANT prenos dát prebieha na frekvencii medzi 2400 MHz až 2524 MHz s výnimkou frekvencie 2457 MHz, ktorá je rezervovaná pre ANT+ zariadenia.

ANT+ je množina definícií, ktoré upresňujú, čo reprezentujú informácie posielané v sieti ANT. Definícia sa nazýva *device profile* a obvykle je vytvorená na základe určitého prípadu použitia. Napríklad senzor rýchlosti na bicykli bude odosielať informácie o rýchlosti, preto implementuje profil *Speed monitor*.

Každé ANT+ zariadenie musí pristupovať do ANT+ siete so sieťovým kľúčom a implementovať minimálne jeden *device profile*. Zariadenie s daným profilom je možné spojiť s ďalšími zariadeniami, ktoré implementujú rovnaký profil. Jedno zariadenie môže samozrejme implementovať aj viac profilov. Napr. športové hodinky môžu prijímať údaje o rýchlosti, tepovej frekvencii, výkone a pre každý údaj implementujú samostatný *device profile*.

V ANT+ sieti obvykle rozlišujeme dva typy zariadení. Meracie zariadenie, ktoré je podľa špecifikácie nazývané *sensor* a zobrazovacie zariadenie, *display*. ANT+ umožňuje pripojenie jedného, alebo aj viacerých zariadení typu *sensor* na jedno zariadenie typu *display*. Na jedných športových hodinách teda môžeme zobraziť viacero meraných veličín, alebo pripojiť jedno zariadenie typu *sensor* na viac zariadení typu *display*.

1.2.4 FIT

Na základe informácií z [18], The Flexible and Interoperable Data Transfer (FIT) je protokol, navrhnutý špeciálne pre ukladanie a prenos dát pochádzajúcich zo športových aktivít. Definuje niekoľko šablón pre ukladanie dát, tzv. *Fit message*. Môžu v nich byť uložené informácie o používateľskom profile, dáta zo športovej aktivity a podobne. Bol navrhnutý tak aby bol kompaktný, interoperabilný a rozšíriteľný.

FIT protokol pozostáva z:

- Definície štruktúry .fit súboru
- Definície dátových polí a dátových typov pre jednotlivé *Fit messages*

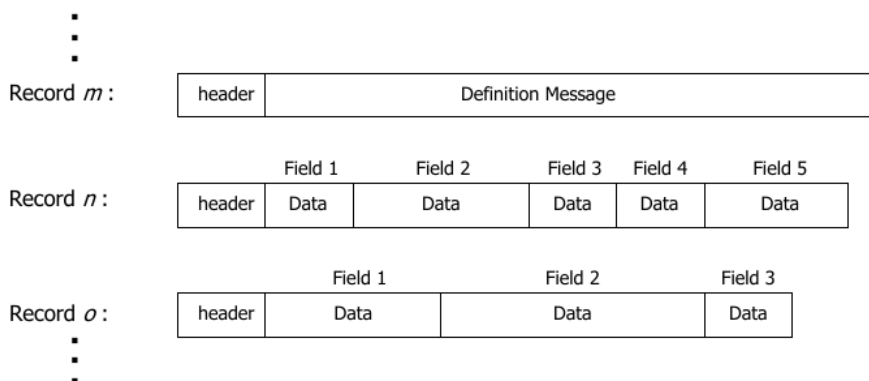
²z angl. Ultra-low power

³PAN - personal area network

⁴LAN - local area network

- Vývojového nástroja, tzv. SDK⁵, ktorý umožňuje vygenerovať potrebný zdrojový kód pre nakonfigurovanie produktov využívajúcich FIT protokol. Obsahuje aj nástroj pre konverziu binárneho .fit súboru do .csv dátového súboru a opačne.

FIT súbor obsahuje dátové reťazce, pričom každý reťazec je zostavený z hlavičky a obsahu. Obsah dátového reťazca môže byť buď špecifikácia dát v nasledujúcom reťazci alebo správa, ktorá obsahuje niekoľko polí s údajmi, viď obr. 2.



Obr. 2: Zjednodušená štruktúra FIT súboru [18]

Na konverziu bitového FIT súboru využijeme konverzný nástroj v jazyku JAVA. Výstupom konverzie sú dva CSV súbory. Prvý súbor je textovou reprezentáciou dát z FIT súboru, čo znamená, že niektoré riadky sú definíciou nasledujúcich údajov a niektoré riadky nesú údaje. Zároveň je potrebné si všimnúť, že údaje nie sú rozdelené fixne podľa stĺpcov. Stĺpec *Field N* oznamuje, čo reprezentuje hodnota v stĺpci *Value N*, stĺpec *Units N* definuje jednotky pre túto hodnotu, pričom $N > 0$. Jeden stĺpec *Field N* nemusí niesť vždy rovnaký údaj. Príklad uvádzame na obr. 3.

Type	Local Number	Message	Field 1	Value 1	Units 1	Field 2	Value 2	Units 2	...	Field N
Definition	0	file_id	serial_number	1		time_created	1			
Data	0	file_id	serial_number	3882026635		time_created	793193395			
Definition	1	file_creator	software_version	1		hardware_version	1			
Data	1	file_creator	software_version	330						
Definition	2	event	timestamp	1		data	1			
Data	2	event	timestamp	793193385	s	timer_trigger	0			

Obr. 3: CSV súbor po konverzii z FIT [18]

Druhý súbor, ktorý je výstupom z konverzie má príponou *_data.csv* a obsahuje všetky dáta usporiadané vo forme stĺpcov. Jeden stĺpec reprezentuje vždy konkrétny

⁵SDK - Software development kit

údaj podľa hlavičky, napr. *event.timestamp[s]*, *record.altitude[m]*, *record.speed[m/s]*... Jednotky sú uvádzané v hranatej zátvorke v záhlaví stĺpca. Oba súbory teda obsahujú všetky dáta, rozdielna je iba ich postupnosť. Rozhodli sme sa používať prvú spomenutú štruktúru. Najdôležitejšie sú riadky, kde stĺpec *Type* nadobúda hodnotu *Data*, v stĺpci *Message* sú hodnoty *record* pre jednotlivé body v tréningu, *session* pre sumarizáciu alebo *lap* pre informáciu o stlačení medzičasu.

1.3 Meranie výkonu

Pre správne pochopenie problematiky sa musíme oboznámiť so zaužívaným spôsobom merania výkonu cyklistu. Výkon je meraný špeciálnym snímačom, ktorý je vo väčšine prípadoch zabudovaný priamo v komponente bicykla. Jedná sa o komponent, ktorý sa priamo stará o prenos sily pôsobiacej na pedál do pohonu zadného kolesa. Môže to byť náboj zadného kolesa, jedna alebo obe kluky stredového zloženia, prípadne pedále. To, kde je snímač umiestnený, závisí od konkrétneho výrobcu a jeho techniky merania. Snímač nazývame merač výkonu⁶.

Najdôležitejším prvkom merača je tenzometer. Ide o súčiastku s uzavretým elektrickým okruhom a konštatným elektrickým odporom. Ak je tenzometer vystavený pôsobeniu mechanickej sily, dochádza k jeho deformácii, zmene odporu a vďaka piezoelektrickému javu aj k zmene napätia. Správnou konfiguráciou je teda možné zistiť presnú silu pôsobiacu na túto súčiastku.

Ako sme uviedli v predchádzajúcej časti, výkon je mierou toho ako rýchlo vykonáva určitá silu prácu. Vezmime do úvahy jednoduchý vzťah [7]:

$$P = F \frac{\partial d}{\partial t} \quad (1)$$

Výkon P je rovný sile, ktorú je potrebné vyvinúť, aby sme teleso s určitou hmotnosťou premiestnili po dráhe d za čas t . Dáta obsiahnuté v súboroch, s ktorými pracujeme (GPX, TCX, FIT), sú diskkrétne hodnoty uložené počas tréningu v niekoľko sekundových intervaloch. Vzťah 1 teda môžeme upraviť, keďže dráha prekonaná za čas predstavuje rýchlosť.

$$P = F v \quad (2)$$

To je podstata fungovania merača výkonu vo veľmi zjednodušenej forme. Sila zmeraná tenzometrom je odoslaná do cyklopočítača/športových hodín, tam je násobená

⁶z angl.: Power Meter

aktuálnou rýchlosťou a zobrazí sa výkon v jednotkách *Watt*. Podrobnejšie skúmanie toho, ako konkrétne typy meračov merajú výkon a jeho ďalšie fyzikálne vyjadrenie nie je predmetom tejto práce, keďže sa budeme snažiť odhadovať výkon na základe pohybu vo vonkajšom prostredí. Sily, ktoré pôsobia proti pohybu športovca pri jazde na bicykli bližšie rozoberáme v kapitole 2.

Prvý merač výkonu bol dostupný verejnosti v roku 1989. Jednalo sa o značku SRM (Schoberer Rad Messtechnik), ktorá svoj produkt testovala s profesionálnymi športovcami od začiatku osemdesiatych rokov [10].

Väčšina výrobcov v súčasnosti uvádza pre svoje merače odchýlku do 2%. Možné odchýlky môžu byť spôsobené stratou pri prenose sily z pedálu na pohon zadného kola, t.j. napnutím reťaze, pružnosťou rámu, pružnosťou kola, kalibráciou merača atď. Je dôležité si uvedomiť, že žiaden merač, ktorý je súčasťou bicykla, z týchto dôvodov nebude ukazovať presný výkon podávaný športovcom. Ak však športovec dlhodobo používa rovnakú metódu merania, porovnávanie jeho výkonov má veľký význam.

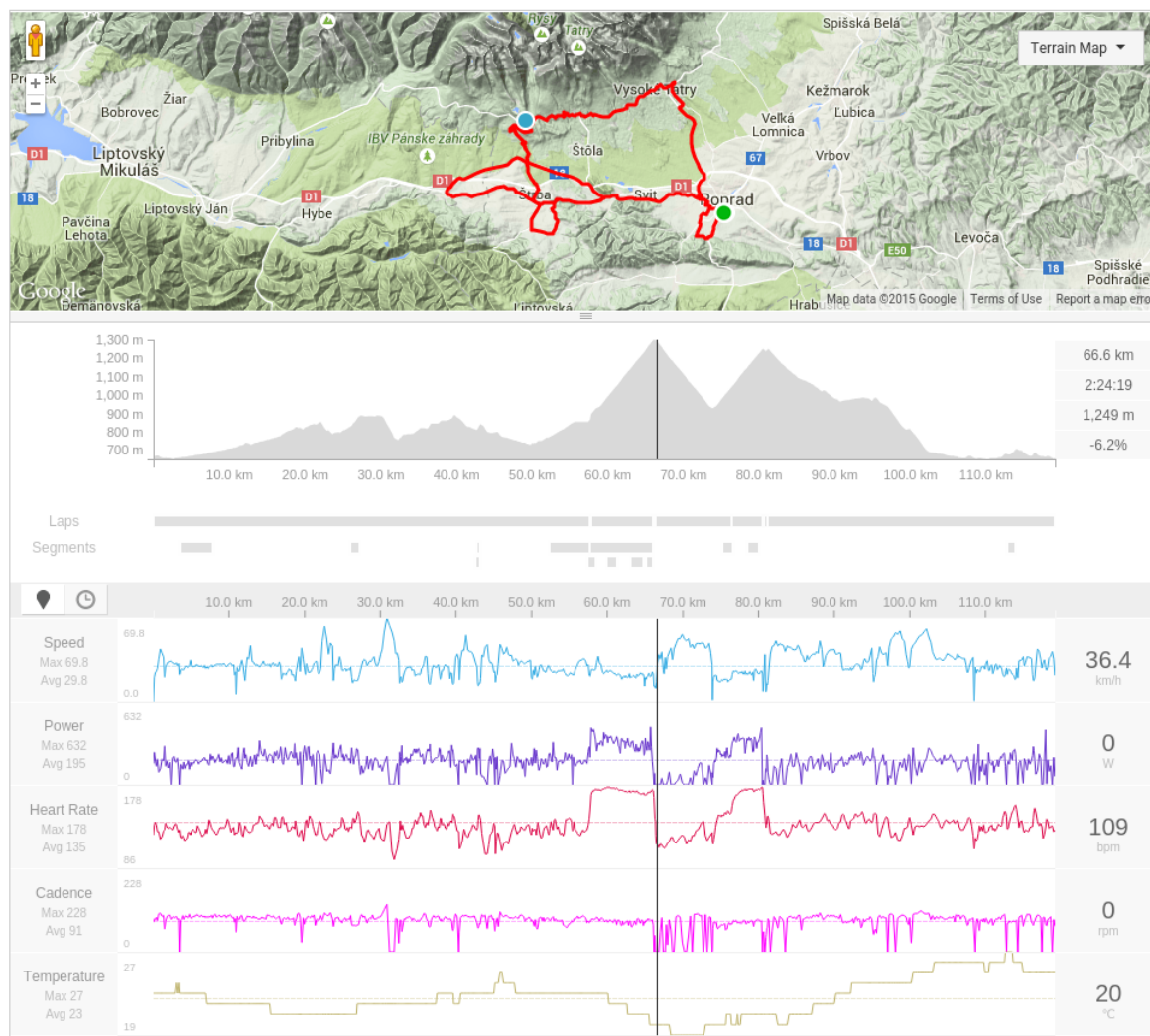
1.4 Systémy pre vyhodnotenie tréningu

1.4.1 Strava

Strava je portál, ktorý sa snaží združovať komunity športovcov po celom svete. Nosným pilierom Strava je webová aplikácia na adrese <http://www.strava.com>, ktorá ponúka komplexnú analýzu tréningových dát a množstvo sociálnych funkcií. Ako uvádza priamo Strava[19], používateľovi umožňuje zažiť tzv. *social fitness*, čo by malo predstavovať spojenie, porovnanie a súťaženie s ďalšími športovcami. Okrem webovej aplikácie je možné využiť aj mobilnú aplikáciu pre platformu Android a iOS. Primárne je Strava určená pre bežcov a cyklistov, grafy a analýza pre ich aktivity sú najpodrobnejšie. Tréningový denník si však môžu viesť aj ďalší športovci.

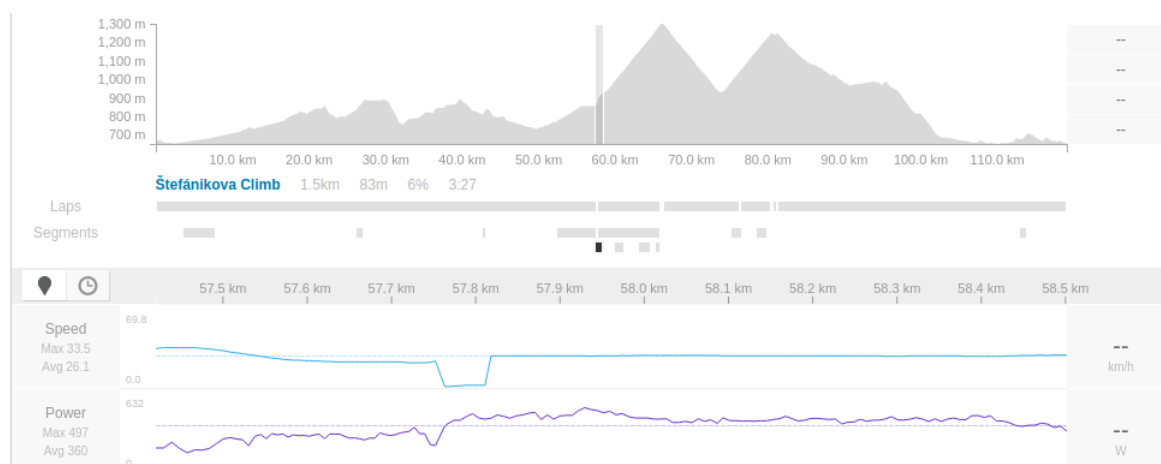
Princíp fungovania Strava môžeme popísať tromi krokmi. Prvým je absolvovanie tréningu a jeho zaznamenanie na zariadenie s GPS, resp. na ich mobilnú aplikáciu. Druhým krokom je uloženie tréningového záznamu na server Strava. Vďaka využitiu API výrobcov ako je Garmin, Suunto, TomTom, Timex a niektorých ďalších umožňuje Strava priame uloženie tréningových dát na server ihneď po pripojení zariadenia/športových hodínok do počítača. Priame ukladanie funguje na platformách Windows a OSX. Ak používateľ používa iný operačný systém alebo má záujem uložiť tréning z iného zariadenia, môže tak urobiť nahraním súboru, ktorý je vo formáte GPX, TCX alebo FIT. Tretím krokom je zobrazenie detailu tréningu a analýza. Údaje

môžeme prehliadať vo forme mapy alebo grafov, ktoré zobrazujú priebeh celého tréningu, hodnoty aktuálnej nadmorskej výšky, rýchlosti, tepovej frekvencie, frekvencie šliapania a teploty v jednotlivých pozíciách, vid' obr. 4.



