České Vysoké Učení Technické v Praze
Fakulta Dopravní
Ústav dopravní telematiky

Posouzení možností využití různých predikčních modelů pro dostupná dopravní data

Diplomová práce

Bc. Filip Hrubý

2023

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta dopravní
děkan
Konviktská 20, 110 00 Praha 1

K620..... Ústav dopravní telematiky

ZADÁNÍ DIPLOMOVÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení studenta (včetně titulů):

Bc. Filip Hrubý

Studijní program (obor/specializace) studenta:

navazující magisterský – IS – Inteligentní dopravní systémy

Název tématu (česky): **Posouzení možností využití různých predikčních modelů pro dostupná dopravní data**

Název tématu (anglicky): Assessment of using different prediction models for available traffic data

Zásady pro vypracování

Při zpracování diplomové práce se řiďte následujícími pokyny:

- Základní charakteristiky dopravního proudu a způsob jejich měření - aktuální trendy a vývoj;
- Explorativní datová analýza dostupných dat;
- Popis predikčních modelů a jejich použití pro predikci dopravních parametrů na dostupných datech;
- Porovnání přesnosti použitých predikčních modelů;
- Zhodnocení výsledků a vyvození obecných možnosti využití predikčních modelů

Rozsah grafických prací: dle požadavků vedoucích diplomové práce

Rozsah průvodní zprávy: minimálně 55 stran textu (včetně obrázků, grafů a tabulek, které jsou součástí průvodní zprávy)

Seznam odborné literatury: R. J. Hyndman a G. Athanasopoulos,Forecasting: Principles and practice. OTexts,2021,

Vedoucí diplomové práce: **Ing. Jiří Růžička, Ph.D.**
Ing. Kristýna Navrátilová

Datum zadání diplomové práce: **1. června 2022**
(datum prvního zadání této práce, které musí být nejpozději 10 měsíců před datem prvního předpokládaného odevzdání této práce vyplývajícího ze standardní doby studia)

Datum odevzdání diplomové práce: **15. května 2023**
a) datum prvního předpokládaného odevzdání práce vyplývající ze standardní doby studia a z doporučeného časového plánu studia
b) v případě odkladu odevzdání práce následující datum odevzdání práce vyplývající z doporučeného časového plánu studia

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE Fakulta dopravní L. S.

Ing. Zuzana Bělinová, Ph.D.
vedoucí
Ústavu dopravní telematiky

prof. Ing. Ondřej Přibyl, Ph.D.
děkan fakulty

Potvrzuji převzetí zadání diplomové práce.

Bc. Filip Hrubý
jméno a podpis studenta

V Praze dne 1. června 2022

Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci, zpracovanou na závěr studia na ČVUT v Praze Fakultě dopravní. Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací. Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze, květen 2023

.....
Bc. Filip Hrubý

Abstrakt

Cílem předkládané diplomové práce je posoudit možnosti využití různých predikčních modelů pro analýzu dat, získaných ze dvou typů dopravních detektorů. Pro analýzu byla použita data, získaná z oblasti v okolí Dobřichovic. Kvalita dat z detektorů byla ověřena explorativní datovou analýzou. Pro predikci dopravní intenzity bylo vybráno 8 různých modelů (ARIMA, Prophet, Naivní sezónní model, K-nejbližší sousedé, Random Forest, XGBoost, Hybridní model s XGBoost a Neuronové sítě). Na základě křížové validace a stanovením přesnosti použitých predikčních modelů byly jako nejlepší vybrány modely XGBoost a Prophet.

Klíčová slova: dopravní proud, dopravní detektory, FCD, predikční modely, strojové učení, explorativní datová analýza, ARIMA, Prophet, K-NN, Random Forest, XGboost, NNAR

Abstract

The aim of this thesis is to assess the potential of using different prediction models for the data analysis obtained from two types of traffic detectors. The data obtained from the area around Dobřichovice were used for the analysis. The quality of the detector data was verified by exploratory data analysis. Eight different models (ARIMA, Prophet, Naive Seasonal Model, K-Nearest Neighbors, Random Forest, XGBoost, Hybrid Model with XGBoost and Neural Networks) were selected to predict traffic volume. Based on cross-validation and by determining the accuracy of the used prediction models, XGBoost and Prophet were selected as the best models.

Keywords: traffic flow, traffic detector, FCD, prediction models, machine learning, explorative data analysis, ARIMA, Prophet, K-NN, Random Forest, XGboost, NNAR

Poděkování

Rád bych poděkoval všem, kteří mi poskytli podporu při vypracování této diplomové práce. Zejména děkuji vedoucím mé práce Ing. Jiřímu Růžičkovi, Ph.D. a Ing. Kristýně Navrátilové za čas, který mi věnovali, konzultace, cenné rady a kritické připomínky, které mně pomohly vylepšit moji práci. Děkuji svým kolegům z Oddělení datové vědy a statistiky Institutu klinické a experimentální medicíny MUDr. Michalovi Kahlemu, Ph. D., Ing. Istvánovi Módosovi, Ph. D. za rady ohledně datové a statistické analýzy. V neposlední řadě děkuji rodičům za podporu při studiu.

