

**ŽILINSKÁ UNIVERZITA V ŽILINE**

---

**AUTOREFERÁT  
DIZERTAČNEJ PRÁCE**

---

**Žilina apríl 2021**

Ing. Tomáš Kello

**Žilinská univerzita v Žiline**  
**Fakulta riadenia a informatiky**

**Ing. Tomáš Kello**

Autoreferát dizertačnej práce

**Operatívne riadenie železničnej dopravy využitím strojového učenia**

na získanie akademického titulu „**philosophiae doctor**“ (v skratke **PhD.**)  
v študijnom programe doktorandského štúdia  
**aplikovaná informatika**

v študijnom odbore:  
**informatika**

**Žilina, apríl 2021**

**Dizertačná práca bola vypracovaná v dennej forme doktorandského štúdia na katedre softvérových technológií Fakulte riadenia a informatiky Žilinskej univerzity v Žiline**

- Predkladateľ:**                    **Ing. Tomáš Kello**  
Katedra softvérových technológií  
Fakulta riadenia a informatiky  
Žilinská univerzita v Žiline
- Školiteľ:**                         **doc. Ing. Emil Kršák, PhD.**  
Katedra softvérových technológií  
Fakulta riadenia a informatiky  
Žilinská univerzita v Žiline
- Oponent:**                         **prof. Ing. Karel Šotek, CSc.**  
Katedra softwarových technológií  
Fakulta elektrotechniky a informatiky  
Univerzita Pardubice
- Oponent:**                         **Ing. Vlastimil Polach, Ph.D.**  
AŽD Praha s.r.o.

**Autoreferát bol rozoslaný dňa: .....**

Obhajoba dizertačnej práce sa koná dňa 18.8.2021 o 10:30 h. pred komisiou pre obhajobu dizertačnej práce schválenou pracovnou skupinou odborovej komisie v študijnom odbore **informatika v študijnom programe aplikovaná informatika**, vymenovanou dekanom Fakulty riadenia a informatiky Žilinskej univerzity v Žiline dňa .....

**prof. Ing. Karol Matiaško, PhD.**  
predseda pracovnej skupiny odborovej komisie  
v študijnom programe **aplikovaná informatika**

Fakulta riadenia a informatiky  
Žilinská univerzita  
Univerzitná 8215/1  
010 26 Žilina

## Úvod

Žijeme v dobe, v ktorej nové dáta vznikajú exponenciálnou rýchlosťou. Zo štatistik vyplýva, že v čase písania tejto práce pribudne každý deň viac ako 2,5 miliardy GB nových dát a toto číslo neustále rastie. Množstvo dát, ktoré máme k dispozícii je enormné, a preto je potrebné vedieť efektívne pracovať s dátami a využiť ich v náš prospech. Jednou z možností, ako využiť dáta, je strojové učenie, ktoré patrí do podoblasti umelej inteligencie. Inteligentné systémy, ktoré používajú algoritmy strojového učenia dokážu predikovať výsledky na základe predchádzajúcich dát, pričom vedia objaviť súvislosti a prepojenia v dátach. Využitie strojového učenia je v dnešnej dobe veľmi široké a môžeme sa s ním stretnúť v oblastiach ako zdravotníctvo, právo, marketing a v rôznych iných odvetviach. Spolu s umelou inteligenciou máme efektívny nástroj na spracovanie veľkého množstva dát, ktoré môžu byť v textovom, zvukovom, alebo vo vizuálnom formáte.

V posledných rokoch sme sa stali svedkami rozvoju mnoho nových frameworkov na vývoj modelov strojového učenia. Keďže je strojové učenie relatívne nový odbor vedy, nedá sa objektívne usúdiť, ktorý jazyk, resp. framework, je na integráciu strojového učenia do aplikácií najlepší. „Boom“ v oblasti sa dá pripísať technológiám s otvoreným zdrojovým kódom a sprístupneniu vývojových nástrojov širokej verejnosti. Spoločnosť Microsoft za posledných pätnásť rokov investovala veľa prostriedkov do snahy umelej inteligencie a výsledkom je multiplatformový framework s otvoreným zdrojovým kódom známy pod názvom ML.NET. Medzi ďalších aktívnych členov v oblasti umelej inteligencie patrí GOOGLE s frameworkom Tensor-Flow a cloudovým riešením Google Cloud Platform. Medzi konkurenciu v oblasti cloudových riešení patrí tiež Microsoft s ich Microsoft Azure službou, Amazon s AWS (Amazon Web Service), IBM a ďalší.

Riešeným problémom je nepresnosť doterajšieho matematického modelu, ktorý počíta teoretickú jazdnú dobu vlaku na známych úsekoch. Matematický model využíva detailné informácie o trati ale nezahŕňa informácie o stave, opotrebovaní trate a ďalšie ťažko definovateľné veličiny. Preto sa pokúsime tento výpočet zlepšiť s použitím historických dát o jazdných dobách na daných úsekoch, tie nepriamo v sebe obsahujú všetky aktuálne problémy, ktoré sa na trati vyskytujú.

Na záver práce využijeme všetky získané znalosti a vytvoríme službu, ktorá bude spustená na webovej adrese. Tým sa použiteľnosť práce zvýši a integrácia v praxi bude jednoduchšia.

## Očakávaný prínos

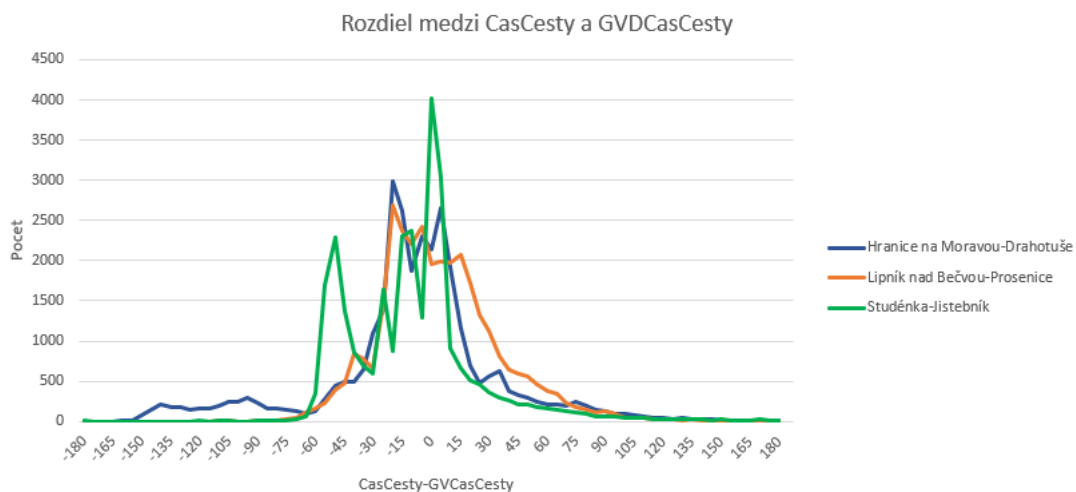
Pre reálne využitie tejto práce v praxi by sme chceli vytvoriť softvérový systém, ktorý bude obsahovať implementovaný model predikcie reálnej jazdnej doby. Tento bude zakomponovaný do aktuálneho dispečerského riadiaceho systému vlakovej dopravy, aby dispečer resp. automatizovaný riadiaci systém dostal presnejšiu informáciu o príchode vlaku do stanice, čo výrazne prispeje k zvýšeniu presnosti pri

určovaní momentu optimálneho postavenia jazdnej cesty pre vlak na vstupe do stanice. To vedie k celkovému zvýšeniu priepustnosti dopravy a optimalizácie využitia koľají a jednotlivých prvkov vlakových ciest.

## Stav riešenej problematiky

Výpočet jazdnej doby vlaku v matematických modeloch je realizovaný pomocou výpočtov z parametrov vlaku a trati. Parametre jazdy vlaku medzi dvoma stanicami sa popisujú z dvoch pohľadov, a to kinematického a dynamického. Konkrétne parametre kinematického pohľadu sú napríklad rozbehnutie, jazda v stabilnej rýchlosti a spomaľovanie. Z dynamického pohľadu je to jazda silou, výbeh a brzdenie.

Tieto modely, ktoré produkujú teoretické výsledky, neberú do úvahy stav lokomotívy, historické záznamy, námraza na koľajniciach a iné faktory, ktoré môžu ovplyvňovať jazdnú dobu vlaku. Z toho dôvodu vznikajú meškania. Ako môžeme vidieť na obrázku 1, skutočná jazdná doba (CasCesty) sa vo veľa prípadoch nezhoduje s teoretickou jazdnou dobou vypočítanou matematickým modelom (GVDCasCesty) a ich rozdelenie hustoty je sústredené v intervale -60s (skrátí jazdnú dobu) a +30s (zvýšil meškание).



Obrázok 1 Graf rozdielu času medzi skutočnou a teoretickou hodnotou.

S cieľom minimalizovať meškание vlakov sa vo svete začali dopravné spoločnosti, ako aj výskumníci, zaujímať o možnosti tvorby nových, presnejších modelov. S veľkým vzostupom strojového učenia vo svete informačných technológií sa táto možnosť stala jasnou voľbou. Vďaka tomu, že dopravné spoločnosti vlastnia veľké súbory dát, presne také, aké pre strojové učenie potrebujeme, má strojové učenie v sektore dopravy veľký potenciál.