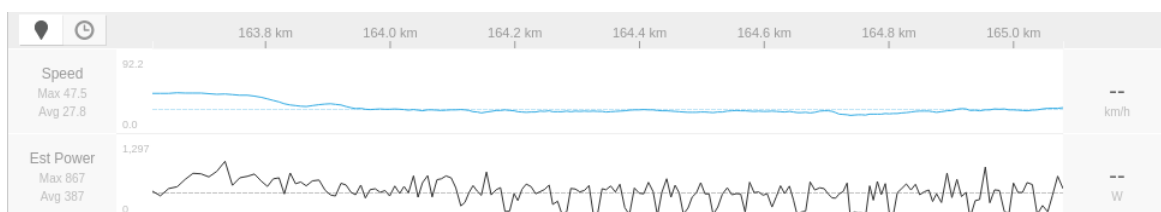
Obr. 4: Strava - Analýza tréningu

Portál Strava sme sa rozhodli popísať z dôvodu, že ako jeden z prvých prináša aproximáciu wattového výkonu cyklistu z GPS dát. Aby sme aproximáciu mohli analyzovať vzhľadom na naše ciele, absolvovali sme 3 rôzne merania na úseku Tatranská Štrba - Štrbské pleso v dĺžke 1500m, so snahou udržiavať rovnomerný výkon bez výkyvov. Pri prvom sme zmerali výkon zariadením Power Tap G3 a dáta zaznamenávali do cyklopočítača Garmin Edge 500. Na obr. 5 môžeme vidieť reálnu hodnotu výkonu.



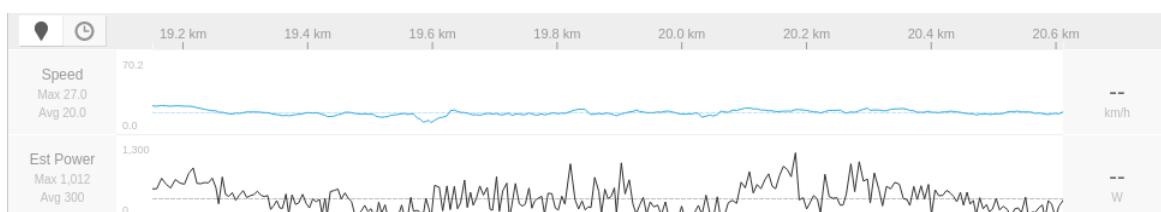
Obr. 5: Strava - Graf výkonu

Druhý tréning bol zaznamenaný tiež cyklopočítačom Garmin Edge 500 ale bez merania výkonu. Na obr. 6 môžeme vidieť výkon odhadnutý algoritmom Strava z rýchlosti a nadmorskej výšky. Môžeme pozorovať veľké výkyvy odhadnutého výkonu v rozmedzí od 0W do 867W, pričom dosahovaná rýchlosť je plynulá a bez výkyvov.



Obr. 6: Strava - Graf odhadnutého výkonu podľa GPS

Posledný záznam je meraný mobilnou aplikáciou Strava, vid' obr. 7. V tréningu sme výkon opäť nemerali a v grafe môžeme vidieť odhadnuté hodnoty. Obrovské výkyvy prisudzujeme odchýlkam v pozícii a aj nadmorskej výške. Použitý mobilný telefón Samsung Galaxy S4 mini nedisponuje barometrickým výškomerom, teda hodnota nadmorskej výšky je získaná z pozície. Ako je známe, GPS čip v mobilných telefónoch nedosahuje takú vysokú presnosť ako špeciálne športové zariadenia a pozícia je často korigovaná aj mobilnými sieťami.



Obr. 7: Strava - Graf odhadnutého výkonu podľa Assisted GPS

Z prezentovaných testov môžeme na prvý pohľad vyvodiť záver, že odhad výkonu portálom Strava nie je presný. Chybné sú najmä hodnoty, ktoré skáču z nuly do maximálnych hodnôt, a to aj napriek tomu, že športovec sa pohyboval takmer konštantnou rýchlosťou a snažil sa dodržať rovnakú frekvenciu šliapania. Hoci nepoznáme presný algoritmus výpočtu v Strava, myslíme si, že príčinou nepresnosti je zvolenie priveľmi malých úsekov, na ktorých sú vypočítavané čiastkové výkony, resp. sú vypočítavané medzi jednotlivými bodmi uloženými do zariadenia. Na malom úseku môže byť odchýlka v nadmorskej výške, nakoľko barometrický výškomer ju určuje podľa tlaku. Môže to byť samozrejme aj nepresnosťou GPS alebo malou frekvenciou ukladania jednotlivých bodov tréningu, pričom nie sú zachytené všetky zmeny terénu.

1.4.2 Bike Calculator

Druhý systém relevantný nášmu výskumu nájdeme na webovej stránke <http://www.bikecalculator.com>. Bike Calculator je údajne najpopulárnejšia kalkulačka výkonu cyklistu fungujúca od roku 1997 [20]. Už na prvý pohľad je jasné, že sa jedná iba o kalkulačku a žiaden vstup zo športových hodín alebo merača nie je možný. Stránka disponuje krátkym vysvetlením fyzikálneho problému výpočtu výkonu a formulárom, do ktorého je možné zadať hodnoty pre výpočet a výsledok je vyhodnotený okamžite.

Power:	210	watts	Velocity:	23.80	kph / mph
B I K E C A L C U L A T O R					
Units:	Metric ▼				
Rider weight:	70	kg / lbs			
Bicycle weight:	7	kg / lbs			
Tires:	Clinchers ▼				
Position:	Hoods ▼				
Grade:	2.3	%			
Head wind:	0	kph / mph			
Distance:	1500	km / mi			
Temperature:	33	C / F			
Elevation:	544	m / ft			

Obr. 8: Bike Calculator - Kalkulačka výkonu cyklistu

Otestovali sme ho pre 12 úsekov, z ktorých každý sme absolvovali 2-5 krát a výkon merali zariadením PowerTap G3. Pri každom meraní sa športovec snažil dodržiavať rovnomerný výkon. Nakoľko Bike Calculator neumožňuje vkladanie čiastkových údajov, použili sme priemer nameraných hodnôt pre každý úsek. V tabuľke 1 uvádzame porovnanie hodnôt z meraní a vypočítaných hodnôt.

Úsek	dĺžka [m]	priemer. stúpanie [%]	priemer. rýchlosť [km/h]	reálny výkon [W]	Bike Calculator [W]	Rozdiel	Rozdiel [%]
1.	1000	7.4	13.5	269	247	-22	-8.18%
2.	1000	8.5	13.0	285	269	-16	-5.61%
3.	1000	10.2	11.9	285	289	4	1.40%
4.	1500	4.9	26.1	267	261	-6	-2.25%
5.	1500	6.6	15.6	272	263	-9	-3.31%
6.	1500	9.2	11.7	273	258	-15	-5.49%
7.	2700	3.0	20.4	201	197	-4	-1.99%
8.	2700	4.8	24.1	357	346	-11	-3.08%
9.	2700	5.2	17.5	257	245	-12	-4.67%
10.	1500	0.7	30.5	180	214	34	18.89%
11.	1500	1.2	26.5	211	177	-34	-16.11%
12.	1500	2.3	23.8	187	210	23	12.30%

Tabuľka 1: Bike Calculator - porovnanie výpočtu

Z tabuľky si môžeme všimnúť, že celková dĺžka segmentu neovplyvňuje rozdiel medzi reálnou a vypočítavanou hodnotou. Zámerne však neuvádzame úseky dlhé iba niekoľko desiatok metrov, aby sme mohli poukázať na fakt, že práve stúpajúci sklon (priem. stúpanie) robí výpočet presnejším. Minimalizuje sa totiž aerodynamický odpor spôsobený vetrom, ktorého smer aj intenzitu je veľmi ťažké odhadnúť a stúpa podiel sily potrebnej na prekonanie zemskej tiaže.

Ako môžeme vidieť vo formulári na obr. 8, typ pneumatík a pozíciu športovca na bicykli vyberáme z predvolených hodnôt. Pre typ pneumatík sú to tri hodnoty, a to plášte, galusky alebo MTB⁷. Pozíciu môžeme zvoliť podľa bicykla - vzpriamená, mierne vzpriamená, zohnutá, aerodynamická. Podľa vybraného vstupu Bike Calculator použije predvolený koeficient valivého odporu, resp. aerodynamického odporu.

Koeficient aerodynamického odporu je však závislý aj od oblečenia cyklistu, od bicykla, od kolies. Koeficient valivého odporu je rozdielny pre jednotlivé modely plášťov, hodnotu ich nahustenia. Môžeme povedať, že je rozdielny pre každú jednu dvojicu športovec-bicykel. Dôkladnejšie vysvetlenie tejto problematiky uvádzame v kapitole 2.

⁷MTB - mountain bike - horský bicykel

2 Pôsobenie síl

Výkon, ktorý musí cyklista vyvíjať aby sa dokázal pohybovať určitou rýchlosťou je úmerný odporovým silám, ktoré na neho a bicykel počas jazdy pôsobia. V tejto kapitole sa budeme týmto odporovým silám venovať podrobnejšie, aby sme v dostatočnej miere poznali, na akých veličinách sú závislé. Tieto znalosti následne využijeme pri návrhu algoritmu strojového učenia. Jednotlivé zložky síl, ktoré pôsobia na cyklistu počas jazdy, sa menia v závislosti od zmeny terénu a poveternostných podmienok.

- Gravitačná sila
- Aerodynamický odpor
- Valivý odpor

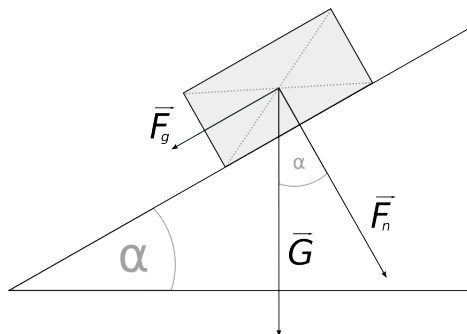
Kým aerodynamický odpor je najviac pociťovaný pri jazde po rovine, jeho podiel klesá spolu s narastajúcim sklonom vozovky a začína prevládať zložka gravitačnej sily. Valivý odpor je spôsobený kontaktom pneumatiky s povrchom vozovky, konkrétne ich deformáciou a mikrotrením.

Princíp superpozície hovorí, že výsledná sila pôsobiaca na teleso je rovná vektorovému súčtu všetkých síl pôsobiacich na dané teleso [7].

$$\vec{F} = \sum_i \vec{F}_i \quad (3)$$

2.1 Gravitačná sila

Pri pohybe cyklistu a bicykla v horizontálnej rovine je reakciou na gravitačnú silu normálová sila, ktorá zohráva úlohu pri valivom odpore (viď kapitola 2.3). Akonáhle však športovec prekonáva nadmorskú výšku, teda vo fyzikálnom pomenovaní sa pohybuje po naklonenej rovine, musí prekonať aj zložku gravitačnej sily F_g , ktorá pôsobí proti smeru jeho pohybu. Zložky gravitačnej sily sú zobrazené na obr. 9.



Obr. 9: Teleso na naklonenej rovine

Silu pôsobiacu proti pohybu cyklistu v smere z kopca vyjadríme vzťahom $F_g = m g \sin \alpha$. Kde m je hmotnosť sústavy, v našom prípade súčet hmotností športovca a bicykla a g je gravitačné zrýchlenie. Sínus pre uhol sklonu môžeme vypočítať zo zmeny nadmorskej výšky $\Delta altitude$ a prejdenej vzdialenosti d [7]:

$$F_g = m g \frac{\Delta altitude}{d} \quad (4)$$

Veľkosť normálovej sily, $F_n = m g \cos \alpha$, vyjadríme vzťahom 5 [7]

$$F_n = m g \sqrt{1 - \left(\frac{\Delta altitude}{d} \right)^2} \quad (5)$$

2.2 Aerodynamický odpor

Aerodynamický odpor je v poslednom období v cyklistike často preberanou tématikou. Upravujú sa tvary trubiek bicyklov, menia sa profily kolies, vylepšujú sa materiály a strih oblečenia. Všetko s cieľom dosahovania vyšších rýchlostí bez nutnosti zvyšovať výkon. Zo vzťahu 6 [15]

$$F_d = \frac{1}{2} \rho v^2 C_d A \quad (6)$$

môžeme vidieť, že sila potrebná na prekonanie aerodynamického odporu rastie s druhou mocninou rýchlosti. ρ je hustota prostredia. Referenčná plocha A je v tomto prípade premietnutá čelná plocha športovca a bicykla. Jedinou možnosťou ako znížiť odporovú silu je teda znížiť koeficient odporu C_d .

Koeficient odporu pre pohybujúce sa teleso C_d je závislý od povrchu a aerodynamického tvaru telesa, ďalej od hustoty a viskozity látky prostredia, v ktorom sa teleso pohybuje. Tiež závisí od rýchlosti a smeru pohybu telesa vzhľadom na smer pohybu látky prostredia (napr. smer vetra). Pre vysoké rýchlosti môžeme C_d vyjadriť ako funkciu Reynoldsového čísla Re a čísla $Mach$. Pre rýchlosti, ktoré dosahujú automobily alebo bicykle môžeme považovať C_d za konštantný. Ak uvažujeme nad aproximáciou výkonu, zisťujeme, že koeficient odporu ani referenčnú plochu nevieme jednoducho merať a teda ich hodnoty požadovať ako používateľský vstup do systému. Do úvahy pripadá ich odhad na základe známych proporcií športovca a reálne zmeraného výkonu.

Vo vzťahu 6 vystupuje hustota ρ , ktorá v našom prípade predstavuje hustotu vzduchu. Tá je pri 15°C na hladine mora rovná hodnote $1,2754 \text{ kg m}^{-3}$ [14]. Hustota

vzduchu sa však mení so stúpajúcou teplotou a tiež stúpajúcou nadmorskou výškou, keďže sa mení tlak vzduchu.

Atmosferický tlak p v nadmorskej výške h určíme podľa vzťahu 7,

$$p = p_0 \left(1 - \frac{L h}{T_0} \right)^{\frac{gM}{RL}} \quad (7)$$

v ktorom figurujú nasledovné konštanty [14]:

- p_0 - atmosferický tlak na hladine mora - 101,325 kPa
- T_0 - štandardná teplota vzduchu - 288,15 K = 15°C
- g - gravitačné zrýchlenie - 9,80665 m s⁻²
- L - teplotný gradient - 0,0065 K m⁻¹
- R - univerzálna plynová konštanta - 8,31447 J mol⁻¹ K⁻¹
- M - molová hmotnosť vzduchu - 0.028964 kg mol⁻¹

Pri známom tlaku vzduchu pre danú nadmorskú výšku, môžeme určiť hustotu vzduchu pri rôznej teplote T vzťahom 8

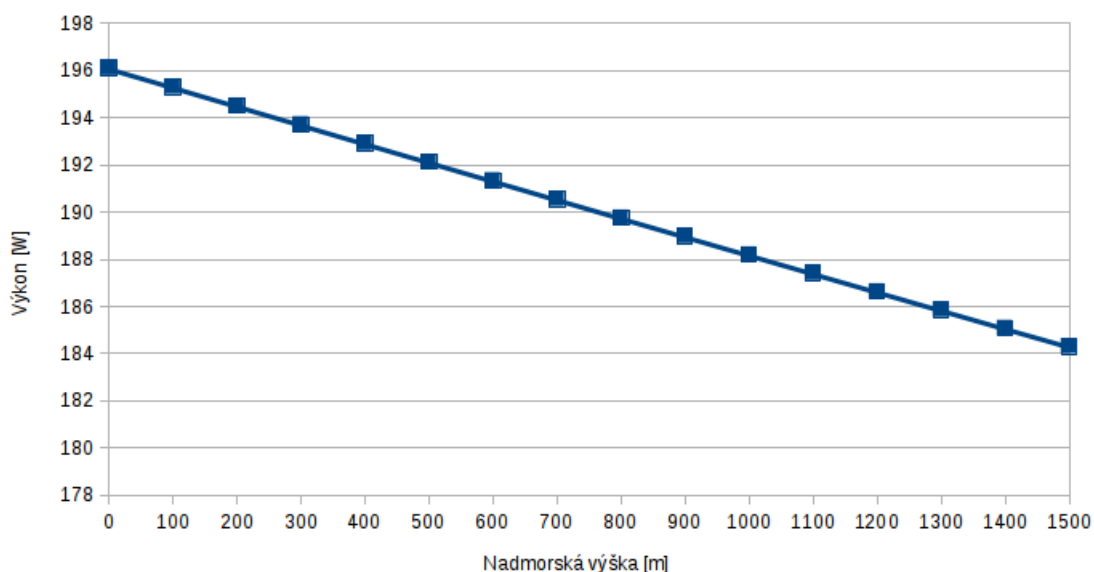
$$\rho = \frac{p M}{R T} \quad (8)$$

Zmena teploty a tlaku vzduchu ovplyvní aj výkon potrebný pre udržanie rovnakej rýchlosti, a to nezanedbateľným rozdielom. Pre priblíženie uvádzame dve simulácie pohybu cyklistu rýchlosťou 30 km h⁻¹ v horizontálnej rovine. V prvom prípade simulujeme pri konštantnej teplote zmenu nadmorskej výšky z 0 m n.m. do 1500 m n.m.