## Zber dát

Riadenie železničnej dopravy spracovanie elektronickej dopravnej dokumentácie na území Českej a Slovenskej republiky zabezpečuje informačný

system GTN. Ten nahrádza pôvodné písomné rozpisy príchodov a odchodov vlakov. Zobrazuje operátorovi dopravy aktuálny stav na trati, spolu s plánovaným pohybom jednotlivých vlakových súprav v budúcnosti. Interaktívne si operátor vie zobrazit' náhľad na celkovú dopravu medzi viacerými železničnými uzlami. Alebo si priblížit' a zobrazit' podrobnosti o stanici, zastávke, odbočke, trati, vlakovej súprave, atď... V prípade stanice sa zobrazia podrobné informácie o počte koľají a prepojeniami medzi nimi. Počas prevádzky je dôležité udržiavať celkovú dopravu v pohybe a podľa plánu, prípadne sa prispôbiť aktuálnym meškaniam a zároveň riadiť dopravu detailne v jednotlivých staniach. Určovať presnú vlakovú cestu a miesto zastavenia v stanici.

Keďže človek nie je neomylný, tak ako všade aj tu sa stávajú nehody. Preto informačný systém GTN obsahuje modul ELDODO, ktorý zaznamenáva nie len históriu pohybu vlakov, ale zároveň zaznamenáva všetky pokyny a celé riadenie dopravy. Vďaka tomu sa dajú rekonštruovať a zistiť chyby, ktoré nastali a viedli k mimoriadnostiam. Pre potreby nášho modelu sú potrebné informácie o prejazdoch a presných časoch, kedy vlak prišiel a odišiel zo stanice. Modul ELDODO zapisuje všetky získané informácie do Oracle databázy. Tie sme si následne stiahli, extrahovali pre nás dôležité informácie a transformovali do podoby záznamov vhodných pre ďalšie spracovanie. Každý záznam obsahoval kompletné údaje o jazde vlaku v medzistaničnom úseku a pozostával z týchto údajov:

- FromId – Id stanice z ktorej vlak vychádza
- FromName – Meno stanice z ktorej vlak vychádza
- ToId – Id cieľovej stanice
- ToName – Meno cieľovej stanice
- TrainId – Identifikácia vlaku
- TrainNumber – Číslo vlakovej súpravy
- TrainType – Typ vlakovej súpravy
- Weight – Váha celej vlakovej súpravy [t]
- Length – Dĺžka celej vlakovej súpravy [m]
- CarCount – Počet vagónov vlakovej súpravy
- AxisCount – Počet náprav vlakovej súpravy
- EngineType – Typ lokomotívy
- SectIdx – Číslo úseku na plánovanej trati jazdnej súpravy
- DepRealTime – Skutočný čas odchodu z východzej stanice
- DepILS – Zdroj skutočného odjazdu je z dôveryhodného zdroja
- ArrRealTime – Skutočný čas príchodu do cieľovej stanice
- ArrILS – Zdroj skutočného príjazdu je z dôveryhodného zdroja

- DepPlanTime – Plánovaný čas odchodu vlaku
- ArrPlanTime – Plánovaný čas príchodu vlaku
- LengthSect – Dĺžka železničného úseku [km]
- DriverId – Identifikácia rušňovodiča

V práci sú použité dáta z traťového úseku s celkovým počtom 22 staníc. Ich zoznam s identifikačným číslom a počtom záznamov je zobrazený v tabuľke 1 zoradený podľa počtu výskytov v dátach.

POČET	ID	MENO
499 094	34694000	Studenka
485 560	33722000	Hranice na Mor.
462 341	34652800	Prosenice
461 803	34804500	Suchdol nad Odr.
459 137	34042200	Lipník nad Bečv.
458 108	33472200	Drahotuse
451 328	38314100	Polanka n. O.
445 089	33654500	Jistebník
394 366	34544700	Polom
287 524	34434100	Ostrava-Svinov
179 748	38274700	Odb.Odra
142 706	34662700	Prerov os.n.
116 138	38062600	Odb Skalka
115 387	38042800	Vyh Dluhonice
92 175	35044700	Ostrava-Vitkovice
58 054	33732900	Hranice mesto
46 836	34601500	Sedl.-Bartosovice
24 699	34144600	Nový Jicin mesto
24 360	33064700	Bilovec
22 034	34164400	Odry
17 216	33404500	Fulnek
6 342	34174300	Hermanky
5 474	35004100	Vitkov
4 898	33094400	Svatonovice
2 428	33084500	Budisov n.Budis.
1 651	33722002	Cement Hranice

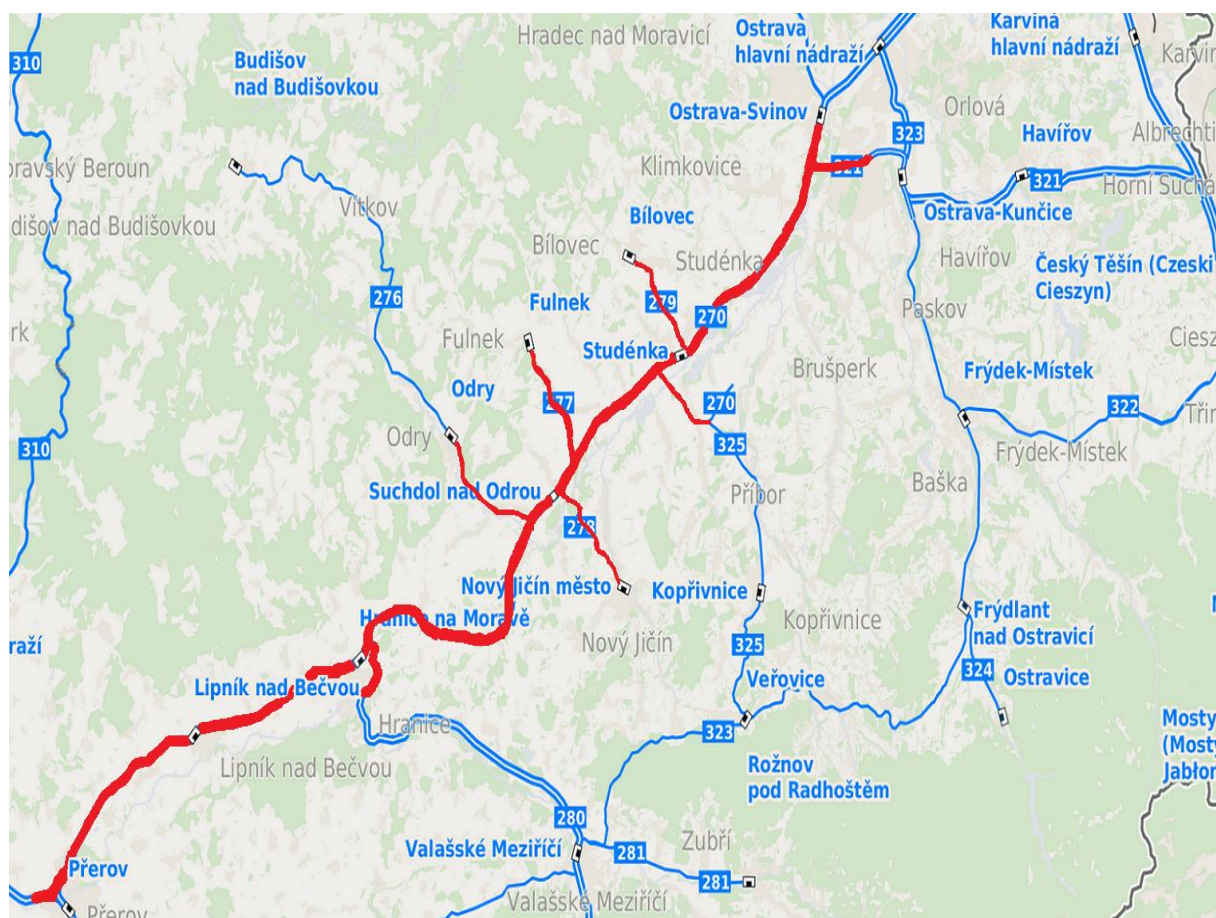
**Tabuľka 1: Zoznam staníc**

Za časové obdobie 5 rokov. Sme nazbierali 5 264 496 záznamov. Dáta sú v rozmedzí od začiatku roka 2016 do konca januára 2021. Počas tohto obdobia premávalo niekoľko druhov vlakov. Ich zoznam, početnosť a opis je zobrazený v tabuľke 2.

POČET	DRUH	POPIS
1 409 521	Ex	Expresný osobný
1 076 633	Os	Osobný
1 038 331	Pn	Prebežný nákladný
832 989	Nex	Nákladný expres
476 455	R	Rýchlik osobný
231 100	Lv	Lokomotivny vlak
59 979	Sluz	Služobný
55 764	Sv	Súpravový vlak
40 583	Sp	Zrýchlený osobný
32 222	Mn	Manipulačný
9 587	Vlec	Vlečkový nákladný
1 332	PMD	Posuny medzi dopravcami

**Tabuľka 2: Druhy vlakov**

Mapu železničných tratí spolu s vyznačenými úsekmi, z ktorých sme stiahli dáta sú vyznačené na obrázku 2. Jedná sa o Olomoucký a Moravsko-sliezsky kraj Českej republiky. V dátach je zahrnutá hlavná trať medzi mestami Přerov a Ostrava ale aj pripojené lokálne trate, zabezpečujúce menej frekventovanú premávku do okolitých miest a dedín.



**Obrázok 2: Mapa železničných tratí**

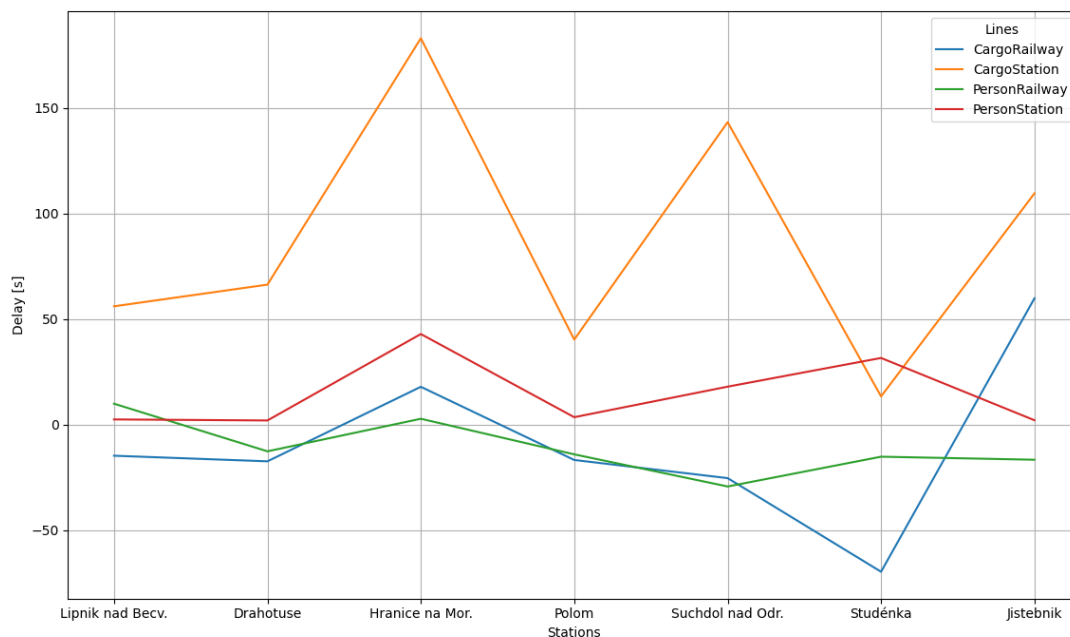


## Rozbor meškanií

V tejto kapitole sa sústreďíme na zdroje meškania a ich závislosti. Niekoľko grafov nám pomôže lepšie porozumieť, kedy tieto meškania vznikajú.

### Porovnanie meškanií nákladných a osobných vlakov

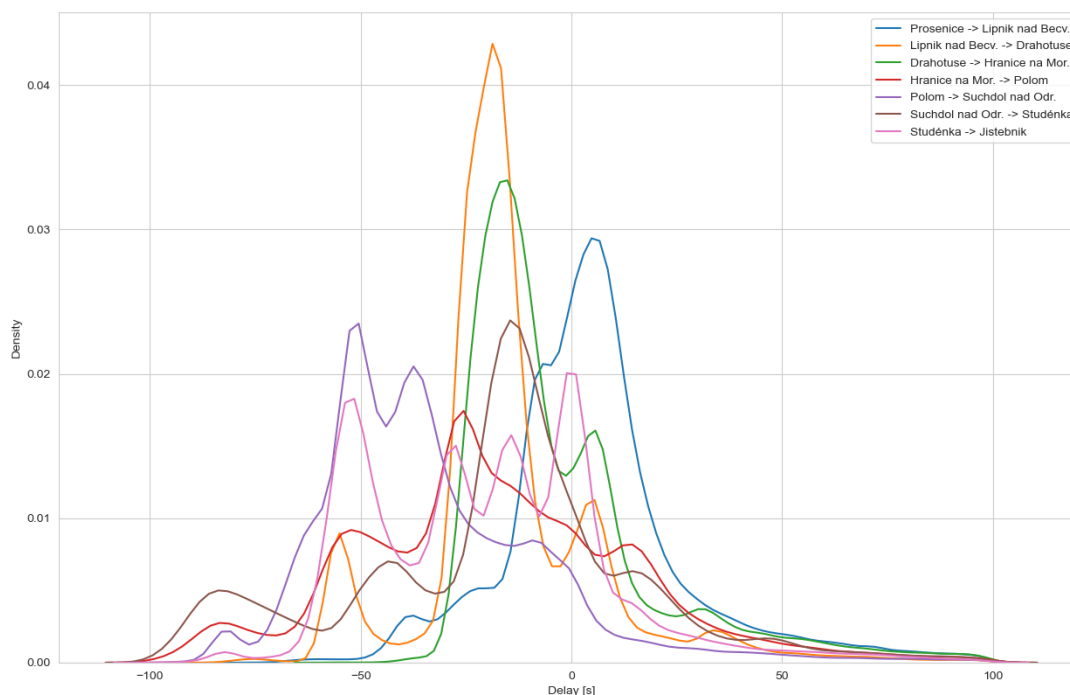
Signifikantný rozdiel v meškanií je medzi nákladnými a osobnými vlakmi a to hlavne v stanici. Tu nákladné vlaky vykazujú výrazne vyššie meškание a teda predĺžený pobyt. Na obrázku 3 môžeme vidieť porovnanie vzniknutých meškanií na tratiach a staniciach osobitne pre nákladné a osobné vlaky. Ako si môžeme všimnúť, väčšina jazdných dôb nevytvára veľké meškania. V plusových hodnotách sa nachádza meškание v stanici pre osobné vlaky a ešte vyššie vzniknuté meškание v staniciach pre nákladné vlaky. Vzniknuté meškania vykazujú závislosti v niektorých staniciach. Napríklad Hranice na Mor. majú zvýšené hodnoty pri všetkých zobrazeniach. Opačne v Studénke v prípade nákladných vlakov je ich meškание nižšie v stanici aj na trati.



**Obrázok 3 Meškание vlakov v medzistaničnom úseku a v staniciach v smere z Lipníka do Jistebníka**

### Porovnanie meškanií na jednotlivých úsekoch trate

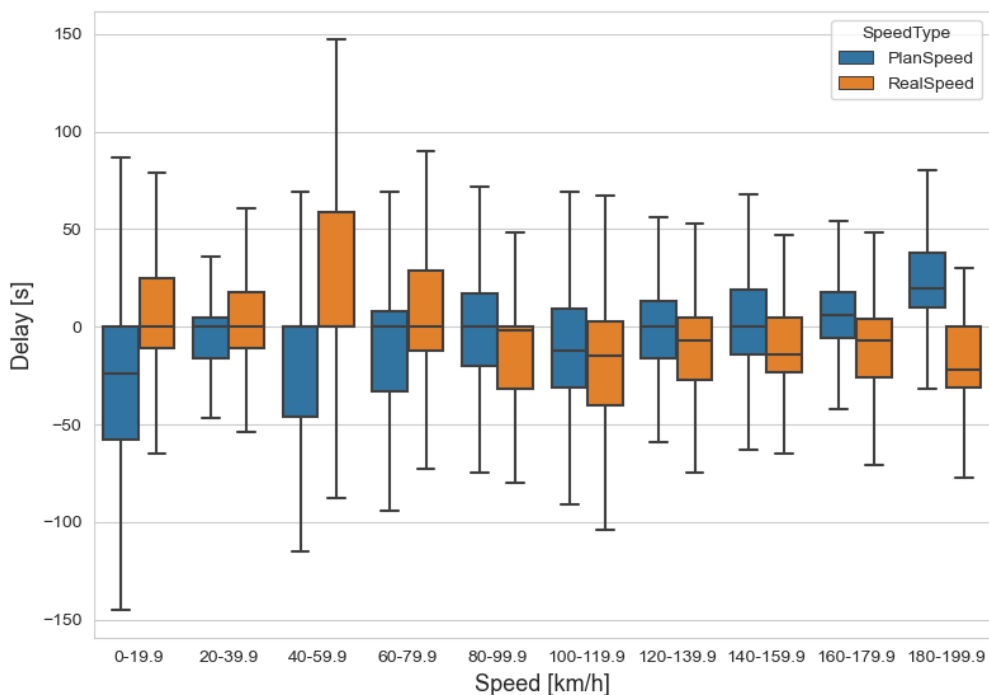
V predchádzajúcom odstavci sme si mohli všimnúť podobné chovanie vlakov v závislosti na stanici. Na obrázku 4 je zobrazené meškание v staniciach detailnejšie pomocou funkcie hustoty. Môžeme teda vidieť jednotlivé vrcholy a teda najčastejšie sa vyskytujúce meškania na danej trati. Napríklad na trati Studénka Jistebník sú tieto vrcholy až 4 a zároveň je rozptyl výrazne vyšší a ťažší na odhad výsledného meškania.



**Obrázok 4 Meškание vlakov na úsekoch**

## Rýchlosť vlaku

Záznamy obsahujú vzdialenosť medzi stanicami a zároveň plánovanú jazdnú dobu. Z toho sa dá vypočítať priemerná rýchlosť vlaku na danom úseku. Samozrejme tento údaj nie je úplne presný, keďže nezapočítavame zrýchľovanie a spomaľovanie vlaku, energetickú efektívnosť, stúpanie, klesanie, strmosť zákrut a tiež čiastočné obmedzenia na trati. Je to údaj akoby celá trať bola jednotná vyjadrená priemernou rýchlosťou medzi bodmi. Preto hodnota, ktorá sa bude nachádzať v stĺpci Speed bude o čosi nižšia, ako skutočná maximálna rýchlosť vlaku na trati.