Simuláciu vykonávame s nasledovnými konštantami:

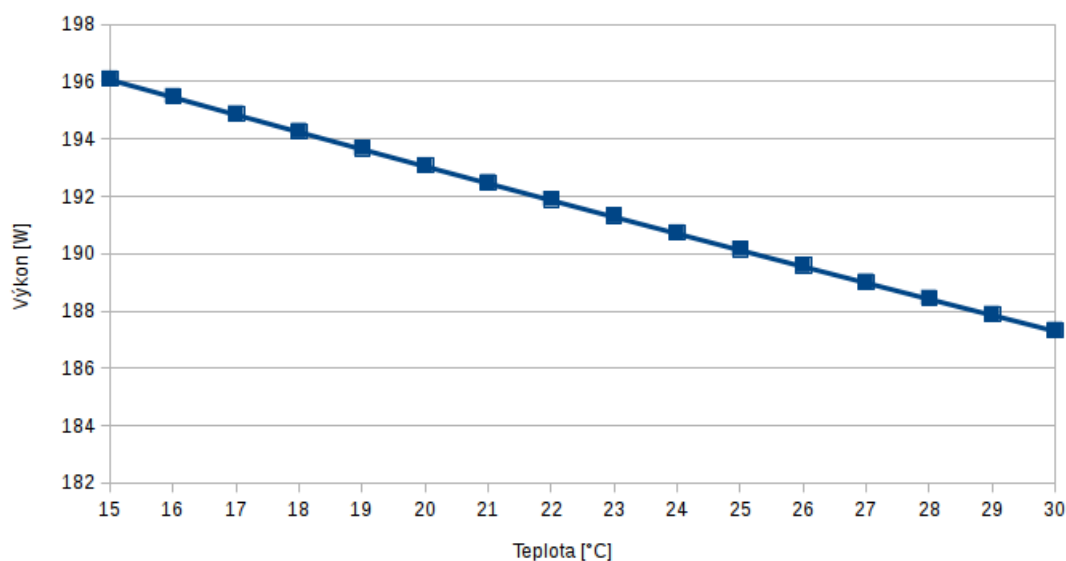
- $C_{rr} = 0.003$, na základe [11]
- $C_d = 1.0$, na základe [16]
- $A = 0.4$, na základe [16]

Z obr. 10 môžeme vidieť, že rozdiel v nadmorskej výške sa prejavil rozdielom 12 W vo výkone.



Obr. 10: Závislosť potrebného výkonu od nadmorskej výšky

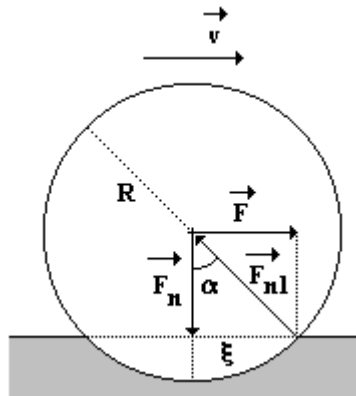
V druhom prípade pri konštantnej nadmorskej výške 0 m n.m. simulujeme zmenu teploty. Stúpajúca teplota vzduchu sa prejaví znížením hustoty vzduchu, a teda aj výkon potrebný na udržanie rovnakej rýchlosti klesá. Pri rovnakých podmienkach ako v predchádzajúcom príklade a teplote 30°C , by postačovalo cyklistovi na udržanie rýchlosti 30 km h^{-1} približne o 9 W nižší výkon než pri teplote 15°C , vid' obr. 11.



Obr. 11: Závislosť potrebného výkonu od teploty vzduchu

2.3 Valivý odpor

Valivým odporom môžeme nazvať silu, ktorá vzniká, keď sa teleso kruhového prierezu (v našom prípade koleso bicykla) valí po pevnej podložke. Fyzikálnou podstatou tohoto javu je hysterézia, ktorá nastáva pri deformácii podložky resp. kolesa pôsobením normálovej tlakovej sily. V reálnych podmienkach samozrejme podložka ani galuska kolesa nie sú dokonale pružné. Preto reakcia na normálovú tlakovú silu je posunutá o vzdialenosť ξ [7].



Obr. 12: Valivý odpor [7]

Silu F , ktorou predmet udržíme v rovnomernom pohybe vyjadríme vzťahom [7]:

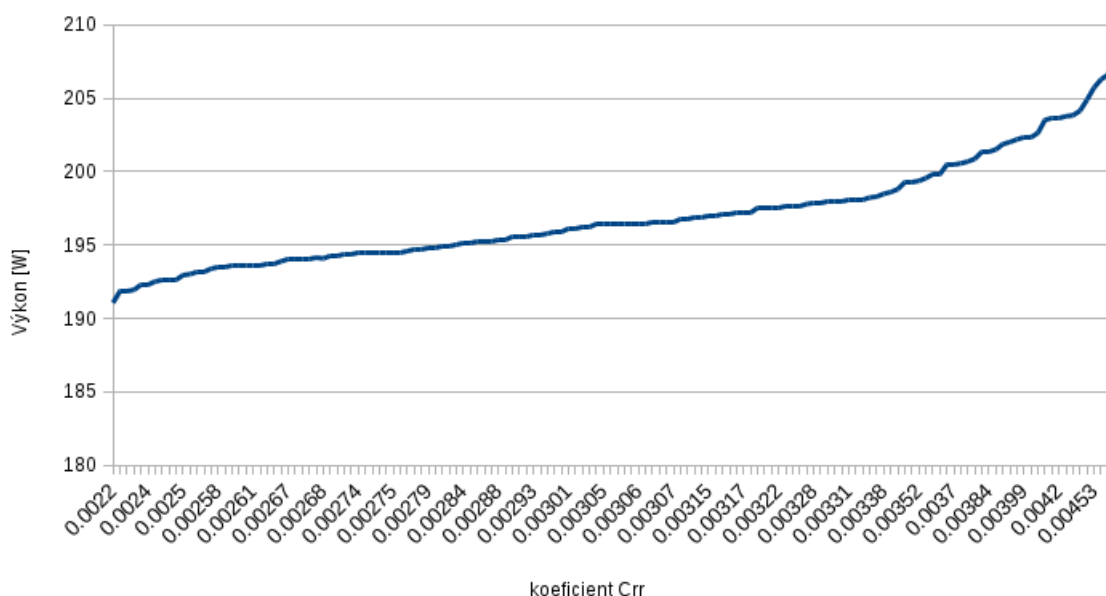
$$F = F_n \tan \alpha \quad (9)$$

Pre malé uhly je $\tan \alpha = \sin \alpha$ a pre koleso s polomerom R je $\sin \alpha = \frac{\xi}{R}$. Dosadením do vzťahu 9 dostávame:

$$F = \frac{\xi}{R} F_n \quad (10)$$

Normálovú silu na naklonenej rovine vypočítame podľa vzťahu 5.

V praxi by ale bolo veľmi náročné zmerať vzdialenosť ξ , preto sa podiel vo vzťahu 10 často nahrádza koeficientom valivého odporu C_{rr} . Koeficient môžeme nájsť pre rôzne typy pneumatík v tabuľkách. Definovaním štandardov pri testovaní pneumatík ako je napríklad štandard SAE J2452 [17] sa vymedzujú podmienky testu. Na týchto dátach môžeme vykonať ďalšiu simuláciu a prezentovať, aký vplyv má koeficient odporu C_{rr} na výkon potrebný pre udržanie konštantnej rýchlosti 30 km h^{-1} , viď obr. 13. Dáta sú simulované s konštantami: $A = 0.4$, $C_d = 1.0$, $T = 15^\circ\text{C}$, $h = 100\text{m n.m.}$



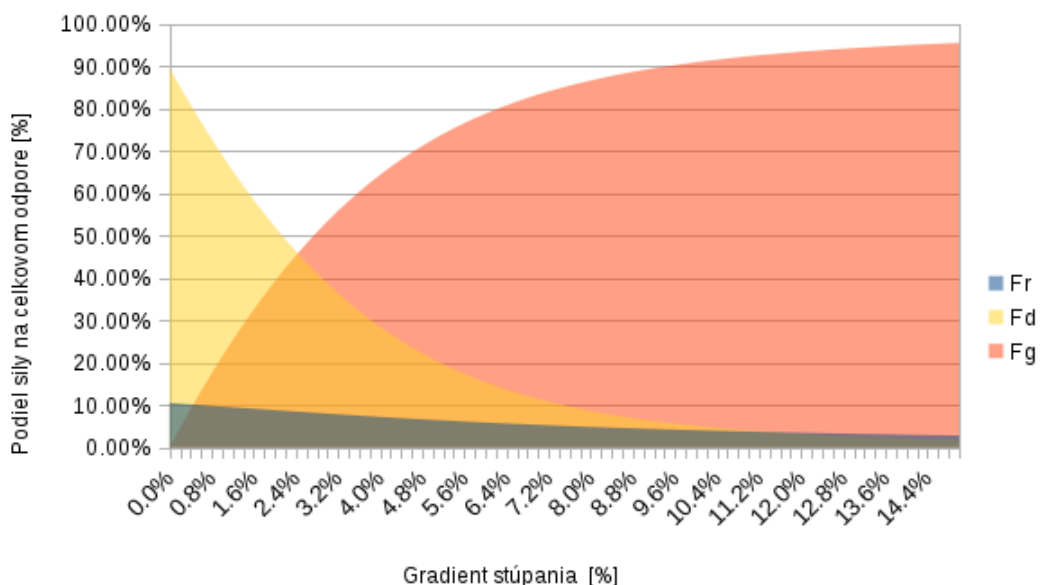
Obr. 13: Závislosť potrebného výkonu od koeficientu odporu pneumatiky

Kým koeficient je podľa tabuľky nezávislý od tlaku, pretože test predpokladá konštatné, ideálne nahustenie pneumatiky, v praxi tomu tak byť nemusí. Nakoľko je na trhu nespočetné množstvo modelov galusiek a plášťov pre bicykle, nie je možné k nim uchovávať a neustále aktualizovať koeficient valivého odporu C_{rr} . Určitú úlohu hrá aj tlak v pneumatike, ktorý každý cyklista volí podľa svojich osobných preferencií, požadovaných jazdných vlastností a komfortu. Žiadať tlak v pneumatike ako vstup do systému od používateľa považujeme za veľmi nekomfortné. Ohľadom dosadenia hodnoty za C_{rr} sa dostávame do rovnakej situácie ako pri koeficiente C_d .

2.4 Podiel jednotlivých zložiek na celkovom odpore

Na záver tejto kapitoly chceme prezentovať podiel jednotlivých silových zložiek na celkovej sile pôsobiacej na cyklistu v závislosti od stúpajúceho sklonu vozovky. Nakoľko dosiahnutie konštantných podmienok v teréne je takmer nemožné, opäť nám vhodne poslúži simulácia. S hodnotami $A = 0.4$, $C_d = 1.0$, $T = 15^\circ\text{C}$, $h = 0\text{m n.m.}$, $C_{rr} = 0.0045$, $m = 77\text{kg}$ a konštantným výkonom 350W simulujeme zmenu stúpania z 0% až do 15%

Z grafu na obr. 14 pozorujeme, že valivý odpor má po celú dobu minimálny podiel na celkovej sile, avšak nemôžeme ho zanedbať. A tiež, že podiel gravitačnej sily a aerodynamického odporu sa vyrovnáva približne pri 2.5% stúpaní.



Obr. 14: Podiel silových zložiek v závislosti od stúpajúceho sklonu vozovky

Ďalšou skutočnosťou, ktorá súvisí najmä s aerodynamickým odporom je vietor. Smer a intenzitu vetra počas tréningu by bolo veľmi náročné určiť. Hoci by sme mohli využiť API predpovede počasia, získané údaje nevieme garantovať pre rôzne segmenty trasy, keďže vietor sa môže lámať medzi budovami alebo stromami. Ďalším faktorom je aj veľkosť skupiny, v ktorej cyklista tréning absolvuje. Je samozrejmé, že v závetrí väčšej skupiny športovec šetrí silu. To, ako presne sa tieto dva faktory odzrkadľujú na výkone, už povedať nevieme a jediná cesta dosiahnutia presných výsledkov by bola testovanie vo veternom tuneli. Takéto testovanie by však bolo nielen časovo ale aj finančne náročné, a preto sme v rámci rozsahu našej práce uprednostnili čo možno najpresnejšiu aproximáciu v známych podmienkach.

V našej aproximácii budeme predpokladať, že smer a intenzita vetra, má zanedbateľný vplyv na výkon, resp. pri dlhších a kľukatých úsekoch je v niektorých častiach jeho intenzita pričítaná a v iných zase odčítaná z celkového aerodynamického odporu. Výkon sa môžeme snažiť aproximovať v úsekoch kedy podiel aerodynamického odporu na celkovej pôsobiacej sile je čo najnižší. Dobré výsledky by mohli byť dosahované pri aproximácii v stúpaniach nad 3% priemerného sklonu, kedy sa podiel gravitačnej sily a aerodynamického odporu vyrovnáva.

3 Opis riešenia

Doposiaľ prezentovanú štúdiu nenecháme iba v teoretickej rovine, ale implementujeme časť webovej aplikácie, ktorá používateľovi umožní nahranie súboru s tréningovými dátami na server, odhadne podávaný výkon a zobrazí dáta vo forme grafov a mapy. Výsledná funkčná časť bude nadväzovať na aplikáciu z našej bakalárskej práce *Webový organizér*. Spoločne budú tvoriť funkčný celok s možnosťami komplexnej analýzy tréningu.

Hlavným cieľom tejto práce je aproximácia podávaného výkonu na základe údajov z nepriamych meraní. Už pred začatím prác bola na mieste otázka, či je možné odhadovať výkon v tak premenlivých podmienkach, aké môžeme pri pohybe na bicykli vo vonkajšom prostredí uvažovať. Na základe empirických skúseností sme usúdili, že s výnimkou extrémnych podmienok v okolitom prostredí, by mohla byť aproximácia relatívne presná. Následne bolo otázne, či údaje, ktoré počas tréningu zaznamenáme, postačovať na to, aby výsledok aproximácie mohol do určitej miery konkurovať cenovo náročnému komponentu, meraču výkonu. Pri analýze sme zohľadnili možné nepresnosti:

- pri určovaní polohy cez GPS a teda aj nepresnosť vo vzdialenosti a rýchlosti
- pri zázname zmeny nadmorskej výšky barometrickým výškomerom, pri zmenách tlaku, teploty
- pri zázname zmeny nadmorskej výšky cez GPS, nepresnosť pri určovaní polohy
- vo vzorových dátach, odchýlka zmeraného a skutočného výkonu (vyrobcami uvádzaná odchýlka 2%)
- vplyv zotrvačnosti pri krátkych zmenách v rýchlosti pohybu.