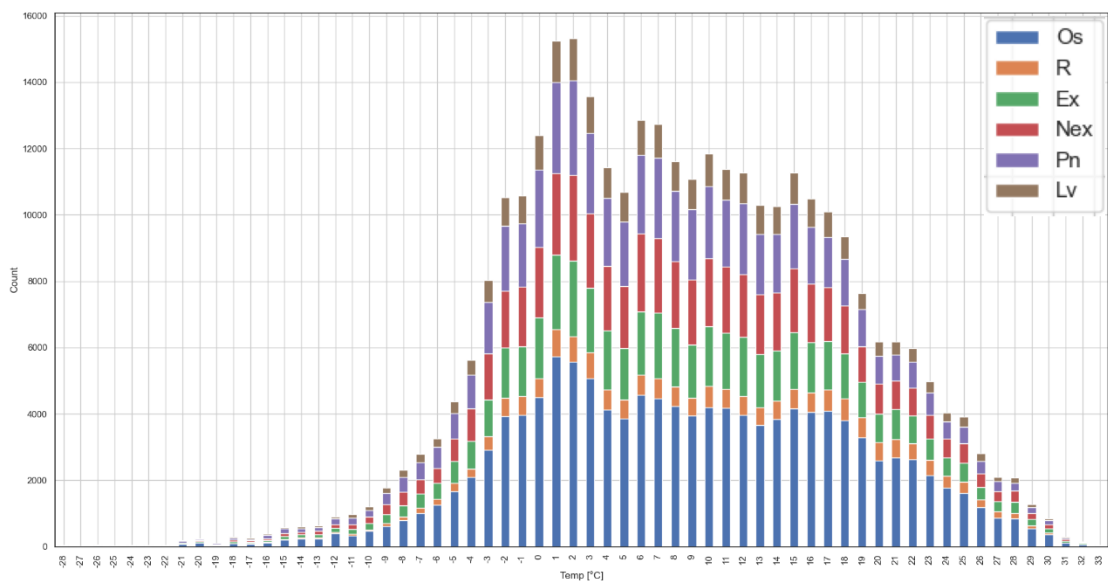


**Obrázok 5 Závislosť meškania na skutočnej/plánovanej rýchlosti**

Graf závislosti je zobrazený na obrázku 5. Skutočná jazdná doba v oranžovom sviečkovom grafe odzrkadľuje výsledok jazdy. Teda ak sa vlak pohybuje v rýchlostiach nad 150km/h, neočakáva sa, že mal problém a teda meškanie by nemalo nastať. Skôr opačne meškanie znižuje. Rovnako to platí v opačnom smere, kde pri priemernej rýchlosti okolo 50km/h sa meškanie výrazne zvyšovalo. Dôležitejšia je pre nás plánovaná jazdná doba zobrazená modrou farbou. Jej trend sa pohybuje nepriamou závislosťou k skutočnej jazdnej dobe. Teda v prípade plánovanej jazdnej doby vo veľkej rýchlosti je veľká pravdepodobnosť, že to vlaková súprava nezvládne a tým vznikne meškanie. Potvrdilo sa tiež, že v prípade pomalších úsekov tratí dokážu vlaky svojou energeticky nevýhodnou jazdou, prudkým brzdením či akceleráciou meškanie znížiť. Táto závislosť zaujímavá a pokúsime sa ju použiť aj vo výslednom modeli.

## Počasie

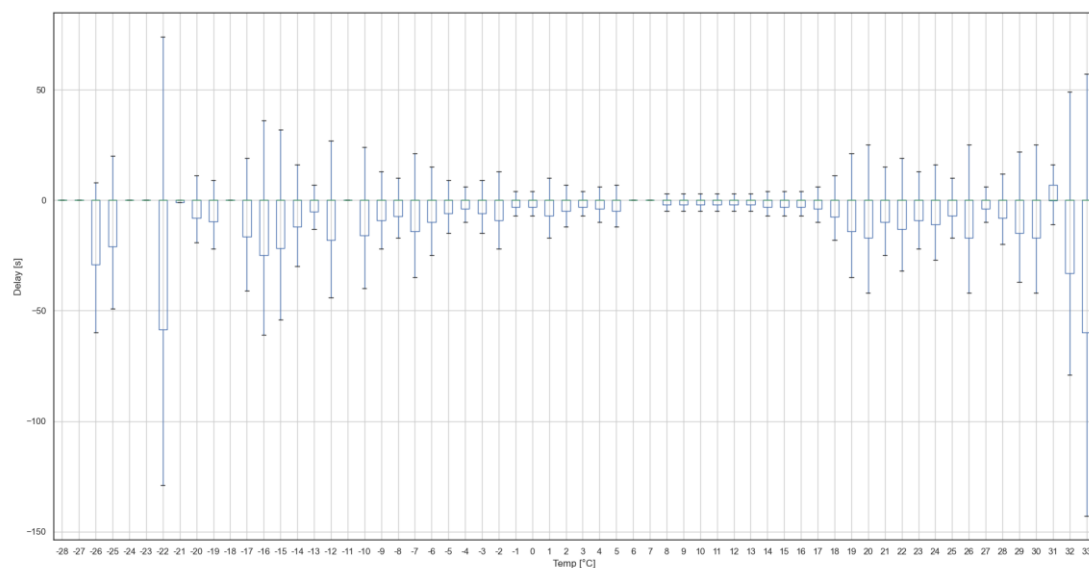
V riadenej oblasti v okolí mesta Čadca, sme sledovali počasie a závislosť teploty na jazdnú dobu. Priemerná teplota ovzdušia za sledované obdobie 4 rokov bola 8,37°C a priemerná teplota jazdy vlakov 8,23°C. Ich rozdelenie podľa teploty ovzdušia a typu vlaku je zobrazené na obrázku 6.



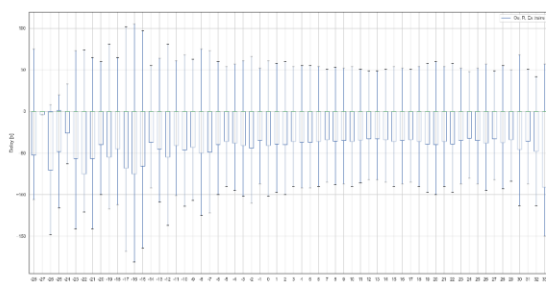
Obrázok 6 Početnosť vlakov idúcich v uvedenej teplote ovzdušia

### Meškanie podľa teploty

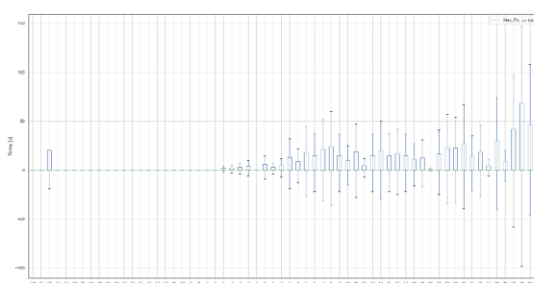
Najočakávanejším vplyvom na jazdnú dobu bolo samotné počasie. Napríklad silné mrazy budú spôsobovať zamrznutie trakčného vedenia a výhybiiek. No pri pohľade na grafy na obrázku 7. Vidíme, že ani všeobecný graf ani rozdelenie vlakov na nákladné a osobné neprispelo k závislosti, ktorú sme očakávali. Horizontálna os zobrazuje meškanie, ktoré vzniklo/skrátilo sa na danom úseku. Teda nezahŕňa meškanie, ktoré bolo naakumulované v predchádzajúcich úsekoch. Zaujímavosťou je hlavne rozptyl jazdných dôb pri nízkych a vysokých teplotách. Dá sa teda predpokladať, že hraničné teploty spôsobujú meškania, ale zároveň v prípade meškania vlaku, nie je obmedzený časom a môže jazdu zrýchliť. Čo spôsobí menšiu energetickú efektívnosť jazdy, ale podarí sa vlaku skrátiť meškanie.



**Rychlíky, Expresné a Osobné vlaky**



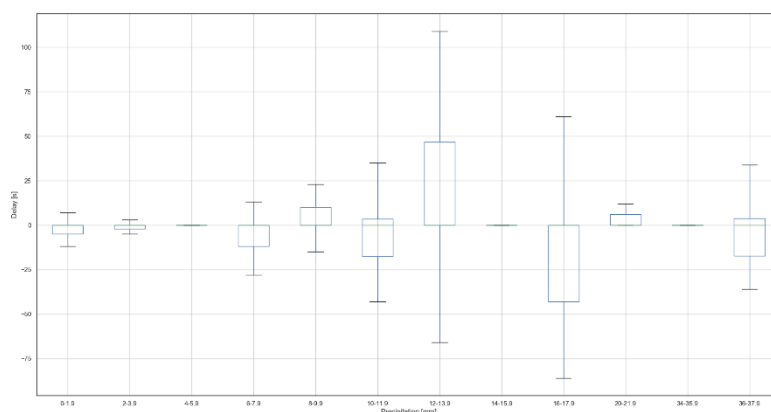
**Nákladné vlaky**



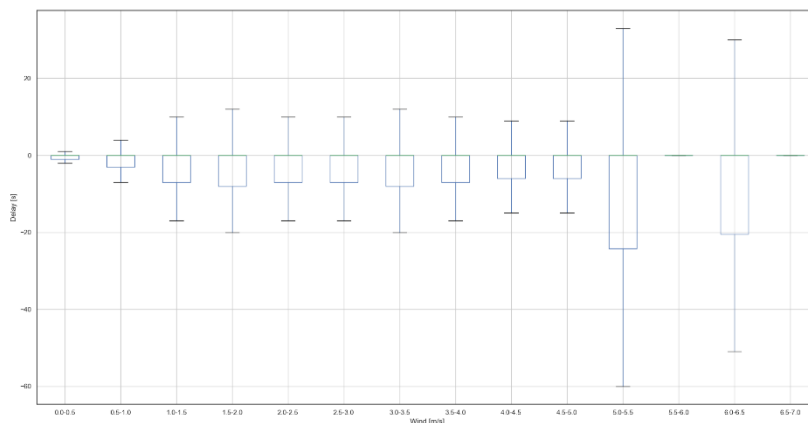
**Obrázok 7 Meškanie vlakov podľa teploty**

## Meškanie podľa ďalších poveternostných podmienok

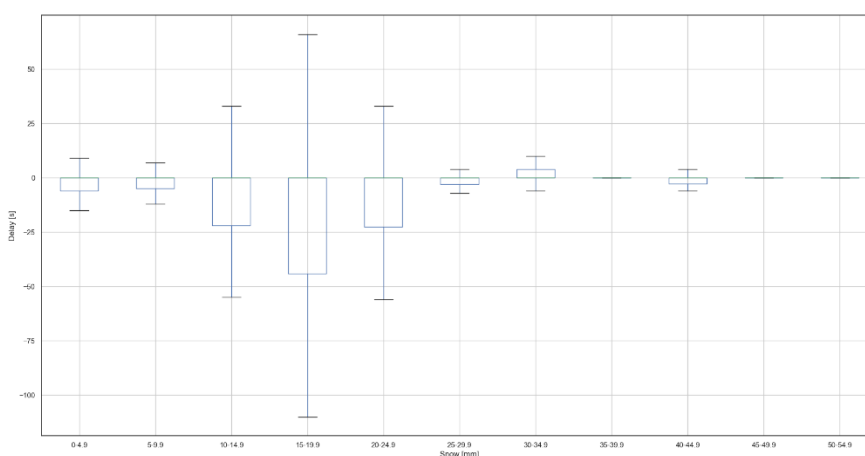
Podobne ako v predchádzajúcej kapitole ohľadom teploty, sme neobjavili ďalšie závislosti v počasi. Na obrázkoch 8 9 a 10 sú zobrazené jednotlivé závislosti, väčšinou hraničné hodnoty spôsobujú meškania, ale zároveň dokážu jazdnú dobu aj skrátiť.



**Obrázok 8 Meškanie podľa zrážok**



Obrázok 9 Meškanie podľa sily vetra



Obrázok 10 Meškanie podľa napadnutého snehu

## Odhad jazdnej doby

### Boosted trees

Implementácia obsahuje definovanie vstupnej funkcie, ktorá je postupne volaná nad menšími časťami dát. Vstupnými parametrami Tensor Flow estimátoru je teda zoznam vstupných stĺpcov, počet opakovaní tréningu nad datasetom a daná vstupná funkcia. Taktiež sa môže obmedziť počet vytvorených rozhodovacích stromov. Prednastavený je limit 100.

```
tf.estimator.BoostedTreesClassifier(feature_columns,
n_batches_per_layer=n_batches).train(train_input_fn, max_steps=100)
```

Po natrénovaní modelu je možné ho uložiť pre neskoršie použitie alebo využiť, kým je načítaný v programe. Funkcia *evaluate* zhodnotí kvalitu modelu, ako mu sa podarilo natrénovať na vstupných dátach. Výstup z funkcie *evaluate* je zobrazený na obrázku 11.

```

accuracy          0.850000
accuracy_baseline 0.715000
auc               0.852667
auc_precision_recall 0.830000
average_loss      0.513541
label/mean        0.320000
loss              0.410000
precision         0.784386
prediction/mean   0.363636
recall           0.750000
global_step       100.000000
dtype: float64

```

Obrázok 11 Boosted trees predikcia

Štatistiky tejto predikcie sú len z trérovacieho datasetu. Teda v prípade pretrénovania, dostaneme veľmi presné výsledky, ktoré by však odzrkadľovali realitu. Preto využijeme druhú časť dát určenú na testovanie modelu. Funkciou *predict* odhadneme jazdnú dobu neznámej vlakovej súpravy. Daný výpočet vykonáme na všetkých testovacích dátach a výsledné hodnoty porovnáme so skutočnou jazdnou dobou.

EPOCHS	Absolut Error [s]	SQR Err [s]
2	22,86	35155,93
10	24,67	18701,35
50	26,50	64780,73
100	26,47	88201,62
Planovaná	42,34	1941127,37

Obrázok 12 Boosted trees chybovosť

Na obrázku 12 môžeme vidieť presnosť predikcie nad všetkými dátami s postupným zvyšovaním parametra epochs – počtu opakovaní a porovnaním s plánovanou jazdnou dobou. Absolútna chyba sa výrazne nezmenila, avšak druhá mocnina, ktorá zvýrazňuje zlé odhady druhou mocninou dosiahla najlepší výsledok pri počte opakovaní 10. Následne sa dá usúdiť pretrénovanosť modelu, keďže sa chyba výrazne zvyšuje na neznámych dátach.

## Konvolučné neurónové siete

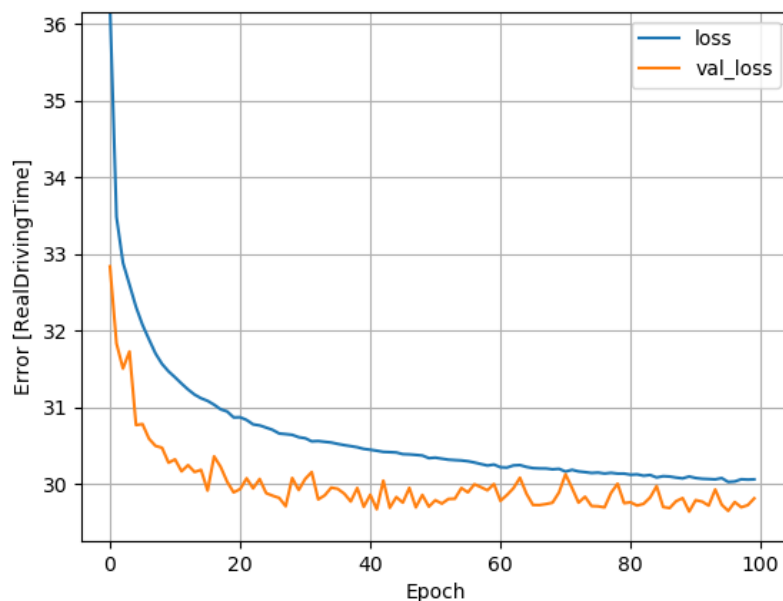
Ako východziu konfiguráciu sme zvolili normalizačnú vrstvu a následne tri výpočtové vrstvy. Ako aktivačnú funkciu využíva ReLU (Rectified Linear Unit activation function) ktorej funkcia vykonáva operáciu  $\max(x, 0)$ . Prvá a druhá vrstva obsahuje 64 uzlov, posledná tretia už len jeden výstupný. Podobne, ako pri algoritme *Boosted trees* sme experimentovali s hodnotou opakovaním trérovania nad vstupnými dátami (Epochs). Trérovanie aj testovanie prebehlo nad všetkými dátami a rozdelenie bolo 85/15% na trérovanie a testovanie.

EPOCHS	Absolut Error [s]	SQR Err [s]
2	25,63	31750,59
10	25,88	25319,74
50	27,62	12493,02
100	26,55	15087,16
Planovaná	42,34	1941127,37

Obrázok 13 Konvolučné neurónové siete presnosť odhadu

Ako môžeme vidieť z obrázku 13, pridávaním počtu opakovaní sa absolútna chyba pomaly zvyšovala. Avšak jej rozdiel nie je tak výrazný, ako zníženie druhej mocniny, ktorá dosiahla najlepšie výsledky pri hodnote EPOCHS = 50. Postupný vývoj tréningovania hodnôt.

- Loss – reprezentuje presnosť tréningovania na známych dátach
- Val\_loss – reprezentuje presnosť validácie a tým všeobecnosť modelu na neznámych dátach



Obrázok 14 Konvolučné n. n. - strata informácie pri tréningovaní a validácii

Na obrázku 14 je zobrazená závislosť straty informácii pri tréningovaní a validácii. V prípade že by sa tieto čiary oddiaľovali, naznačovalo by to pretréningovanie modelu. Taktiež ak by oranžová čiara reprezentujúca validáciu obsahovala príliš veľký šum a teda skoky výrazne nad a pod modrú čiaru. Jednalo by sa o nedostatočné množstvo vstupných dát, kde výsledky sú silno viazané na rozdelenie datasetu.



### Počet vrstiev

Ďalším parametrom, ktorý sme sa snažili upraviť bol počet použitých vrstiev. Ako nám z predchádzajúceho tréningu vyšlo, najlepšie výsledky boli dosiahnuté pri Epoch okolo 50. Tak sme zvolili hodnotu 60. Z výsledkov na obrázku 15 vidíme, že pridávaním vrstiev sa výsledky nezlepšovali a najlepšou voľbou je ostať na 3 vrstvách.

Layers	Absolut Error	SQR Err	Time [s]
2	24.41	28,716.49	1713.7491
3	26.69	19,608.34	1755.8918
4	27.31	24,133.46	1893.4427
5	27.59	69,939.45	2109.8926
6	28.81	64,803.88	2216.8467
7	28.14	64,719.56	2360.9581
8	28.41	59,813.95	2440.3578

Obrázok 15 Konvolučné n. n. zmena počtu vrstiev

### Počet uzlov vo vrstve

Posledným parametrom, ovplyvňujúcim tréning a výsledky modelu je počet uzlov v jednotlivých vrstvách. Je možné každú vrstvu nastaviť inak a tým vznikne obrovské množstvo kombinácií. My sme sa pokúšali o uniformné nastavenie všetkých vrstiev rovnako. Na obrázku 16 sú výsledky pri zmene počtu uzlov. Najlepšie výsledky sme dosiahli, pri počte uzlov 64 na vrstve.