Po dôslednejšom preskúmaní dát, ktoré sú z meračov k dispozícii sme žiaľ museli priustiť, že nie sú natoľko presné, aby sme z nich mohli aproximovať krátke niekoľko sekundové intenzívne výkony. Výkon pri krátkych zrýchleniach (3 až 5 sekúnd) pri súčasnej presnosti dát nedokážeme odhadnúť s uspokojujúcou presnosťou. Uvažovali sme teda o rozdelení priebehu tréningu na úseky dlhé v priemere 40 až 150 m a pracovať s priemerným stúpaním a rýchlosťou, od čoho sme si sľubovali odstránenie nepresností pri výpočte.

Popísaním odporových síl pôsobiacich na cyklistu (vzťahmi 4, 6, 9) sme zistili, že k fyzikálnemu výpočtu výkonu by boli potrebné koeficienty C_d , A , C_{rr} , ktoré nevieme

zistiť jednoduchým meraním, ani ich globálne zovšeobecniť pre všetkých športovcov. Ako vhodné sa javí použitie strojového učenia, podľa [1], oblasti umelej inteligencie, ktorá sa snaží automaticky hľadať zmysluplné vzory v dátach.

3.1 Strojové učenie

Podstatou strojového učenia je využitie automatizovaných procedúr založených na logických alebo binárnych operáciách, pomocou ktorých sa stroj z dát učí a zároveň dáta predvída. V strojovom učení je dôležitý pojem zovšeobecnenie⁸, v zmysle schopnosti primerane reagovať na nové vstupné vzorky, po rozpoznaní vzorov v tréningovej množine. Algoritmy strojového učenia pracujú s vytvoreným modelom vstupov a majú za cieľ predvídať alebo klasifikovať výstupy. Podľa [1], na základe druhu učenia rozoznávame tri základné typy strojového učenia:

- *Supervised* - učenie so vzorom - k vektoru vstupných tréningových dát je daný vzorový vektor výstupov.
- *Unsupervised* - učenie bez vzoru - k vstupným dátam nie je daný výstup.
- *Reinforcement* - učenie so spätnou väzbou - na základe vstupných dát je vykonávaná akcia, jej vhodnosť je bodovo hodnotená. Algoritmus upraví stratégiu s cieľom získať čo najvyššie bodové hodnotenie. Používaný hlavne v robotike.

Existuje veľké množstvo konkrétnych algoritmov, preto ich nebudeme individuálne menovať, ale posnažíme sa obmedziť výber na základe nášho cieľa. Ak by sme chceli získať chýbajúce koeficienty C_d , A , C_{rr} museli by sme zvoliť *unsupervised learning* - učenie bez vzoru, pretože tréningové dáta pre tieto koeficienty nemáme k dispozícii. V tomto smere sa však nenaskytoval žiaden vhodný postup, ktorým by sme mohli nájsť koeficienty ako výstup z funkcie vstupov.

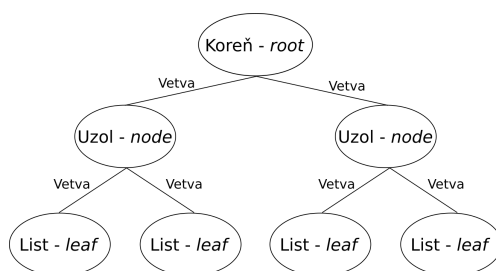
Vhodnejšie sa teda javí zvoliť učenie so vzorom. Zostavíme model vstupov (dostupných tréningových dát) a pre ne, ako vzor výstupu, zodpovedajúci výkon cyklistu. Pri výbere vhodných tréningových dát, správnej konfigurácii a tréningu, by algoritmus mohol byť schopný rozpoznať vzťahy medzi vstupmi a vzorovým výstupom a v praxi potom aproximovať výkon cyklistu iba na základe vstupov z nepriamych meraní.

Ak z úvah vylúčime algoritmy strojového učenia, ktoré sa používajú na klasifikáciu vzoriek do tried, a zameráme sa na tie, ktorými môžeme aproximovať funkciu, pripadajú do úvahy *Umelé neurónové siete* a *Rozhodovacie regresné stromy*.

⁸zovšeobecnenie - z angl. generalization

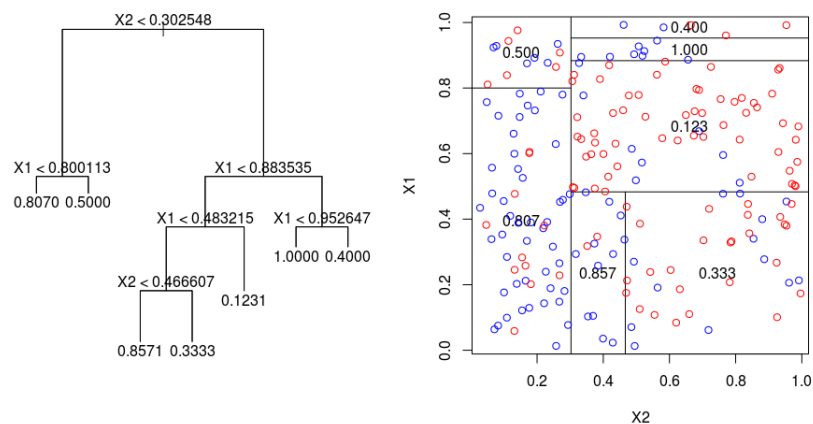
3.1.1 Rozhodovacie regresné stromy

Ako uvádza [2], klasifikačné a regresné stromy sú typom strojového učenia ktoré využívajú štruktúru rozhodovacích stromov. Cieľom klasifikačných stromov je priradenie vstupu do definovaných tried, regresné stromy majú za úlohu určiť numerickú hodnotu výstupu na základe vstupných dát. Vstupné dáta postupne prehádzajú štruktúrou stromu na základe definovaných kritérií a podmienok, až sa dostanú do tzv. listu - *leaf*, ktorý reprezentuje výstupnú hodnotu. Koreň reprezentuje vstupnú hodnotu. Podmienky sú reprezentované uzlami a vetvami. Pre ilustráciu uvádzame obrázok 15.



Obr. 15: Rozhodovací strom

Cieľom regresného stromu je čo najpresnejšie určiť výstupnú hodnotu \mathbf{Y} na základe vstupného vektora \mathbf{X} . Výstupná hodnota \mathbf{Y} patrí množine reálnych čísel R . Pomocou algoritmov implementujúcich regresný strom sa množina reálnych čísel rozdelí na menšie množiny, ktoré reprezentujú výstupnú hodnotu. Na obrázku č.16 je znázornený príklad regresného stromu a rozdelenie množiny výstupných hodnôt na základe tohto stromu. Pri týchto výstupoch musíme rátať s istou odchýlkou výstupnej hodnoty.

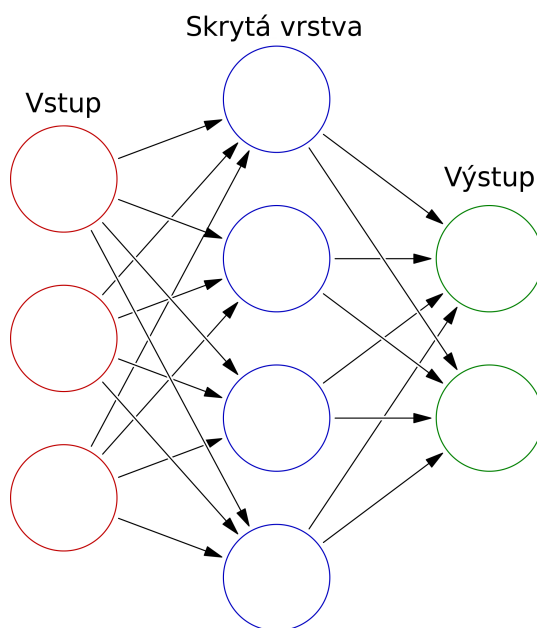


Obr. 16: Rozdelenie množiny hodnôt

Regresný strom má ako rekurzívny model dve hlavné výhody: interpretovateľnosť a modelovanie interakcií. Naproti tomu, jeho veľkými nevýhodami sú nestabilita a obtiažnosť zachytiť jednoduché vzťahy[2]. Majú tendenciu mať vysokú variabilitu, v tom zmysle, že veľmi malá zmena v dátach môže produkovať veľmi odlišné rozdelenie hodnôt. Príčinou toho je, že akákoľvek zmena v hornej úrovni stromu sa šíri do dolných častí a ovplyvňuje ostatné vetvy.

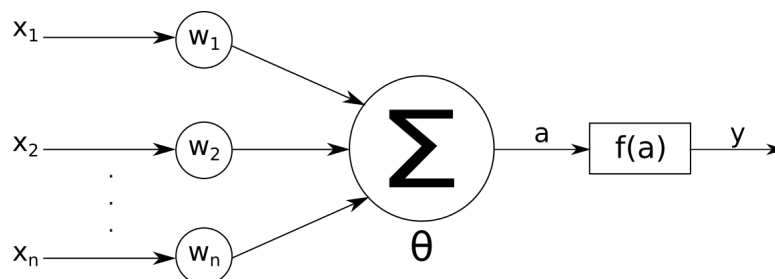
3.1.2 Umelé neurónové siete

Umelé neurónové siete sú modelom výpočtu inšpirovaný štruktúrou ľudského nervového systému. Nervový systém v zjednodušenom ponímaní pozostáva z veľkého počtu základných výpočtových jednotiek - **neurónov** navzájom pospájaných **dendritmi** do komplexnej siete, v ktorej medzi sebou komunikujú. Umelá neurónová sieť je formálna výpočtová konštrukcia založená na tejto štruktúre. Môžeme ju popísať ako orientovaný graf, ktorého uzly zodpovedajú neurónom a hrany medzi nimi reprezentujú dendrity. Spoločným znakom umelých neurónových sietí je množina adaptívnych hrán, ktoré sa upravujú tréningom. Práve vďaka adaptívnym spojovacím hranám, má neurónová sieť schopnosť učenia sa. Aplikácie neurónových sietí nájdeme pre viacero problémov, veľmi často sa okrem klasifikácie do tried používajú aj na aproximáciu vo všeobecnosti neznámych funkcií viacerých vstupov.



Obr. 17: Neurónová sieť

Neurónová sieť je zložená zo vstupnej a výstupnej vrstvy. Voliteľne, podľa zložitosti konkrétneho problému, môže mať sieť aj jednu a viac skrytých vrstiev. Sieť bez skrytej vrstvy sa nazýva *perceptrón* a používa sa na klasifikáciu vstupov do dvoch tried[1]. Vstupná vrstva pokrýva počet vstupov/atribútov, distribuuje vstupné signály do ďalšej vrstvy siete. Počet neurónov vo výstupnej vrstve zodpovedá žiadanému počtu výstupov. Skrytá a výstupná vrstva majú výpočtové funkcie.



Obr. 18: Neurón

Každý neurón prijíma ako vstup vážený súčet výstupov neurónov pripojených k jeho vstupným hranám. Výstup neurónu zohľadňuje prah citlivosti θ neurónu a *aktivačnú funkciu* $\mathbf{f}(\mathbf{a})$, čo môžeme popísať vzťahom 11. Niektoré zdroje, napr. [3] to referujú ako *kombinačná funkcia*. Vstupom do kombinačnej funkcie je vektor hodnôt \mathbf{x} , výstupom skalárna hodnota \mathbf{y} , w_i reprezentujú váhy vstupných hrán.

$$y = f\left(\sum_{i=1}^n (w_i x_i) - \theta\right) = f(a) \quad (11)$$

Cieľom pri učení neurónovej siete je opakovaným spracovaním tréningových vzorov a úpravou váh minimalizovať rozdiel (chybu) medzi výstupom siete a požadovaným (vzorovým) výstupom. Množstvo algoritmov používaných na tréning neurónovej siete je založených na výpočte gradientu chyby. Algoritmom spätného šírenia chyby sa vypočítava podiel váhy hrany na celkovej chybe, na základe ktorého sú jednotlivé hrany postupne upravované.

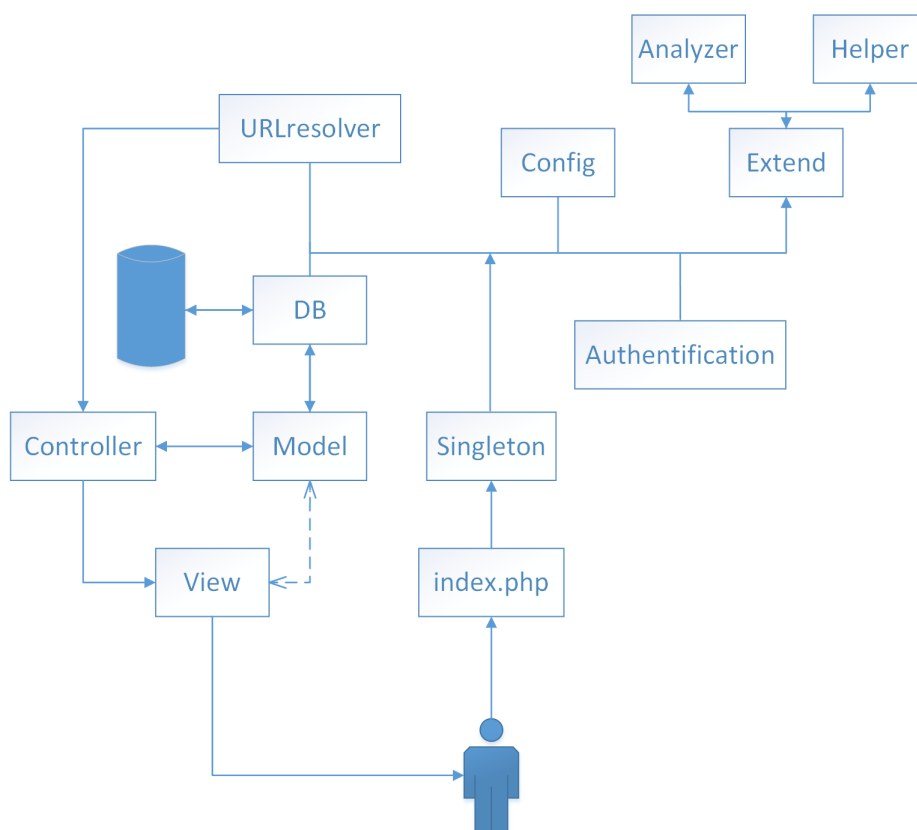
Nevýhodou neurónových sietí je, že sú niečo ako čierna skrinka, so vstupmi a výstupmi, ale je náročné skúmať ako daný problém riešia. Čo má dopad aj na programovú stránku. Riešenie nesprávnej funkčnosti siete a hľadanie chýb je komplikované. Tréning je do veľkej miery závislý na správnom nastavení siete a výbere vzorových dát, pričom konfigurácia siete si často vyžaduje postup metódou *pokus-omyl*. Aj napriek nevýhodám je ich veľkou silou schopnosť nájsť vzťah medzi komplikovanými alebo nepresnými dátami, čo umožňuje ich použitie aj na problémy, ktoré nie sú riešiteľné

človekom alebo inými počítačovými technológiami.

3.2 Použité technológie a riešenia

Bakalárska práca *Webový organizér*, predstavovala webovú aplikáciu s možnosťou manuálneho ukladania absolvovaného aj plánovaného tréningu s možnosťami sumarizácie tréningového objemu pre vybrané obdobie. Ponúkala aj jednoduché sociálne funkcie, ako napríklad vyhľadávanie priateľov a nastavenie súkromia. Serverová strana aplikácie *Webový organizér* je naprogramovaná v jazyku PHP s použitím frameworku *Yii*. Ako databázový server používa MySQL. Klientská časť stojí na jazyku HTML, JavaScript a samozrejme CSS.

V nadstavbe, ktorá je vytvorená v rámci diplomovej práce, sa o serverovú stranu stará rovnako programový kód v jazyku PHP, avšak pre naše potreby sme vytvorili vlastný minimalistický framework. Vytvorený framework implementuje návrhový vzor *Model-View-Controller* a *Singleton*. Presný popis štruktúry ani fungovanie frameworku uvádzať nebudeme, nakoľko nie je predmetom tejto práce. Zjednošenú architektúru vytvorenej aplikácie uvádzame na obr. 19.



Obr. 19: Architektúra webovej aplikácie

Nová časť aplikácie úplne preberie na starosť ukladanie a zobrazovanie tréningov. Oproti bakalárskej práci je pozmenená možnosť uloženia manuálneho záznamu tréningu a zobrazenie tréningov. Navyše je pridané ukladanie tréningu s nahraním dátového súboru a jeho analýza, tj. zobrazenie dát v grafoch a aproximácia výkonu.

Na ukladanie dát využívame existujúce schémy z bakalárskej práce v MySQL databáze, ktoré sme len mierne modifikovali, pridali niektoré tabuľky a niekoľko nových stĺpcov spolu s cudzími kľúčmi.

Údaje o tréningu, ktoré aplikácia prijme vo forme FIT súboru, konvertujeme použitím *FIT SDK conversion tool* do CSV a uložíme do databázy. Nakoľko hodnoty sú športovými hodinkami počas tréningu ukladané v približnom intervale 1-5 sekúnd, pracujeme so súbormi, ktoré obsahujú od 1 000 do 30 000 jednotlivých bodov. Každý obsahuje informácie o polohe, čase, rýchlosti, nadmorskej výške, teplote, tepovej frekvencii, frekvencii šliapania, výkone atď. Spočiatku sme sa obávali veľkosti týchto dát a počtu riadkov v príslušnej databázovej tabuľke. Pre prvotné účely aplikácie, by však MySQL databáza mala byť postačujúca. V produkčnom móde môže byť tabuľka pre tréningové dáta nahradená dátovým úložiskom *Big Table* alebo iným.

Na odhad výkonu sme sa rozhodli implementovať doprednú neurónovú sieť, ktorú považujeme pre tento prípad aproximácie funkcie ako najvhodnejší prístup. Zároveň s touto oblasťou máme predchádzajúce praktické skúsenosti.

Výstup serverovej časti aplikácie je vo forme HTML5, ktorý na strane klienta do finálneho zobrazenia dotvorí programový kód v jazyku JavaScript. Jedná sa najmä o vykreslenie grafov, čo umožňuje knižnica *D3.js*[22].

Jednoduchý, prezentovateľný dizajn vytvoríme v CSS s použitím frameworku *Bootstrap*[21], vďaka ktorému aplikácia získa responzívny dizajn.

3.3 Prípady použitia

Nižšie uvádzame popisy jednotlivých prípadov použitia, ktoré pokrýva naša aplikácia. Diagram s prípadmi použitia je zobrazený na obr. 20.

UC_01 Prihlásenie do systému

- Popis: prihlásenie používateľa do systému prostredníctvom jeho emailu a hesla
- Vstupné podmienky pre používateľa: správne heslo a email použitý pri registrácii
- Chybový tok: nesprávne vstupné údaje, používateľ môže akciu opakovať
- Zmena stavu systému: prihlásenie a presmerovanie na požadovanú alebo hlavnú

stránku

UC_02 Manuálne uloženie tréningových údajov

- Popis: športovec uloží údaje o športovej aktivite
- Vstupné podmienky pre používateľa: UC_01, zadanie dátumu v ktorom bol tréning vykonaný, názvu, trvania
- Chybový tok: chýbajúce údaje je nutné opraviť
- Zmena stavu systému: zobrazenie tréningu UC_04

UC_03 Uloženie tréningových údajov zo športového merača

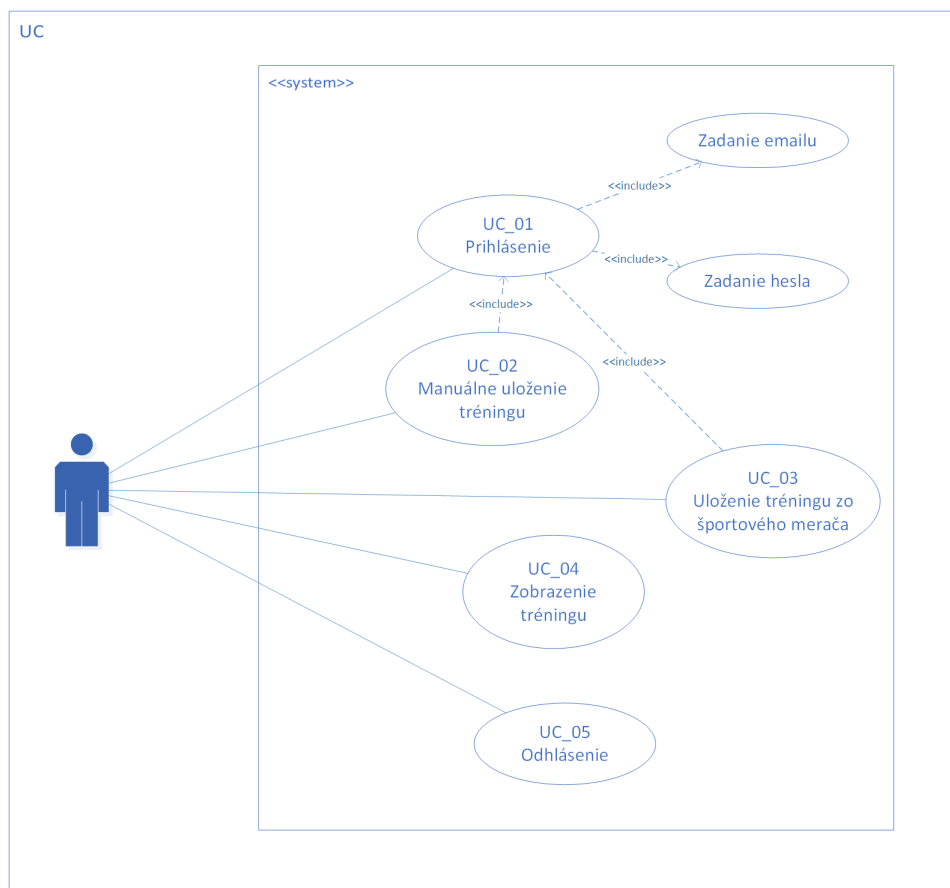
- Popis: športovec uloží údaje o tréningu nahraním dátového súboru (FIT, GPX, TCX)
- Vstupné podmienky pre používateľa: UC_01, nahranie minimálne jedného dátového súboru
- Chybový tok: môže nastať, ak sa niektorý súbor nepodarí nahráť, nepodarí sa ho konvertovať, nepodarí sa načítať údaje. Chybný súbor je možné nahráť znovu.
- Zmena stavu systému: zobrazenie tréningu UC_04

UC_04 Zobrazenie tréningu

- Popis: detailný náhľad uloženého tréningu, ak záznam o tréningu obsahuje aj údaje zo športového merača, zobrazia sa informácie vo forme grafu a mapy
- Vstupné podmienky pre používateľa: požiadavka na detailné zobrazenie konkrétneho tréningu
- Chybový tok: nemôže nastať
- Zmena stavu systému: načítanie a zobrazenie aktivity

UC_05 Odhlásenie

- Popis: Odhlásenie zo systému
- Vstupné podmienky pre používateľa: požiadavka na odhlásenie
- Chybový tok: nemôže nastať
- Zmena stavu systému: vymazanie session používateľa

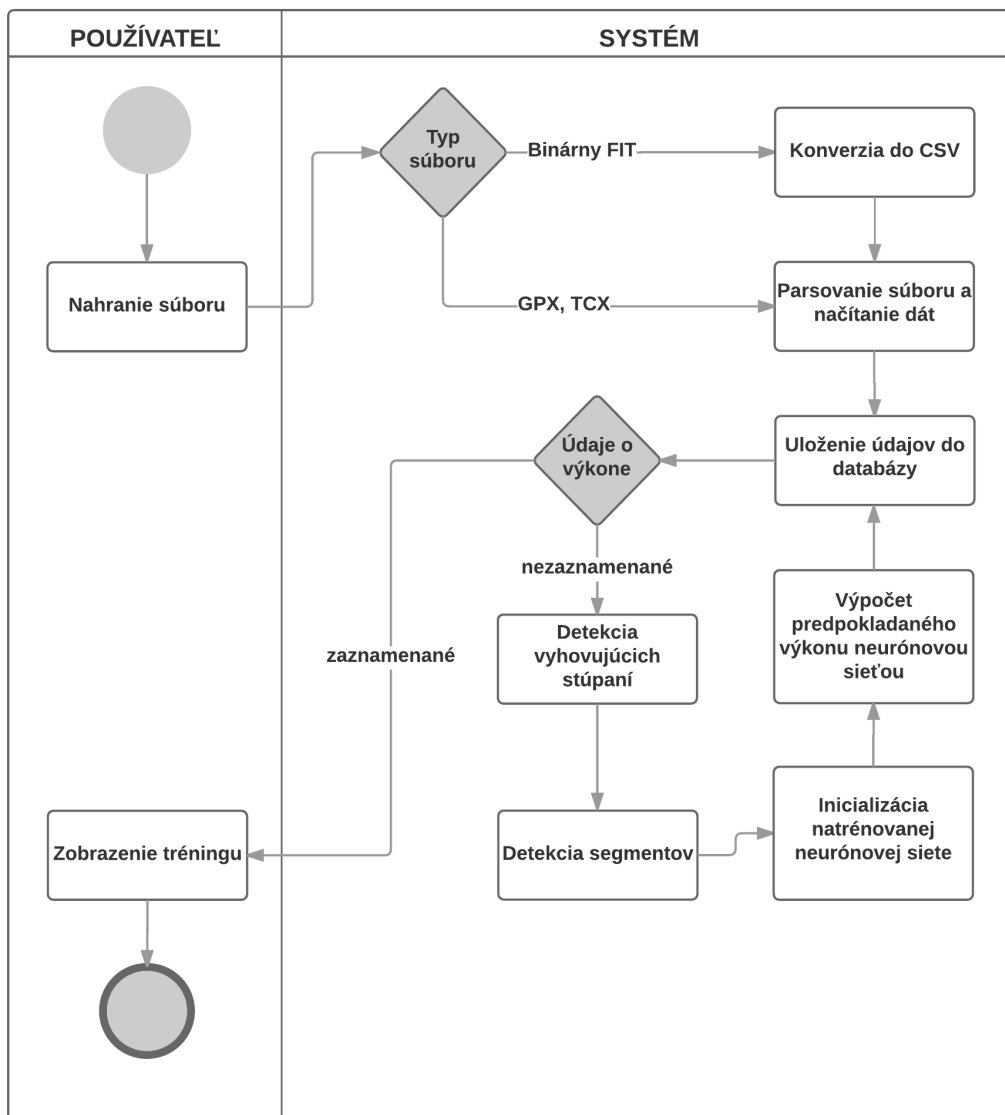


Obr. 20: Diagram prípadov použitia

3.4 Diagram aktivít

V tejto časti uvádzame diagram pre priebeh najdôležitejšej aktivity v aplikácii, konkrétne prípadu použitia *UC_03*. Ako sme spomenuli v predchádzajúcej časti, po nahraní dátového súboru používateľom na server nasleduje jeho rozbor a načítanie dát. V prípade, že sa jedná o binárny FIT súbor, načítaniu predchádza konverzia súboru do CSV formátu. Tok v systéme pokračuje uložením údajov do databázy. Ak bol tréning absolvovaný s meračom výkonu, záznam z tréningu sa zobrazí používateľovi. Ak výkon medzi údajmi chýba, detekujú sa segmenty, na ktorých bude odhadovaný výkon. Následne sa inicializuje vopred natrénovaná neurónová sieť a údaje zo segmentov sa postupnej posielajú na jej vstup. Výstup siete sa pridá medzi údaje o tréningu, a tie sa uložia do databázy. Pri ďalšom zobrazení tréningu výpočet cez neurónovú sieť už neprebíha, použijú sa uložené dáta. Postupnosť krokov je ilustrovaná na nasledujúcom diagrame aktivít.

ULOŽENIE TRÉNINGU ZO ŠPORTOVÉHO MERAČA



Obr. 21: Diagram aktivít

3.5 Implementácia

V tejto časti podrobne preberieme implementáciu algoritmu pre aproximáciu výkonu cyklistu. V predchádzajúcich častiach sme sa zaoberali úvahami o presnosti rôznych odhadov, vysvetlili sme výpočet a ukázali podiel jednotlivých zložiek na celkovej sile potrebnej pri pohybe cyklistu. Táto štúdia nás viedla k záveru, že dostatočnú presnosť môžeme dosiahnuť pri aproximácii výkonu v úsekoch, v ktorých je priemerný sklon aspoň 3%, celková dĺžka úseku aspoň 500 metrov a násobok sklonu v číselnom vyjadrení a dĺžky úseku je väčší ako 40. Môže teda ísť o kratšie, prudšie úseky alebo

dlhšie s miernejším stúpaním. Hraničné príklady úsekov sú napr.:

- dĺžka 500 m, priemerný sklon 0.08
- dĺžka 1 334 m, priemerný sklon 0.03

Empirickým overením sme dospeli k záveru, že už pri úsekoch tejto dĺžky a sklonu, rýchlosť získaná zotrvačnosťou prispieva k pohybu zanedbateľne v premietnutí na celý sledovaný úsek. Rýchlosť, ktorou sa športovec v stúpaní pohybuje, je teda získaná výhradne podávaním jeho výkonu.

3.5.1 Nahrание a konverzia súboru

Skôr, než môžeme samotné dáta analyzovať, potrebujeme umožniť používateľovi aby ich mohol na server nahráť. V našom frameworku máme implementované mapovanie cesty v URL adrese na príslušný ovládač a jeho metódu. To znamená, že cesta `/upload/files/` spúšťa metódu `files()` v triede `UploadController`. Pokiaľ pri tejto požiadavke nie sú metódou `POST` odoslané žiadne súbory, iba využijeme metódu frameworku `render()` a zobrazíme náhľad (*view*) s formulárom pre nahrание súboru.

```
if( !empty($_FILES) ){
    /* Spracovanie súborov prebehne tu */
}
else{
    /* V opačnom prípade sa zobrazí formulár */
    $this->render('files', array() );
}
```

Aplikácia umožňuje nahráť viac súborov súčasne. Jediné obmedzenie je na strane servera, v konfigurácii PHP treba obmedziť parametre `upload_max_filesize` a `post_max_size` na požadovanú veľkosť. Pred presunutím súboru na server sa skontroluje, či sa jedná o jeden z podporovaných súborov (FIT, GPX, TCX) a PHP funkciou `move_uploaded_file` je následne presunutý do zložky `/uploads/activities_raw/`, pod názvom s nasledujúcim tvarom: `IDpoužívateľa_hashNázvu.priponaSúboru`.

Ak server prijal binárny súbor formátu FIT, ihneď po jeho presunutí vykonáme konverziu použitím nástroja na konverziu z *FIT SDK*.