Units	Abs Error	SQR Err	Success	Time [s]
32	26.43	49,405.54	92.50%	1843.5852
48	26.34	40,375.31	92.52%	1832.3817
64	27.31	24,133.46	92.27%	1893.4427
80	28.35	83,214.13	92.00%	1966.413
96	28.94	59,573.44	91.85%	1954.0286
112	28.46	58,507.27	91.97%	2029.7264
128	27.58	61,742.59	92.20%	2223.8114

Obrázok 16 Konvolučné n. n. zmena počtu uzlov vo vrstve

Výsledky predikcii nás dovedli k záveru, že pre náš typ úlohy je najlepšie použitie troch vrstiev s 64 neurónmi na vrstvu a počet tréningov nad vstupnou množinou s náhodne rozdelenými hodnotami s počtom 60. Tieto nastavenia budú ďalej používané v prípade, že použijeme konvolučné neurónové siete k predikcii skutočnej jazdnej doby. Vstupnými dátami boli všetky vlaky po úpravách a ich priemerná jazdná doba za naše skúmané obdobie dosahuje 5 minút 26 sekúnd. Najlepšie dosiahnutá úspešnosť odhadu jazdnej doby bez použitia selekcie dát je na hodnote 91,62%

# Zlepšenie predikcie selekciou dát

## Typ vlaku

Prvým rozdeľovacím kritériom sme zvolili typ vlaku. V databáze sa nachádza 12 unikátnych typov ako môžeme vidieť aj v tabuľke 2, čo by nám však rozdelilo vstupné dáta na moc malé časti a tým by tréning nemuselo dosiahnuť kvalitné výsledky. Spôsobom jazdy je výrazný hlavne medzi nákladnými, osobnými a servisnými vlakmi. Rozhodli sme sa pre rozdelenie do troch skupín.

- Nákladné (Nex, Pn, Mn, Lv)
- Osobné (R, Os, Ex)
- Ostatné (PMD, Vlec, Sp, Sv, Sluz)

TrainType	Count	Plan Abs Err	Plan Sqr Err	Plan Succes	Pred Abs Error	Pred Sqr Err	Time [s]	Pred Succ
Os Ex R	2936651	28.87	9,059.87	90.92%	17.9	784.43	6020	94.17%
Nex Lv Mn Pn	2123540	69.22	1,081,267.88	84.45%	41.35	134,165.43	4267	90.09%
PMD Vlec Sp Sv Sluz	161312	54.86	18,835.60	85.86%	36.09	5,089.92	314	90.22%

**Tabuľka 3 Predikcia podľa typu vlaku**

V tabuľke 3 môžeme vidieť dosiahnuté výsledky predikcie pri rozdelení dát podľa typu vlaku. Osobné vlaky sa nám podarilo predikovať výrazne lepšie. Vzhľadom že osobné vlaky zastrešujú väčšiu časť dát, vážený priemer chyby celkového modelu by dosiahol lepšiu hodnotu po rozdelení datasetu, ako keď sme používali celú vzorku dát a vytvorili jeden model. Pri pohľade na úspešnosť sa každý model zlepšil o 5% k plánovanej jazdnej dobe.

## Časové obdobie

Dátum skrýva v sebe niekoľko možností delenia. Rozhodli sme sa vyskúšať rozdeliť záznamy podľa hodiny odjazdu, dňa v týždni a ročného obdobia. Využívame pri tom SQL funkciu DATEPART, ktorá ako argument používa typ rozdelenia. Konkrétne hour, weekday a month reprezentujúce hodinu, deň v týždni a mesiac.

Time	Count	Plan Abs Err	PlanSQR Err	Plan Succ	Pred Abs Err	Pred Sqr Err	Pred Succ	Time [s]
0-6	1,043,121	41.59	26,417.89	89.13%	26.76	2,428.19	92.72%	2263
6-12	1,325,095	41.56	65,696.25	88.57%	23.62	2,501.84	93.17%	2771
12-18	1,573,082	38.82	28,101.24	89.06%	23.41	1,924.88	93.10%	3377
18-24	1,141,336	61.94	1,673,456.52	84.12%	28.87	13,272.11	91.91%	2430

Season	Count	Plan Abs Err	PlanSQR Err	Plan Succ	Pred Abs Err	Pred Sqr Err	Pred Succ	Time [s]
Winter	1,285,561	39.64	19,337.57	89.04%	27.33	10,831.29	92.18%	2767
Spring	1,258,790	38.62	22,237.56	89.52%	24.43	1,719.35	93.11%	2729
Summer	1,211,688	42.83	53,457.59	88.26%	25.09	1,960.87	92.77%	2606
Autumn	1,465,464	60.01	1,511,103.40	84.61%	49.36	1,485,155.53	86.99%	3200

WeekDay	Count	Plan Abs Err	PlanSQR Err	Plan Succ	Pred Abs Err	Pred Sqr Err	Pred Succ	Time [s]
Mo, Fri	1,512,198.00	47.19	529,700.31	87.32%	25.95	45,164.33	92.61%	2926
Tu, We, Th	2,341,386.00	48.46	615,631.39	87.20%	24.27	2,795.70	93.15%	5005
Sa, Su	1,367,919.00	40.88	60,849.69	88.67%	26.43	45,066.48	92.37%	2618

**Tabuľka 4 Predikcia podľa časového obdobia**

V prvej časti tabuľky 4 sú časovo rozdelené dni podľa hodín do štyroch skupín. Podľa počtu záznamov vidíme, že najmenej vlakov premáva po polnoci do skorého rána a najviac v popoludňajších hodinách. Z pohľadu meškania sa výrazne odlišujú len večerné vlaky, u ktorých výrazne stúpla hlavne štvorcová chyba. To naznačuje vznik menej častých ale väčších meškaní. V prípade ročných období sa opäť ukazuje jeden z dát ako problémovjší. Tým je jeseň, kedy výrazne stúpla absolútna aj štvorcová chyba. Tu si s ňou však model umelej inteligencie nevedel moc poradiť a nezlepšil výsledky tak výrazne ako v prípade rozdelenia podľa časov. Poslednou časťou je rozdelenie týždňa na pracovný začiatok a koniec, stred týždňa a víkend. Rozdelenie hustoty dát je rovnomerné, meškania vznikajú približne rovnako počas celého týždňa. Rozdiel je v dĺžke meškania. Počas týždňa zvyknú meškania dosahovať väčšiu hodnotu na rozdiel od víkendov, kedy sú meškania síce častejšie ale nie tak výrazné. Vidíme to na porovnaní plánovanej absolútnej chyby s plánovanou štvorcovou chybou.

Z pohľadu časového rozdelenia sa nám podarilo najviac zlepšiť predikciu vo večerných hodinách, u ktorých sa znížila absolútna aj štvorcová chyba. Na druhú stranu odhad jazdnej doby v jeseni nepriniesol výrazne zlepšenie ani v jednom z hodnotiacich parametrov.

## Úsek trate

Železničná riadená oblasť obsahuje 23 železničných tratí. Ako reprezentatívnu vzorku sme vybrali 8 náhodných tratí tak, aby sme vedeli porovnať kvalitu predikcie.

Po zoradení hustoty premávky na jednotlivých úsekoch sme vybrali veľmi frekventované ale aj menej využívané úseky. Keďže sme použili selekciu dát, tak sa zmení aj chyba plánovanej jazdnej doby. V tabuľke 5 je náhodný výber úsekov a smeru s počtom záznamov v danom smere a následne absolútna a štvorcová chyba plánovanej jazdnej doby a predikovanej spolu s časom tréovania. Ako môžeme vidieť, úspešnosť sa výrazne zlepšila. Obzvlášť štvorcová chyba je výrazne nižšia, čo

znamená, že modelu sa darí predikovať vzniknuté väčšie meškania. Najmenšia chyba je na trati Prosenice -> Přerov, kde absolútna chyba plánovanej jazdy je len 0,5 sekundy. Reprezentuje priemernú chybu. V takomto prípade je veľmi ťažko konkurovať tak presnému odhadu. Predikčný model zvýšil absolútnu chybu, ale podarilo sa mu výrazne znížiť štvorcovú chybu, ktorá je jeho optimalizačným parametrom pri tréningu.

Railway	Count	Plan Abs Err	PlanSQR Err	Plan Succ	Pred Abs Err	Pred SQR Err	Time [s]	PredSucc
Hranice na Mor. -> Drahotuse	240589	39.76	8,481.25	85.74%	25.78	1,647.93	468	90.26%
Jistebník -> Studenka	231449	39.59	28,608.11	88.87%	20.32	1,432.54	475	93.96%
Prosenice -> Lipník nad Bečv.	218091	45.52	38,388.27	87.58%	25.97	2,279.83	445	92.52%
Studenka -> Jistebník	212733	41.2	14,696.70	87.85%	27.8	1,947.98	410	91.47%
Polom -> Hranice na Mor.	205860	59.83	15,247.58	87.79%	34.03	4,181.67	435	92.67%
Prosenice -> Přerov os.n.	134887	0.5	113.8	99.88%	0.92	1.71	258	99.77%
Přerov os.n. -> Prosenice	123662	86.64	78,748.26	83.86%	63.13	5,805.70	266	87.70%
Bilovec -> Studenka	24211	47.63	6,316.06	93.83%	44.91	4,465.35	49.5379	94.16%

**Tabuľka 5 Odhad jazdnej doby na jednotlivých úsekoch trate**

Experimentálne sme sa pokúsili zamerať na jeden úsek trate s tým, že modelu pridáme informácie aj o iných úsekoch. Očakávali sme zlepšenie predikcie, keďže model má možnosť naučiť sa správanie vlaku aj na iných úsekoch a tým zlepšiť predikciu na našom skúmanom. Informácie z iných úsekov sú len ako druhotná informácie a preto pre zdôraznenie dôležitosti dát zo skúmaného úseku, sme tieto dáta duplikovali. Konkrétne 9 násobne. Tým sme vytvorili pomer 9:1 dát zo skúmaného úseku k dátam z iných úsekoch. Tento experiment nepriniesol pozitívne výsledky a veľmi sa podobal modelu, ktorý pracoval nad všetkými dátami.