Ďalším krokom v metóde `files()` je spracovanie CSV súboru (konvertovaného z FIT) alebo XML súborov (GPX, TCX). K tomu je určená trieda `ActivityParser`. Na konci metódy `files()` je ešte potrebné uložiť údaje do databázy. Sumarizáciu tréningu uložíme do tabuľky `training_entry` cez model `WorkoutModel` a priebeh tréningu z dátového

súboru do tabuľky *training_entry_record* cez model *ActivityModel*. Týmto je nahratie súborov ukončené a používateľovi sa zobrazí stránka, na ktorej môže zmeniť typ športu, pridať tréningu názov a popis.

```
$w_model = new WorkoutModel();

$w_model->id_user = $user_id;
$w_model->start_time = $session_data->start_time + EPOCH_TIMESTAMP_OFFSET;

/* Kvôli stručnosti vynechané priradzovanie niektorých údajov */

$w_model->total_timer_time = $session_data->total_timer_time;
$w_model->total_elapsed_time = $session_data->total_elapsed_time;

if( $w_model->save(TRUE) ){
    $this->renderPartial(
        'update.form',
        array('model' =>$w_model, 'activities' =>$activities),
        'workout'
    );
}
```

3.5.2 Detekcia stúpaní

Aby mohla byť vykonaná samotná aproximácia výkonu, musí byť z priebehu tréningu detekované, kde sa začínajú a končia stúpania, ktoré vyhovujú stanoveným podmienkam (minimálne 3% sklon a 500 m dĺžka). V skutočnosti je priebeh tréningu síce spojitý, no hodnoty, s ktorými pracujeme sú diskrétné. Najjednoduchšou možnosťou detekcie je teda iterácia cez všetky body a overovanie nadmorskej výšky. Aby sme zjednotili údaje, ktoré reprezentujú vybraný úsek v tréningu a udržali programový kód konzistentný, vytvorili sme jednoduchú triedu *SegmentModel*

```
class SegmentModel {

    /** @var int Začiatkový index segmentu v dátach */
    protected $index_start;

    /** @var int Koncový index segmentu v dátach */
    protected $index_end;

    /** @var float Dĺžka segmentu */
    protected $length;

    /** @var float Rozdiel v nadmorskej výške */
    protected $elevation;

    /** Konštruktor segmentu */
    public function __construct(
        $index_start, $index_end, $length, $elevation
    ) {
```

```

        $this->index_start = $index_start;
        $this->index_end   = $index_end;
        $this->length      = $length;
        $this->elevation    = $elevation;
    }

    /** Vypočíta sklon stúpania pre segment */
    public function grade(){
        return $this->elevation / $this->length;
    }

    /** ... get metódy */
}

```

Detekciu úsekov má na starosti trieda *Analyzer*. V prvom kroku metódy *detectClimbs()* sú detekované všetky neklesajúce úseky, následne sú nevyhovujúce odstránené.

```

public function detect_climbs( WorkoutModel $workout ){
    /** ... Načítanie dát */

    for( $i = 0; $i < count($this->data); $i++ ){
        if( empty($this->data[$i]['distance']) ){ continue; }

        $start = [ /* index, distance, altitude */ ];
        while($this->data[$i]['altitude'] < $this->data[$i+1]['altitude'] ){
            $i++;
        }
        $end = [ /* index, distance, altitude */ ];

        $climbs[] = new SegmentModel(
            $start['index'], $end['index'],
            $end['distance'] - $start['distance'],
            $end['altitude'] - $start['altitude']
        );
    }

    /** Odstráni nevyhovujúce stúpania
    foreach ( $climbs as $k => $climb ){

        if( !($climb->get_length() >= 500
            && $climb->grade() >= 0.03
            && $climb->get_length() * $climb->grade() > 40 ) )
        {
            unset($climbs[$k]);
        }
    }
}

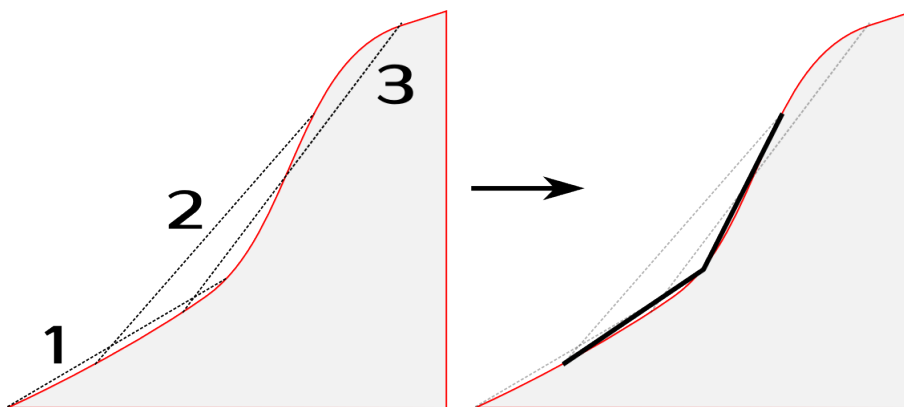
```

V tomto momente máme detekované stúpania, v ktorých chceme výkon aproximovať. Ideálna by bola aproximácia, ktorá by pre každý uložený bod v týchto stúpaniach určila konkrétnu hodnotu výkonu. Vyrátavali by sa teda priemerné výkony medzi jednotlivými bodmi, ktoré boli uložené na tréningu v priebehu stúpaní. Týmto postupom sme však nedostahovali uspokojivé výsledky, nakoľko nepresnosť GPS a barometrického výškomera pri výpočte znásobovala odchýlku od reálneho výkonu. Tento problém sme videli aj pri aproximácii výkonu portálom *Strava*. Z tohto dôvodu v

ďalšom kroku delíme stúpania na menšie časti, tzv. segmenty, na ktorých je konštantné stúpanie a pri snahe športovca o čo najplynulejšiu jazdu predpokladáme aj rovnomernú rýchlosť.

Hľadanie úsekov s konštantným stúpaním nevykonávame jednoduchým prechádzaním cez všetky dáta a vzájomným porovnávaním aktuálneho a nasledujúceho bodu. Lepšie rozdelenie stúpania dostávame hľadaním, pri ktorom vytvárame úsečku z bodu N a $N+x$, kde x je vopred definovaný krok. Túto úsečku následne porovnáme s úsečkou vytvorenou medzi $N+1$ a $N+x+1$. Náročnosť algoritmu je rovnaká ako v prípade, kedy by sme porovnávali body medzi sebou jednotlivo, cyklus prejde cez všetky body. Vytváraním dlhších úsečiek však minimalizujeme odchýlky výškomera. Kým prvou spomenutou metódou by sme dostávali úseky tvorené 2 až 3 bodmi, použitou metódou je to v priemere 5 až 10 bodov, tj. v priemere 40 až 150 m, v závislosti od rýchlosti pohybu a frekvencie ukladania dát.

Princíp detekcie segmentov pre lepšiu predstavivosť zobrazujeme na obr. 22. Každá úsečka je vytvorená medzi bodmi N a $N+x$, kedy N je navyšované pri každej iterácii. Sklon úsečky 1 je porovnaný so sklonom úsečky 2. Ak je rozdiel sklonu väčší ako 0.005 predpokladáme, že ležia nad samostatnými segmentami. Pri tomto porovnávaní však môžeme natrafiť na prípad, kedy by nasledujúci segment mal začínať pod úsečkou, presne ako na obr. 22, pod úsečkou 2. Preto po nájdení úsečiek s rozdielnymi sklonmi ešte rozdelíme aj samotnú úsečku a celý proces porovnania sklonov zopakujeme, kým nie je určený koniec jedného segmentu a začiatok nového.



Obr. 22: Detekcia segmentu

```
private function detect_segments( SegmentModel $climb ){
    $step = 10;
    for( $i = $i+1; $i <= $climb->get_index_end() - $step; $i++ ){
        /** Spočíta gradient
```



```

    $length = $this->data[$i+$step]['distance']
              - $this->data[$i]['distance'];
    $elevat = abs($this->data[$i+$step]['altitude']
              - $this->data[$i]['altitude']);
    $grade = $elevat / $length;
    if( abs($grade - $grade_prev)>0.005 && $i-$seg_start>$step){
        /* Rozdelenie segmentu */
        /* Pokračuje v detekcii presného začiatku a konca */
    }
    $seg = new SegmentModel( /* Potrebné hodnoty */ );
}

```

3.5.3 Zobrazenie tréningu

Hlavnou požiadavkou pri zobrazení tréningu s GPS údajmi, je možnosť prehliadať absolvovaný tréning na mape a možnosť zobrazíť jednotlivé body ľahko čitateľnou formou, teda ideálne formou grafu.

Prvú požiadavku splníme vďaka API od Google Maps. Aplikačné rozhranie umožňuje zobrazíť a vycentrovať mapu pre konkrétnu polohu alebo množinu bodov, vykresliť body alebo ich spojením priamo vytvoríť krivku. Prikreslením ďalšej krivky do mapy môžeme zvýrazniť konkrétny úsek v tréningu podľa priblíženia/selektie v grafe. Do mapy je taktiež možné pridať ukazovateľ pozície. O vykreslenie mapy s týmito možnosťami sa postará nasledovný kód, ktorý sa nachádza v náhľadovom súbore */views/workout/view.chart.php*. Keďže sa jedná o Javascript, je len odoslaný do prehliadača a svoju funkciu vykoná až na strane klienta. Výsledok je následne zobrazený na obr. 23

```

<script type="text/javascript">
function map_initialize() {
    /* ... definovanie štýlu a nastavení mapy */

    style = new google.maps.StyledMapType(map_styles);
    route_coordinates = [<?php echo $coordinates ?>];

    route_path = new google.maps.Polyline({
        path: route_coordinates,
        geodesic: true,
        strokeColor: '#33CC99',
    });

    map = new google.maps.Map(
        document.getElementById('map-canvas'), map_options
    );

    map_marker_mov = new google.maps.Marker({
        /* ... Nastavenie pre ukazovateľ pozície */
    });

    map_bns = new google.maps.LatLngBounds();

```

```

<?php echo $bounds; ?>//Body tréningu cez ktoré sa mapa centruje

map.mapTypes.set('map_style', style);
map.setMapTypeId('map_style');
map.fitBounds(map_bns); // Vycentrovanie mapy
route_path.setMap(map); // Vykreslenie krivky
}

function map_selection(dt){
    for(var i = 0; i < dt.length; i++){
        selection_coordinates.push(
            new google.maps.LatLng( dt[i].position_lat, dt[i].position_long )
        );
        selection_bounds.extend(
            new google.maps.LatLng( dt[i].position_lat, dt[i].position_long )
        );
    }

    /* Reset predchádzajúceho zobrazenia */
    selection_path.setMap(null);

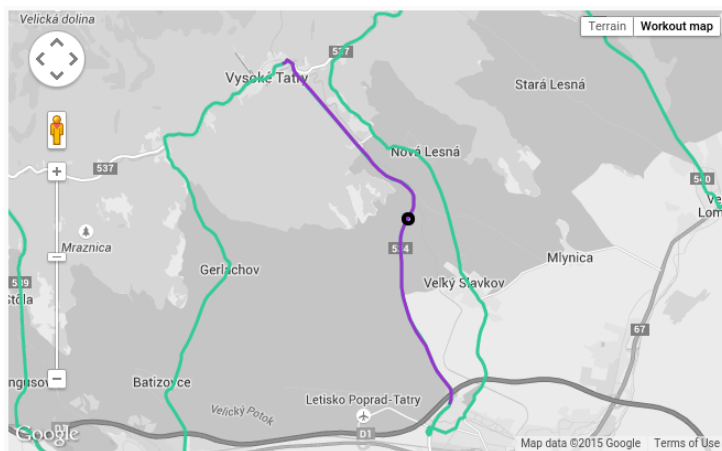
    selection_path = new google.maps.Polyline({
        path: selection_coordinates,
        geodesic: true,
        strokeColor: '#9933cc',
    });

    map.fitBounds(selection_bounds);
    selection_path.setMap(map);
}

function map_reset(){
    map.fitBounds(map_bns);
    selection_path.setMap(null);
}

google.maps.event.addDomListener(window, 'load', map_initialize);
</script>

```



Obr. 23: Zobrazenie mapy v detailnom náhľade tréningu

Ďalšia časť programového kódu v tomto náhľade sa postará o vykreslenie grafu.

Štandard HTML5 umožňuje do HTML kódu vkladať SVG grafiku. Na zobrazenie priebehu celého tréningu použijeme knižnicu *D3.js*. Knižnica uľahčuje prácu s jednotlivými elementami SVG, nakoľko s nimi pracuje ako s objektami, ktoré majú vlastné metódy. Programový kód vykresľujúci kompletný graf s priebehom tréningu je pomerne rozsiahly, preto uvedieme len najdôležitejšie časti.

```

1 var chart_def = {<?php echo $chart_def; ?>};
2
3 var charts_count, i;
4 charts_count = i = Object.keys(chart_def).length;
5
6 var root = d3.select("#chart-canvas").append("svg")
7     .attr("width", WIDTH )
8     .attr("height", HEIGHT * i + MARGINS.top + MARGINS.bottom);
9
10 chart_draw(chart_data);

```

Premennú *\$chart_def* inicializujeme ešte v PHP, obsahuje definície veličín, ktorá sa majú zobraziť, tj. ku ktorým existujú dáta. Táto premenná obsahuje aj jednotky pre jednotlivé veličiny a počet desatinných miest pri zobrazení. Riadok 6 zabezpečí vloženie elementu *<svg>* do HTML s fixnou výškou a šírkou. O vykreslenie sa stará funkcia *chart_draw()*.

```

function chart_draw(chart_data){

    /* X rozsah podľa celkovej vzdialenosti*/
    x_range = d3.scale.linear()
        .range([MARGINS.left, WIDTH-MARGINS.right])
        .domain([
            d3.min(chart_data, function(d) {return d.distance;}),
            d3.max(chart_data, function(d) {return d.distance;})
        ]);

    /** Inicializacia funkcii
    for(var key in chart_def){

        /* ... Definovanie funkcií pre získanie maxima, minima a priemeru*/
        min[key], max[key], avg[key];

        /* Y rozsah podľa maxima a minima */
        y_range[key] = d3.scale.linear()
            .range([HEIGHT*i, HEIGHT*(i+1)])
            .domain(
                [min[key] - 0.05 * min[key], max[key] + 0.05 * max[key]]
            );

        /* Funkcia pre vykreslenie priebehu */
        var draw_fn = d3.svg.line()
            .x(function(d) {return x_range(d.distance);})
            .y(function(d) {
                return
                    y_range[key](d[key]) || y_range[key](min[key]);
            });

        chart_def[key].draw_fn = draw_fn;
    }
}

```

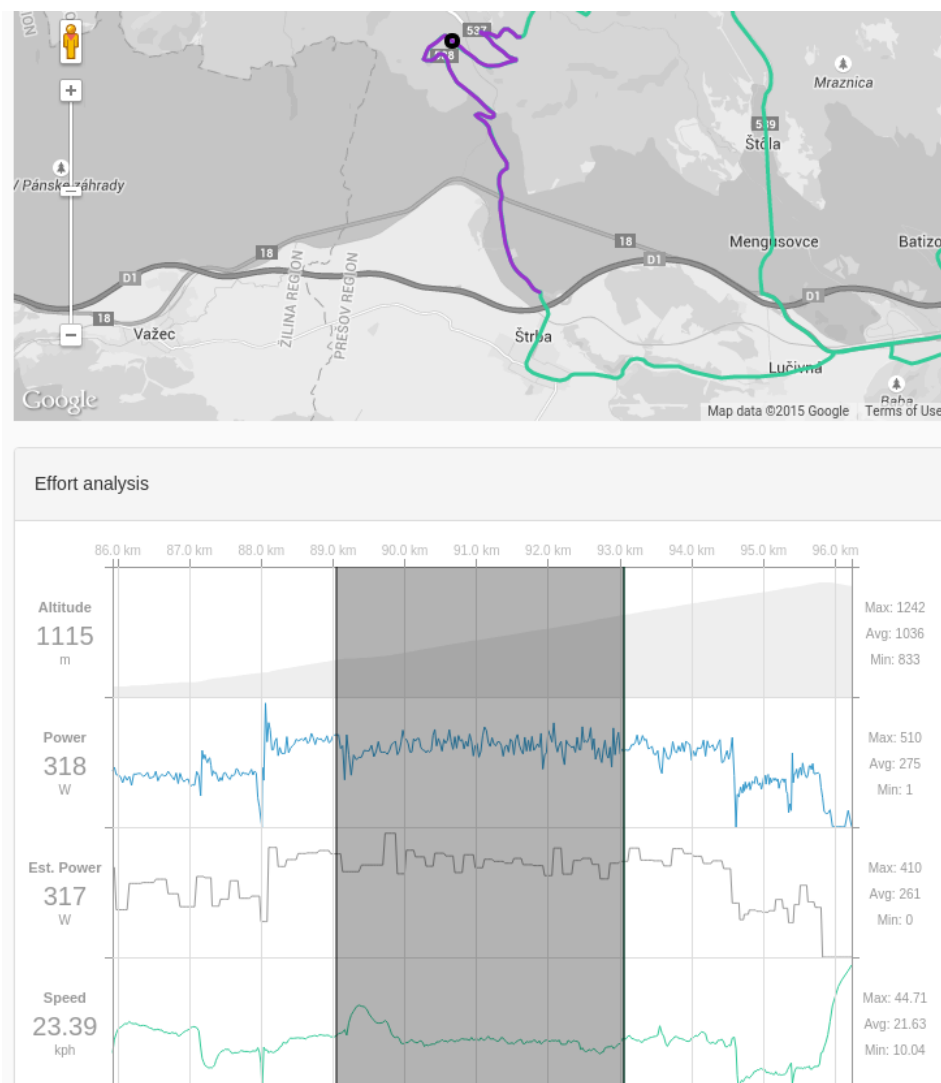
```

chart_def[key].x_range = x_range;
chart_def[key].y_range = y_range[key];
i--;

/** Vykreslenie krivky
svg.append('svg:path')
    .attr('d', chart_def[key].draw_fn(chart_data))
    .attr("class","datapath " + key)
    .attr("transform", "translate(" + (-MARGINS.left)+ ",0)");
}

```

Vykresľovanie osí grafu nebudeme rozvádzať dopodrobna. Graf disponuje interaktivitou. Po prechode myšou na konkrétnu pozíciu sa vedľa grafu zobrazia hodnoty pre daný body a na mape sa zobrazí ukazovateľ pozície. Kliknutím na graf a ťahaním je možné vybrať časť tréningu. Graf sa priblíži a na mape sa zvýrazní vybraná časť rozdielnou farbou. Vid' nasledujúci obrázok.



Obr. 24: Zvýraznenie výberu pred priblížením konkrétnej časti tréningu

3.5.4 Implementácia neurónovej siete

Ako sme uviedli v predchádzajúcich častiach práce, na aproximáciu výkonu sme sa rozhodli implementovať doprednú neurónovú sieť. Základným komponentom neurónovej siete je neurón, ktorý má určitý počet vstupov a váhy vstupujúcich hrán. Na základe toho máme jednoduchú triedu neurón, ktorej konštruktor sa postará o náhodnú inicializáciu váh v intervale $<-1, 1>$:

```
class Neuron {
    /** @var int Počet vstupov do neurónu */
    public $numInputs = 0;

    /** @var array Váha pre každý vstup do neurónu */
    public $weights = [];

    /** @param int $numInputs */
    public function __construct($numInputs) {
        $this->numInputs = $numInputs;

        //Váha pre prah neurónu - preto +1
        for($i = 0; $i <= $this->numInputs; $i++){
            array_push($this->weights, Helper::random(-1,1));
        }
    }
}
```

Neurónová sieť pozostáva z vrstiev. Každá skrytá a výstupná vrstva má určitý počet neurónov na základe konfigurácie siete. Konštruktor triedy *NeuralLayer* sa postará o vytvorenie neurónov obsiahnutých vo vrstve:

```
class NeuralLayer {
    /** @var int Počet neurónov vo vrstve */
    public $numNeurons = 0;

    /** @var array of Neuron - Neurony vo vrstve */
    public $neurons = [];

    /**
     * @param int $numNeurons
     * @param int $numInputsPerNeuron
     */
    public function __construct($numNeurons, $numInputsPerNeuron) {
        $this->numNeurons = $numNeurons;

        for ($i = 0; $i < $numNeurons; $i++) {
            array_push($this->neurons, new Neuron($numInputsPerNeuron));
        }
    }
}
```

Inštanciu neurónovej siete vytvárame z triedy *NeuralNet*. Skrátenú definíciu uvádzame na nasledujúcich riadkoch.

```

class NeuralNet {
    // atribúty
    /** @var int Počet vstupov siete */
    private $numInputs;
    /** @var int Počet výstupov siete */
    private $numOutputs;
    /** @var int Počet skrytých vrstiev */
    private $numHiddenLayers;
    /** @var array Počet neurónov v skrytej vrstve */
    private $numNeuronsPerLayer = [];
    /** @var array Pole všetkých vrstiev siete aj s výstupnou vrstvou */
    private $layers = [];
    /** @var array Vstupy pre aktuálny výpočet */
    private $inputs = [];
    /** @var array výsledok výpočtu count($outputs) == $numOutputs */
    private $outputs = [];
    /** @var array Viacrozmerne pole s výstupmi pre každú vrstvu */
    private $partialOutputs = [];

    //metódy
    /** Konštruktor */
    public function __construct();
    /** Vytvorenie neurónovej siete */
    public function createNet();
    /** Vráti váhy siete */
    public function getWeights();
    /** Nastaví váhy */
    public function putWeights($weights);
    /** Vypočíta výstup zo vstupov */
    public function compute($in);
    /** Trénovanie siete */
    public function train($trainData);
    /** Spúšťa normalizáciu dát pred výpočtom
    public function preProcessing($data);
    /** Spúšťa prevod dát na pôvodné hodnoty, po výpočte */
    public function postProcessing($data);
    /** Počíta gradient chyby spätným šírením */
    private function computeDeltas($outputs, $desired);
    /** Upraví váhy v priebehu tréningu */
    private function updateWeights($deltas, $learnRate);
    /** error function */
    private function meanSquaredError($output, $desired);
    /** Volá aktivačnú funkciu podľa konfigurácie vrstvy */
    private function transfer($layer, $value);
    /** Volá deriváciu aktivačnej funkcie */
    private function deriveTransfer($layer, $value);
}

```

Metóda *compute()* je implementácia vzorca 11, preto ju nebudeme detailnejšie rozvádzať. Výstupy každej vrstvy siete sú uložené do poľa *partialOutputs*. Aktivačnú funkciu **f(a)** neurónu je možné zvoliť v konfigurácii pre každú vrstvu individuálne, keďže sme z dôvodu testovania implementovali hyperbolický tangens aj sigmoidu. Nakoľko sa výstupné dáta nenachádzajú v obore hodnôt týchto funkcií, výstupná vrstvu používa lineárnu aktivačnú funkciu.

3.5.5 Trénovanie neurónovej siete

Na trénovanie neurónovej siete sme zvolili algoritmus spätného šírenia chyby⁹. Trénovanie je možné vykonávať inkrementálne alebo dávkovo. Pri inkrementálnom sú váhy neurónov upravované po vypočítaní chyby každej vzorky dát. Naopak, dávkové učenie upravuje váhy až po prejdení celej množiny trénovacích dát. Po otestovaní oboch spôsobov sme sa priklonili ku inkrementálnemu učeniu, pretože výstupy rýchlejšie konvergovali k očakávaným hodnotám, ako to odporúčal aj zdroj [3]. Inkrementálne trénovanie sa vykonáva v nasledujúcich krokoch:

1. Pre každú trénovaciu vzorku vykonaj:
 - Výpočet výstupu $\mathbf{y}(x_i)$
 - Výpočet chyby
 - Korekcia váh
2. Krok 1 opakuj kým nie je dosiahnutý \mathbf{N} počet epoch alebo nie je splnená iná ukončovacia podmienka

Výpočet celkovej chyby vykonávame v metóde *meanSquaredError()* pomocou nasledujúceho vzťahu [3]:

$$E = \sum \frac{1}{2}(\text{desired} - \text{output})^2 \quad (12)$$

Po ukončení jednej epochy urobíme priemer chýb každej vzorky a táto hodnota je jednou z ukončovacích podmienok tréovania.

Spätné šírenie a úprava váh. Cieľom spätného šírenia je úprava váh siete tak, aby bol vypočítaný výstup bližšie k vzorovému výstupu. Výstupný signál šírime sieťou späťne a počítame podiel konkrétnej váhy na celkovej chybe. Úpravu váhy vykonávame podľa nasledujúceho vzťahu [3]:

$$w_i(t) = w_i(t-1) - \eta \frac{\partial E}{\partial w_i} \quad (13)$$

Kde, $w_i(t)$ je nová váha hrany i ,

$w_i(t-1)$ je pôvodná váha hrany i ,

η je stupeň učenia¹⁰ definovaný v konfigurácii,

⁹Error-backpropagation algorithm

¹⁰learning rate

$\frac{\partial E}{\partial w_i}$ je podiel váhy hrany i na celkovej chybe.

Ak hrana w_i vstupuje do j -teho neurónu k -tej vrstvy, potom podiel $\frac{\partial E}{\partial w_i}$ vypočítame na základe:

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial out_{kj}} * \frac{\partial out_{kj}}{\partial net_{kj}} * \frac{\partial net_{kj}}{\partial w_i} \quad (14)$$

kde, out_{kj} je výstup j -teho neurónu k -tej vrstvy,

net_{kj} je výsledok vnútornej aktivity j -teho neurónu k -tej vrstvy, na obr.18 ako **a**.

Signál chyby šírieme spätne, začína na výstupnej vrstve, pre ktorú čiastkové výpočty počítame nasledovne:

$$\frac{\partial E}{\partial out_{kj}} = -(desired_j - output_j) \quad (15)$$

$\frac{\partial out_{kj}}{\partial net_{kj}}$ je derivácia aktivačnej funkcie podľa ováňovaných vstupov, líšia sa podľa aktivačnej funkcie. Súčin týchto dvoch hodnôt ukladáme do poľa ako δ_{kj} , kvôli výpočtom zmeny váh v predchádzajúcich vrstvách. A posledná časť:

$$\frac{\partial net_{kj}}{\partial w_i} = out_{(k-1)i} \quad (16)$$

Pri výpočte zmeny váh skrytých vrstiev nastáva zmena pri výpočte člena $\frac{\partial E}{\partial out_{kj}}$ rovnice 14.

$$\frac{\partial E}{\partial out_{kj}} = \sum_{n=1}^m \delta_{(k+1)n} w_n \quad (17)$$

kde m je počet neurónov $k+1$ vrstvy,

n je n -tý neurón $k+1$ vrstvy,

w_n je váha hrany, ktorá vystupuje z j -tého neurónu k -tej vrstvy a vstupuje do n -tého neurónu, $k+1$ vrstvy.

Opäť ukladáme súčin prvý dvoch členov rovnice 14 ako δ_{kj} , pre prípad viacerých skrytých vrstiev.

Po prejdení všetkých hrán a úprave ich váh pokračuje tréning siete ďalšou tréningovou vzorkou. Tento postup sa opakuje, pokiaľ nie je splnená jedna z ukončovacích podmienok:

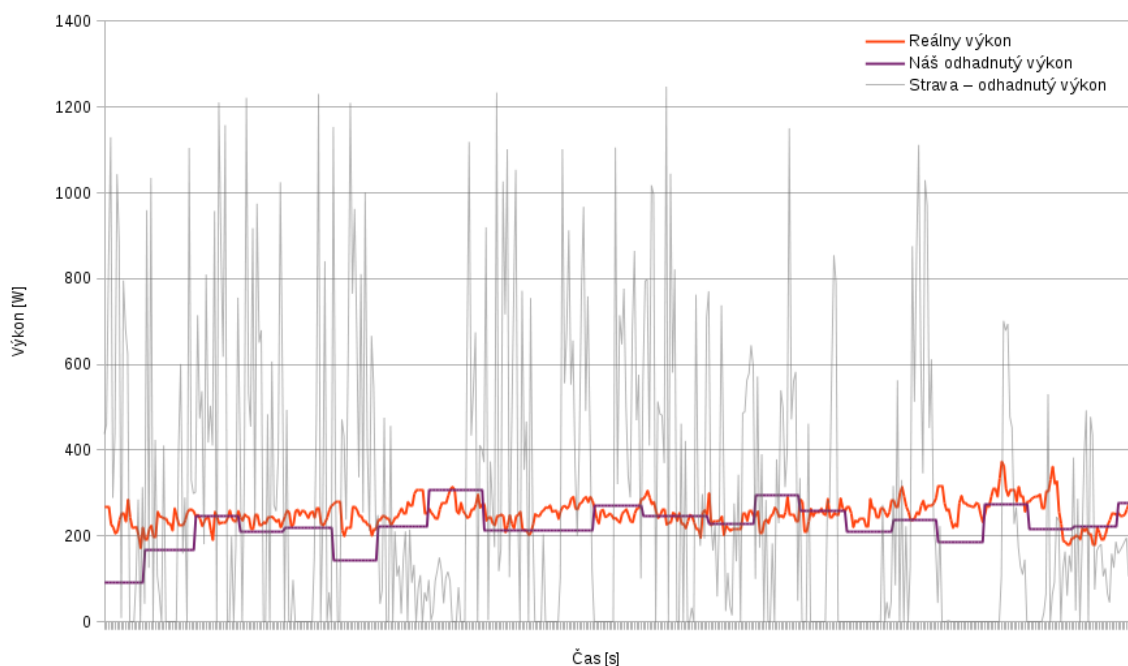
- dosiahnutie predpísanej presnosti celkovej chyby,

- dosiahnutie definovaného počtu epoch,
- nárast celkovej chyby.

3.6 Porovnanie výsledkov

Záver tejto kapitoly venujeme porovnaniu odhadnutých hodnôt s hodnotami, ktoré boli reálne zmerané meračom výkonu. Niektoré merania sme opakovali viackrát a niektoré sme vykonali s dvomi športovými meračmi. Na Garmin Edge 500 sme mali pripojený merač výkonu Stages Rival alebo PowerTap G3, pričom sme dáta nahrali do nášho systému a dáta z druhého, PowerTap Joule GPS, sme nahrali na server Strava, aby sme mohli výsledky porovnať. Z viacerých meraní sme vybrali tie, pri ktorých nedochádzalo k výpadkom signálu, či už GPS alebo merača výkonu. Merania prebiehali v rôznych teplotných podmienkach, od 10 do 350°C a v úsekoch s rôznym priemerným sklonom, od 3 do 12%.

Meranie č.1, úsek č.1 - Kačín (48.1903622, 17.0818403) dĺžka 2.9km, prevýšenie 98m, priemerné stúpanie 3%, čas 8:45, teplota 28°C, nadmorská výška štartu 232 m n.m. Graf zmeraného a odhadnutého výkonu môžeme vidieť na obrázku č. 25 a následne aj hodnoty v tabuľke 2.

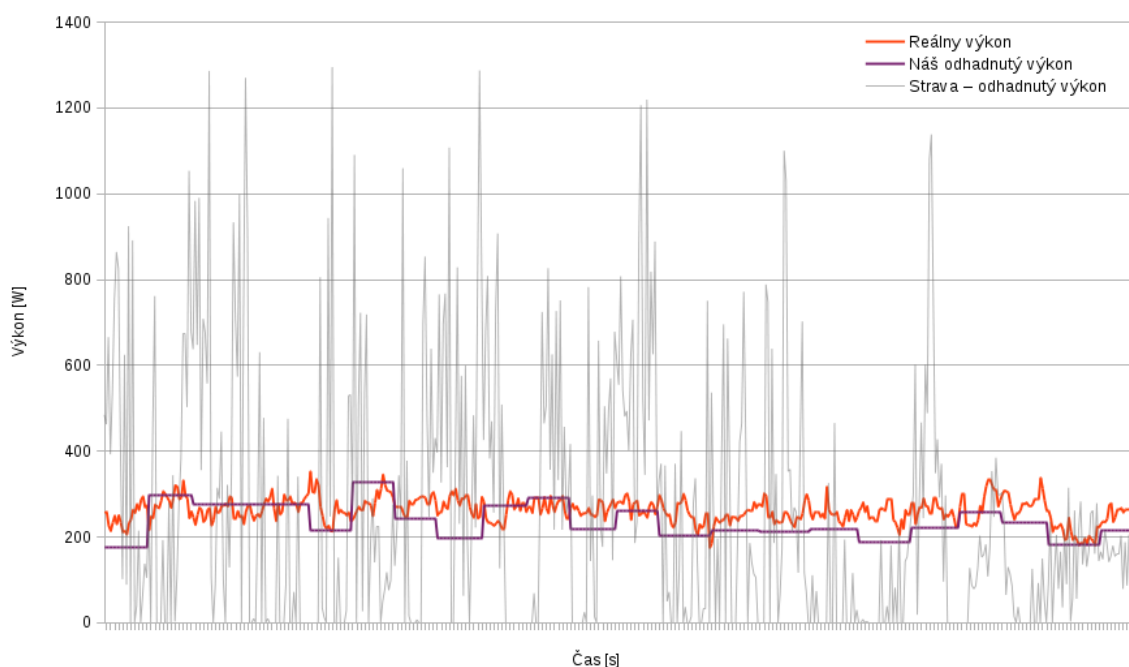


Obr. 25: Porovnanie aproximácie výkonu - úsek č.1 - meranie č.1

Tabuľka 2: Meranie č.1, úsek č.1

	Nameraný výkon [W]	Odhad výkonu algoritmom [W]	Odhad výkonu cez Strava [W]
priemerný výkon	250.49	225.26	286.13
priemerná hodnota rozdielu nameraného a odhadnutého výkonu		40.61	253.86
smerodajná odchýlka rozdielu		35.68	203.75

Meranie č.2, úsek č.1 - Bolo zopakované na úseku **Kačín** ale v nasledujúci deň pri teplote 18°C. Graf zmeraného a odhadnutého výkonu môžeme vidieť na obrázku č. 26 a následne aj hodnoty v tabuľke 3.