## **Zlepšenie odhadu teoretickej jazdnej doby využitím predchádzajúcich úsekov**

Využijeme riadenú oblasť PREOS a jej najdlhšiu hlavnú vetvu medzi Přerovom a Ostravou. Konkrétne sa zameriame na odhad medzistaničného úseku Jistebník Polanka s využitím informácií z 6 úsekov pred ním. Mapa danej oblasti je zobrazená na obrázku 2. Pôvodnú databázu prejazdov sme transformovali do záznamov, ktoré obsahujú skutočné a plánované časy odjazdov a príjazdov s indexom 1-7, o ktorý úsek sa jedná. Ďalej obsahujú výpočty o meškaniach na trati (railway), stanici (station) a oneskorení príjazdu a odjazdu.

K výpočtu očakávaného meškania na medzistaničnom úseku číslo 7 sme použili Lineárnu regresiu. Vstupom boli meškania daného vlaku v predchádzajúcich úsekoch a staniaciach, teda model nemal k dispozícii informácie o vlakovej súprave, typu vlaku, váhy a iných dostupných dát. Výpočet bol teda len z informácií o jeho jazdnej dobe.

Druhý model predikcie sme vytvorili použitím algoritmu neurónových sietí. Nastavenia neurónovej siete a počet vrstiev sme použili rovnaké, ako v predchádzajúcej kapitole pri predikcii. Výsledky sú zobrazené v tabuľke 6. Prekvapivo plánovaný čas má najmenšiu štvorcovú chybu. Avšak jeho priemerná absolútna chyba dosahuje vysokú hodnotu. Čo znamená, že priemerne sa skutočná jazdná doba líši o 40 sekúnd.

algorithm	Absolut err	SQR Err
plan time	40.05	13,961.66
Linear r.	30.15	15,864.84
Neural network	20.87	24,423.53

Tabuľka 6 Predikcia podľa aktuálnej jazdenej doby

## Kombinácia odhadov

V predchádzajúcich kapitolách sme vytvorili model umelej inteligencie, ktorý sa naučil odhadovať jazdnú dobu pomocou historických dát a vedomostí z nich získaných. Jedná sa teda o off-line predikciu, ktorá nepotrebuje prísun nových dát a odhaduje na základe zadaných parametrov vlaku. Parametre odhadu sú označené ako „TrainSet spec“. Tento model by sme chceli použiť v kombinácii s on-line odhadom z kapitole 0, kde ako vstupné informácie potrebujeme aktuálne dosiahnuté jazdné doby vlakovej súpravy. Reprezentuje ho riadok „Track delays“ v tabuľke 7. Prvý riadok reprezentuje plánovanú jazdnú dobu. Ďalšie dva reprezentujú výsledky predchádzajúcich modelov tentokrát na menšej vzorke. Konkrétnej trati s 163780 záznamami.

Druhá polovica tabuľky reprezentuje spojenie dvoch odhadov, kde sme vyskúšali tri matematické funkcie. Podľa výsledkov je predikcia na základe vlastností vlaku výrazne lepšia v oboch parametroch avšak aj online predpoveď z aktuálne dosiahnutých meškaní reprezentuje určité zlepšenie. Preto sa pokúsime získať výhody z oboch a tým vytvoriť ešte lepšie odhady. Tou najjednoduchšou funkciou je priemer dvoch časových odhadov v tabuľke reprezentovaný ako „Mean“. Hodnotu absolútnej chyby si udržalo na veľmi nízkej úrovni a štvorcová chyba sa tiež moc nezhoršila. Avšak oba parametre sú o čosi horšie ako v prípade „TrainSet spec“. Druhou funkciou sme vypočítali vzdialenosť spomenutých dvoch predikcií od plánovanej jazdnej doby a ako výstupnú hodnotu sme použili tú vzdialenejšiu v prípade algoritmu „HighestDiff“ a bližšiu v algoritme „LowestDiff“.

Algorithm	Abs Error	SQR Error	Success
Plan time	34.70	21,811.70	85.61%
Track delays	29.40	21,504.62	90.37%
TrainSet spec	18.46	10,492.40	93.73%
Mean	24.66	14,581.46	91.80%
HighestDiff	16.43	9,707.20	94.38%
LowestDiff	30.04	21,289.83	90.19%
Records count: 163780			

Tabuľka 7 Kombinácia predikcie

V prípade použitia funkcie, ktorá použila odhad viac podobajúci sa plánovanej jazdnej doby sa absolútna chyba ešte zhoršila a štvorcová chyba ostala na hodnote horšej predikcie. Tým je tento spôsob nepoužiteľný a len zhoršuje odhad. No v prípade opačného postupu, kde vyberáme za výstupnú hodnotu tú vzdialenejšiu hodnotu sa nám podarilo dokonca znížiť absolútnu chybu. Spájanie dvoch predikcii sme implementovali v programe Microsoft Excel, kde sme si vložili do tabuľky všetky záznamy, ich plánovanú, skutočnú jazdnú dobu a obe predikcie. Následne sme vypočítali „HighestDiff“ pomocou vzorca na obrázku 17.

```
=IF(ABS([@pred1]-[@plan])>ABS([@pred2]-[@plan]),[@pred1],[@pred2])
```

Obrázok 17 Vzorec kombinujúci predikcie

## Záver

V tejto práci sme sa venovali predikcii jazdnej doby vlakovej súpravy v medzistaničnom úseku, možnostiam zvyšovania úspešnosti odhadu a spätnej predikcii vlastností vlakovej súpravy.

Železničná doprava je riadená na základe grafikonu, ktorý pri aktualizácii využíva matematické výpočty k získaniu jazdnej doby v medzistaničnom úseku. Využíva presnú definíciu trate, predpisy a maximálne povolené rýchlosti. V reálnom svete ovplyvňuje jazdnú dobu aj veľa ďalších faktorov. My sme sa sústredili na dosiahnuté jazdné doby v minulosti, ktoré v sebe zahŕňajú všetky vplyvy na jazdnú dobu. Získali sme záznam o všetkých prejazdoch vlakových súprav cez stanice a ich vlastnosti za obdobie päť rokov. Slovenský hydrometeorologický ústav (SHMU) nám poskytol a rozšíril databázu o záznamy o počasí v meste Čadca. Zozbierané dáta sme transformovali do databázy MS Sql pre lepšiu analýzu a prácu s nimi.

Zlepšenie úspešnosti predikcie závisí na kvalite dát ale aj na vedomosti o zdroji meškaní a spomaľovania vlakovej súpravy. V práci sme analyzovali závislosť skutočnej jazdnej doby na počasí, teplote, zrážkam, sile a smere vetra. Závislosť na type vlaku, jeho dĺžky, váhy, trasy aj času ci obdobiu jazdy. Každé rozdelenie a bližšie špecifikovanie vstupnej množiny zvýšilo presnosť. Avšak selekciou a znížením veľkosti vstupných dát sa znižuje aj spoľahlivosť modelu. Z hľadiska počasia je najväčšie množstvo vlakov vypravovaných pri teplote okolo 5 stupňov, kedy boli zaznamenané aj najpresnejšie jazdy. Hraničné teploty, teda veľké teplo ale aj veľká zima zvyšovali rozptyl jazdných dôb v oboch smeroch. Zaznamenané boli dlhšie jazdné doby a tým zvyšovanie meškania, ale aj skrátenie jazdnej doby. V priebehu dňa je najmenšie množstvo vypravovaných vlakov po polnoci do skorých ranných hodín a meškania vznikajú najviac v poslednej štvrtine dňa medzi 18 a 24 hodinou. V prípade ročných období sme zistili, že najväčšie meškania sú v jesennom období v mesiacoch September, Október a November. A počet vlakov v priebehu roka je uniformne rozložený. Špecifické správanie a nerovnomerný pomer jednotlivých typov vlakov nám priniesol najúspešnejšie predikcie. V prípade osobných vlakov sme sa dostali na úspešnosť 94,17% a celkovo zlepšili predikciu o 5% v porovnaní s plánovanou jazdnou dobou. Medzistaničných úsekoch v našej skúmanej oblasti

medzi Přerov a Ostravou je 22, každý úsek má iné špecifikácie, sklon, polomer zákrut a technické vybavenie. Taktiež záleží v ktorom smere je jazdná doba predikovaná. Vzniká nám 44 úsekov rozdelenia. Pre prehľadnosť sme náhodne vybrali 8 reprezentatívnych vzoriek s hustou ale aj menej hustou dopravou. Skutočná jazdná doba sa líšila od plánovanej v priemere o 10,58% čo sa nám podarilo znížiť na 7,17% a dosiahnuť úspešnosť 92,83%.