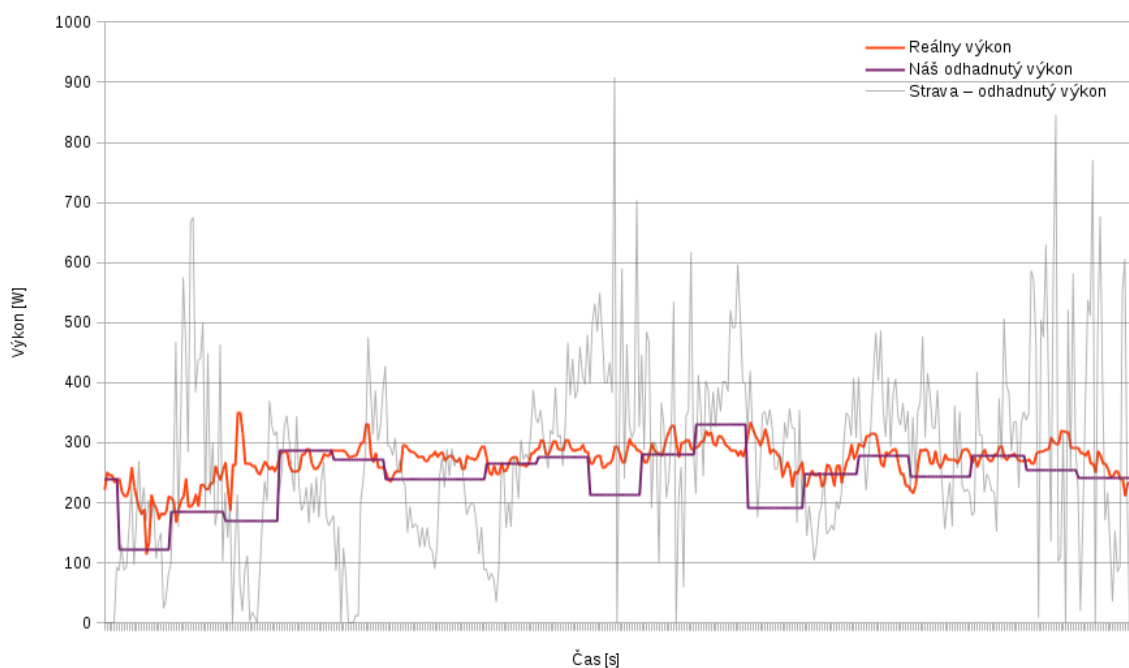


Obr. 26: Porovnanie aproximácie výkonu - úsek č.1 - meranie č.2

Tabuľka 3: Meranie č.2, úsek č.1

	Nameraný výkon [W]	Odhad výkonu algoritmom [W]	Odhad výkonu cez Strava [W]
priemerný výkon	261.43	235.19	255.65
priemerná hodnota rozdielu nameraného a odhadnutého výkonu		39.44	241.71
smerodajná odchýlka rozdielu		26.01	180.01

Meranie č.3, úsek č.2 - Pekná cesta (48.2014473, 17.1273581) dĺžka 2.0km, prevýšenie 167m, priemerné stúpanie 8%, čas 9:00, teplota 22°C, nadmorská výška štartu 222 m n.m.

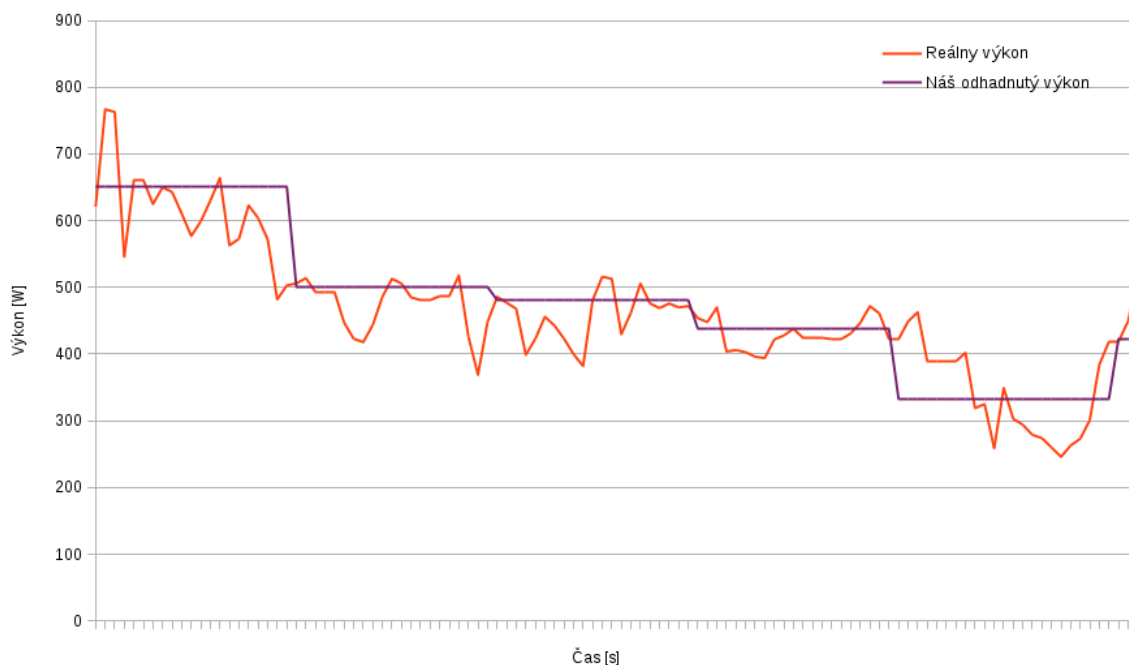


Obr. 27: Porovnanie aproximácie výkonu - úsek č.2 - meranie č.3

Tabuľka 4: Meranie č.3, úsek č.2

	Nameraný výkon [W]	Odhad výkonu algoritmom [W]	Odhad výkonu cez Strava [W]
priemerný výkon	268.59	242.40	275.50
priemerná hodnota rozdielu nameraného a odhadnutého výkonu		34.86	113.30
smerodajná odchýlka rozdielu		31.47	96.56

Meranie č.4, úsek č.3 - Zámok Náchod (50.4227016, 16.1647466), dĺžka 0.5 km, prevýšenie 60 m, priemerné stúpanie 11.0%, čas 1:51, teplota 12°C, nadmorská výška štartu 341 m n.m.

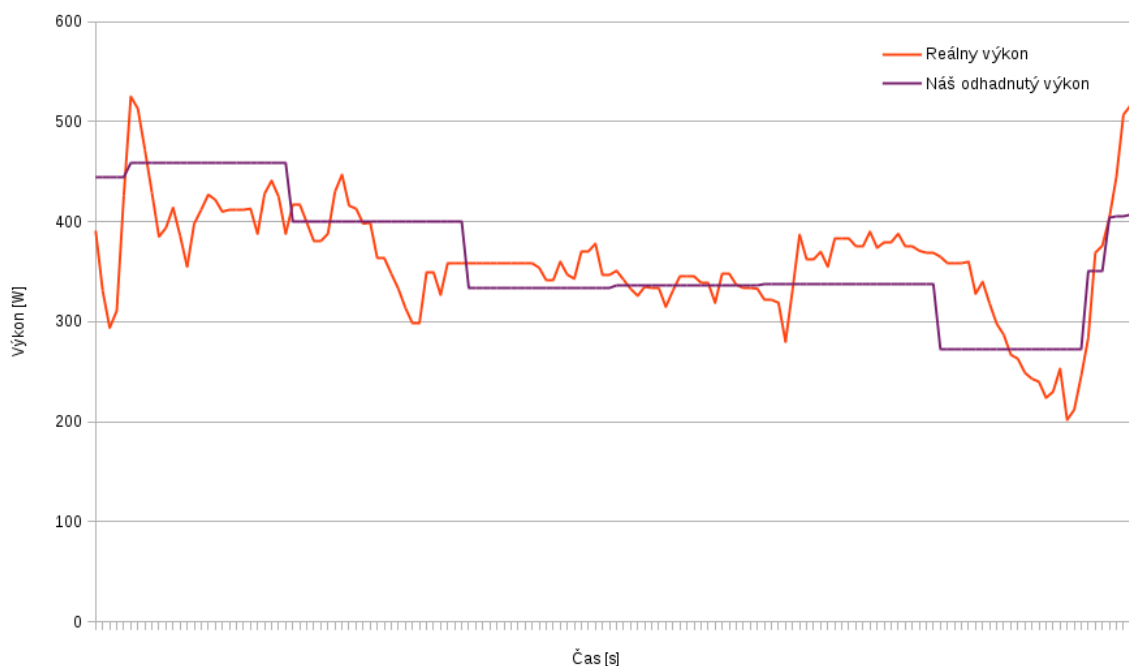


Obr. 28: Porovnanie aproximácie výkonu - úsek č.3 - meranie č.4

Tabuľka 5: Meranie č.4, úsek č.3

	Nameraný výkon [W]	Odhad výkonu algoritmom [W]
priemerný výkon	461.07	476.49
priemerná hodnota rozdielu nameraného a odhadnutého výkonu		41.49
smerodajná odchýlka rozdielu		36.05

Meranie č.5, úsek č.3 - Bolo zopakované na úseku č.3 **Zámok Náchod** v rovnaký deň



Obr. 29: Porovnanie aproximácie výkonu - úsek č.3 - meranie č.5

Tabuľka 6: Meranie č.5, úsek č.3

	Nameraný výkon [W]	Odhad výkonu algoritmom [W]
priemerný výkon	368.93	381.11
priemerná hodnota rozdielu nameraného a odhadnutého výkonu		37.40
smerodajná odchýlka rozdielu		29.48

4 Záver

Výsledkom našej práce je algoritmus, ktorý aproximuje výkon cyklistu v kľúčových stúpaniach na trase tréningu na základe hodnôt z nepriamych meraní. Algoritmus pozostáva z troch krokov: detekcia stúpaní, rozdelenie stúpaní na čiastkové segmenty, odhad výkonu neurónovou sieťou. Algoritmus sme implementovali do webovej aplikácie, ktorá umožňuje nahranie súboru s údajmi o športovej aktivite a jej vyhodnotenie. Aplikácia podporuje viacero formátov vstupných súborov (FIT, GPX, TCX), ktorý bežne používajú zariadenia pracujúce s protokolom ANT+.

Je zrejmé, že presný momentový výkon športovca z pozície, rýchlosti a zmeny nadmorskej výšky nikdy presne určiť nebudeme vedieť. Podstatou nášho algoritmu, je rozdelenie celého stúpania na menšie časti s nemenným sklonom, tzv. segmenty. Pre tieto časti následne vypočítame výkon, akoby ho športovec podával celý čas konštantný. Vytýčenie segmentov v rámci stúpania sme sa snažili navrhnuť tak, aby segmenty boli dlhé natoľko, aby odchýlky v pozícii alebo nadmorskej výške mali minimálny vplyv na výsledok výpočtu. Na druhej strane sa však segmenty snažíme udržať relatívne krátke, aby sme v nich mohli aspoň čiastočne zachytiť náhle, niekoľko sekundové zmeny tempa. V skutočnosti teda nepredpovedáme momentový výkon ale niekoľko sekundový priemerný výkon. Myslíme si, že je to jediná možnosť ako z dostupných dát výkon aproximovať a po porovnaní reálnych a vypočítaných hodnôt na niekoľkých rôznych vzorkách a dosiahnutých odchýlkach sme sa v tomto názore utvrdili.

Aj napriek skutočnosti, že priemerný výkon odhadnutý našim systémom bol horší ako výkon odhadnutý portálom Strava, je dôležité poukázať na presnejšie hodnoty momentálnych výkonov a menšie odchýlky od nameraných hodnôt.

Cieľom tejto práce nebolo konkurovať existujúcim systémom v množstve funkcionality. Cieľom práce bolo umožniť finančne dostupný tréning, ktorý je riadený podľa hodnôt výkonu športovca.

Na záver práce nám už len ostáva skonštatovať, že v tomto výskume je stále priestor na doplnenie. Pri dostatočnom testovaní v rôznych podmienkach, ideálne však vo veternom tuneli by mohol byť vytvorený algoritmus pre aproximáciu výkonu aj v rovinatých častiach tréningu, ktorý nám momentálne chýba. I keď práve stúpania sú kľúčovým miestom, kde sa rozhodujú preteky, nemálo náročných tréningov a pretekov je situovaných v rovinatom prostredí.

Literatúra

- [1] MARSLAND, S.: *Machine Learning: An Algorithmic Perspective*. Chapman and Hall/CRC, 2009. 406s. ISBN 978-1420067187
- [2] ALPAYDIN, E.: *Introduction to Machine Learning*, 2.vydanie. Cambridge, Massachusetts: The MIT Press, 2009. 584s. ISBN 978-0262012430
- [3] MONTAVON, G. - ORR, G. - MÜLLER, K. L.: *Neural Networks: Tricks of the Trade*, 2. vydanie. Berlin: Springer-Verlag, 2012. 769s. ISBN 978-3-642-35288-1
- [4] ORR, G. - SCHRAUDOLPH, N. - CUMMINS, F.: *Preconditioning the Network*. [cit. 27. 10. 2015]. Dostupné na World Wide Web: <<https://www.willamette.edu/~gorr/classes/cs449/precond.html>>
- [5] DÍVALD, L.: *Kontrolovaný tréning*. Bratislava: Laurenc Dívald, 2009. 116s. ISBN 978-80-970358-1-5
- [6] KENNEY, W. L. - WILMORE, J. H. - COSTILL, D. L.: *Physiology of Sport and Exercise*, 5. vydanie. Champaign: Human Kinetics, 2011. 640s. ISBN: 860-1401254188
- [7] REICHL, J. - VŠETIČKA, M.: *Encyklopedie fyziky - Valivý odpor*. [cit. 25. 4. 2015]. Dostupné na World Wide Web: <<http://fyzika.jreichl.com/main.article/view/37-valivy-odpor>>
- [8] HALLIDAY, D. - RESNICK, R. - WALKER, J.: *Fyzika: vysokoškolská učebnice obecné fyziky*. Brno: Vysoké učení technické, 2000. 1278s. ISBN 80-214-1868-0
- [9] WILSON, D. G.: *Bicycling Science*, 3. vydanie. Cambridge: The MIT Press, 2004. 485s. ISBN: 978-0262731546
- [10] CYCLING-WEEKLY: *Power meters: Everything you need to know*. [cit. 15. 4. 2015]. Dostupné na World Wide Web: <<http://www.cyclingweekly.co.uk/news/power-meters-everything-you-need-to-know-35563>>
- [11] CLARK, S. K. - DODGE, R. N.: *A Handbook for the Rolling Resistance of Pneumatic Tires*. Michigan: University of Michigan, 1979. 156s. ASIN: B0006XMSXI

- [12] BIKETECH-REVIEW: *Roller Data*. [cit. 25. 4. 2015]. Dostupné na World Wide Web: <<http://www.biketechreview.com/tires/rolling-resistance/475-roller-data>>
- [13] SCHWALBE: *Rolling Resistance*. [cit. 14. 4. 2015]. Dostupné na World Wide Web: <http://www.schwalbetires.com/tech_info/rolling_resistance>
- [14] ISO 2533:1975 : *Standard Atmosphere* - International Organization for Standardization
- [15] NASA: *The Drag Equation*. [cit. 16. 4. 2015]. Dostupné na World Wide Web: <<https://www.grc.nasa.gov/www/k-12/airplane/drageq.html>>
- [16] CYCLING POWER LAB: *Cycling Aerodynamics & CdA - A Primer*. [cit. 20. 10. 2015]. Dostupné na World Wide Web: <<http://www.cyclingpowerlab.com/CyclingAerodynamics.aspx>>
- [17] SAE J2452:1999 : *Stepwise Coastdown Methodology for Measuring Tire Rolling Resistance* - Society of Automotive Engineers
- [18] THIS IS ANT: *Developer*. [cit. 20. 12. 2014]. Dostupné na World Wide Web: <<http://www.thisisant.com/developer>>
- [19] STRAVA: *About*. [cit. 10. 3. 2015]. Dostupné na World Wide Web: <<https://www.strava.com/about>>
- [20] BIKECALCULATOR: *How to use*. [cit. 10. 3. 2015]. Dostupné na World Wide Web: <<http://bikecalculator.com/how.html>>
- [21] BOOTSTRAP: *Getting started*. [cit. 20. 12. 2014]. Dostupné na World Wide Web: <<http://getbootstrap.com/getting-started/>>
- [22] MBOSTOCK: *D3*. [cit. 20. 12. 2014]. Dostupné na World Wide Web: <<https://github.com/mbostock/d3>>

Obsah CD nosiča

Obsah priloženého CD nosiča je nasledovný:

- v priečinku *dokument* sa nachádza text diplomovej práce vo formáte PDF
- v priečinku *aplikácia* sú zdrojové súbory výslednej funkčnej aplikácie, a tiež skript pre vytvorenie tabuliek v databáze
- v priečinku *dokumentácia* je dokumentácia zdrojového kódu vo formáte HTML obsahujúca popis jednotlivých tried a ich metód