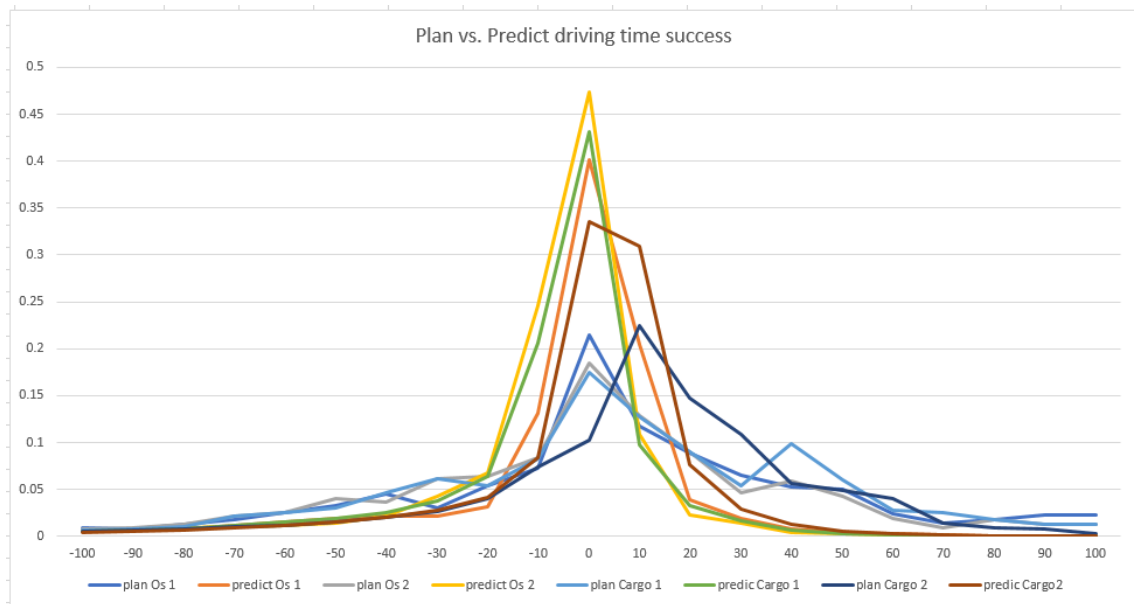
Vytváranie modelov sme programovali v jazyku python, C# a na porovnanie použili implementáciu klauzových riešení. V prípade klauzov boli možnosti implementácie obmedzené, ale ponúkali výhodu vo veľkom výpočtovom výkone počas tréningu bez potreby vlastniť výkonný hardvér. Následne natrénovaný model už nepotrebuje výkonný server a môže bežať na prenajatých serveroch za zlomkovú cenu. V prípade lokálneho vývoja v python a C# sme nepresiahli 6 hodín tréningu. V jazyku Python sme implementovali algoritmus Boosted trees a konvolučné neurónové siete. Oba algoritmy sú upravovateľné niekoľkými parametrami. Nami najlepšie nastavenie dosiahlo úspešnosť 90,9% v prípade Boosted Trees a 92,5% v prípade konvolučných sietí pri použití 4 vrstiev s 64 uzlami na vrstvu. V programovacom jazyku C# sme použili algoritmy FastTree, SDCA a FastForest. Najlepšie z nich dopadol algoritmus FastTree, ktorý dosiahol úspešnosť 91,29%. Porovnanie algoritmov prebiehalo nad všetkými dátami a použitím selekcie sa výsledná predikcia ešte zvýšila.

Spomenuté predikcie boli vypočítané na základe parametrov a vlastností vlakovej súpravy. Jedná sa o off-line predikciu, kde nevstupujú do modelu aktuálne namerané hodnoty. Kombináciou selekcie a algoritmu umelej inteligencie. V prípade selekcie sme využili rozdelenie podľa medzistaničného úseku a na základe typu vlaku do troch skupín osobných, nákladných a ostatných vlakov. A algoritmu konvolučných neurónových sietí sme dosiahli najlepšiu off-line predikciu na hodnote 93,73% priemerne na všetkých dátach. Selekcia rozdelila dáta tak, že každý medzistaničný úsek obsahuje tri modely podľa typu vlaku. Výsledná aplikácia určí trať a typ vlaku, podľa čoho vyberie konkrétne natrénovaný model.

Vedomosti o aktuálnom stave jazdnej súpravy sme využili v ďalšej časti. Aktuálne problémy, opotrebenie a iné faktory môžu dočasne ovplyvňovať schopnosti súpravy dosahovať plánované jazdné doby, ktoré sa nedajú predikovať len z čísla vlaku, či jeho typu. Vybrali sme si jeden medzistaničný úsek a sledovali dosiahnuté jazdné doby a státie v staniaciach predchádzajúcich úsekoch. Nevýhodou tohto riešenia je nevyhnutnosť znalosti jazdných dôb z predchádzajúcich medzistaničných úsekoch čo môže byť problém hlavne na okrajových úsekoch riadenej oblasti. Museli by sme túto informáciu získavať zo susednej oblasti pri príchode jazdnej súpravy do našej riadenej oblasti. Keďže predpovedáme jazdnú dobu na základe aktuálne nameraných hodnôt, voláme to tiež predikcia v reálnom čase. Úspešnosť modelu 90,37% je o niečo nižšia ako v prípade off-line predikcie avšak našim cieľom je ich kombinácia.

Na obrázku 18 sú zobrazené výsledky výsledného programu, ktorý kombinuje on-line a off-line predikciu. Pre prehľadnosť sú vybrané 2 trate a predikcia prebehla nad osobnými a nákladnými vlakmi. Ideálnym grafom by bol Dirakov impulz, ktorý

by nadobúdala hodnotu „1“ v chybe „0“ reprezentujúca 100% záznamov s nulovou chybou a pre zvyšné hodnoty by bola hodnota „0“. My môžeme vidieť, že grafy predikcii sú užšie a vyššie. To znamená že viac záznamov bolo predikovaných s menšou chybou.



Obrázok 18 Predikcia výsledného modelu

K spájaniu sme vyskúšali tri algoritmy, kde sa potvrdil ako najlepší výpočet „Highest diff“ podľa rovnice:

$$=IF(ABS([\text{@pred1}]-[\text{@plan}])>ABS([\text{@pred2}]-[\text{@plan}]],[\text{@pred1}],[\text{@pred2}])$$

Ktorá vyberala predikciu s väčším absolútnym rozdielom od plánovanej jazdnej doby. Týmto spôsobom sme využili off-line predikciu s úspešnosťou 93,73% a on-line predikciu s úspešnosťou 90,37% a dosiahli výslednú hodnotu 94,38%. Čo znamená, že nami predikovanej jazdná doba sa líši len o 5,62% od skutočnej jazdnej doby.

Skutočné jazdné doby sa odlišujú od plánovaných o 85,61%, našu pridanú hodnotu dizertačnej práce a presnejší odhad o 8,77% jazdnej doby by sme chceli využiť aj v praxi. K tomuto účelu sme implementovali internetovú službu v jazyku C#, ktorá očakáva na vstupe parametre jazdnej súpravy. Podporuje použitie off-line predikcie, len na základe vlastností jazdnej súpravy, on-line predikciu a aj ich kombinácie v prípade, že sú vyplnené všetky dostupné informácie o vlakovej súprave.

Záverečnou časťou sme sa venovali spätnej predikcii parametrov vlakovej súpravy. Implementácia sa dá využiť pri kontrole vstupnej trénovanej množiny, prípadne ako prepracovanie záznamu k predikcii k zlepšeniu úspešnosti samotnej predikcie. Začali sme odhadom typu vlaku na základe skutočnej jazdnej doby. Použili sme konvolučné neurónové siete zamerané na kategorickú premennú. Nastavenie s najlepšou úspešnosťou dosiahli tri vrstvi s hodnotami neurónov na jednotlivých



vrstvách 16, 32 a 64. Keďže vstupný parameter bol len jeden a výstupných kategórii 10, dosiahli sme úspešnosť 55,01%. Rozšírením vstupnej hodnoty skutočnej jazdnej doby o plánovanú jazdnú dobu a váhu vlakovkej sústavy sa nám podarilo zvýšiť úspešnosť prvej predikcie na 86,63%. V prípade, že by sme akceptovali dva najpravdepodobnejšie typy vlaku, dostali by sme sa na úspešnosť 94,43%. Najčastejšie vyskytujúcou chybou odhadu bol vlak typu *Priebežný nákladný* označený ako *Nákladný expres* a *Rýchlik* označený ako *Expresný vlak*.

Medzi numerické parametre vlaku patria váha, dĺžka, počet náprav a vagónov. Z poznatkov o predikcii typu vlaku sme vytvorili niekoľko kombinácií parametrov neurónových sietí a aplikovali na všetky kombinácie vstupných parametrov zo skupiny typ vlaku, skutočná a plánovaná jazdná doba. Najlepšiu predikciu sme dosiahli v prípade počtu vagónov na hodnote 84,84%, váhy vlaku na 65,64% a dĺžku vlakovkej súpravy na 72,03%.

Riadenie železničnej dopravy v prípade bezporuchového stavu nie je problém a vlaky by dodržiavali plánované jazdné doby. Aj v nich je už zahrnutá rezerva a tiež možnosť optimalizácie spotreby elektrickej energie šetrným využívaním akcelerácie a brzdenia. To vedia využiť vo svoj prospech v prípade, že jazdnej súprave vzniklo meškanie pomocou menej ekonomickej jazdy. Našou prácou sme sa pokúsili predpovedať vznik komplikácii podľa historicky opakovaných sa udalostí. Podarilo sa nám zvýšiť úspešnosť predikcie jazdnej doby a tým využiť program pri riadení železničnej dopravy upozornením dispečera o predpoklade vzniku meškania pre daný spoj. Ďalším využitím je integrácia do systému ASVC, ktorý zabezpečuje automatické stavanie ciest. Program na výpočet predikcie jazdnej doby pomocou umelej inteligencie je dostupný ako online služba a tým je zjednodušená jeho integrácia a využitie.