Seznam tabulek

Seznam obrázků

Seznam zkratek

API Application Programming Interface. 11

AR Autoregrese. 32

ARIMA Autoregressive integrated moving average. xiv, 26, 27, 31–35, 46, 53, 58–60, 63

CFCD Cellular Floating Car Data. 7, 8, 63

CITS Cooperative Inteligent Transportation System. 8, 63

CSV Comma separated values. 11

EDA Explorativní datová analýza. ix, 1, 10–12, 29, 63

FCD Floating Car Data. ix, xiii, 1, 7–9, 18–22, 34, 35, 38, 42, 47, 50, 54, 59–63

GFCD GPS Floating Car Data. 7, 8, 63

GPS Global Positioning System. xi, 8, 63

ITS Inteligent Transportation System. 8, 24

KNN K-Nearest Neighbors. ix, 26, 27, 36–38, 59, 63

MA Moving average. 32

MAE Mean absolute error. 51, 52, 55–57, 59

MAPE Mean absolute percentage error. 51, 52, 55–57, 59, 60

MASE Mean absolute scaled error. 27, 28, 52, 55–57, 59, 61

MLE Maximum likelihood estimation. 33

NNAR Neural network autoregression. 28, 49, 50, 60, 64

OBU On-Board Unit. 8

RMSE Root mean square error. 52, 55–57

RSS Residual Sum of Squares. 40

RSU Road Side Unit. 8

SMAPE Symmetric mean absolute percentage error. 52, 55–57

SQL Structured Query Language. 11

TMC Traffic Message Channel. 18

XGBoost eXtreme Gradinet Boosting. 26, 28, 42, 44–47, 60, 61, 63, 64

XML Extensible Markup Language. 11

ŘSD Ředitelství silnic a dálnic. 18, 63

Obsah

Kapitola 1

Úvod

Jedním z hlavních problémů současné dopravy jsou kongesce, jež vznikají překročením kapacity dané komunikace, nebo náhodně vzniklou mimořádnou událostí jako nehoda, uzavírka, oprava komunikace atd. Dopravní situaci lze vyhodnocovat na základě dat získaných ze strategických dopravních detektorů. Jedná se buď o detektory stacionární, které jsou umístěny na prvcích dopravní infrastruktury jako kamerové systémy, radary nebo indukční smyčky. Nebo lze také využít informace získané vozidly, která se účastní dopravního proudu. Pro řidiče jsou klíčové nejenom informace o aktuální dopravní situaci, ale také odhady intenzity provozu s časovým předstihem před plánováním cesty. Takové predikční modely by mohly pomoci předcházet kongescím a umožnit včas regulaci dopravní situace. To povede ke zlepšení plynulosti a bezpečnosti dopravy.

Cílem předkládané diplomové práce je posoudit možnosti využití různých predikčních modelů pro dostupná dopravní data. V rámci teoretického úvodu budou popsány základní charakteristiky dopravního proudu a způsoby jejich měření. Strategické dopravní detektory budou rozděleny do kategorií a bude popsán princip jejich fungování. Dále bude popsána technologie Floating Car Data (FCD), tato technologie slouží k monitorování dopravního proudu a je založena na sběru polohových dat z vozidel.

Pro účely této diplomové práce bylo získáno několik souborů dopravních dat z různých lokací a z různých druhů dopravních detektorů. Tyto datové soubory budou popsány pomocí explorativní datové analýzy (EDA). Nejprve bude popsán postup procesu EDA, kterým se práce bude řídit. Poté budou popsány jednotlivé charakteristiky dopravního proudu, které lze z datových souborů vyčíst.

Následně budou popsány predikční modely, které lze použít pro predikci stavu dopravního proudu. Každý použitý model bude nejprve obecně popsán. Poté budou vytvořeny predikční modely pro získané datové soubory.

Budou stanoveny metriky popisující přesnosti použitých predikčních modelů a bude popsán způsob jejich získávání. Jednotlivé predikční modely budou navzájem porovnány.

Také bude porovnána přesnost jednotlivých modelů podle vstupních dopravních dat. Nakonec budou vyvozeny zhodnoceny výsledky a vyvozeny závěry o obecných možnostech využití predikčních modelů.

Kapitola 2

Základní charakteristiky dopravního proudu a způsob jejich měření

V této kapitole jsou popsány dostupné datové zdroje, která jsou použity pro vypracování této diplomové práce. Také jé vysvětlen princip dopravních detektorů, na kterých datové zdroje staví. Nejprve jsou popsány jednotlivé dopravní veličiny, které je možné pomocí dopravních detektorů měřit.

2.1 Popis dopravních veličin

Stav dopravního proudu je definován pomocí několika základních dopravních veličin: dopravní intenzity, rychlosti a hustoty. Z těchto veličin lze pomocí fundamentálních grafů dopravy definovat stupně dopravy.

2.1.1 Dopravní intenzita

Dopravní intenzita je určena počtem vozidel projíždějících danou sekcí pozemní komunikace za jednotku času. Dopravní intenzita se značí písmenem q a má jednotky voz/hod.

$$q = \frac{N}{T} \tag{2.1}$$

Kde N je počet vozidel projíždějících daným úsekem za danou časovou periodu T. Jedná se o nepřímo měřenou veličinu, většinou se dopočítává na základě obsazenosti dopravních detektorů, nebo manuálním počítáním vozidel při dopravních průzkumech.

2.1.2 Rychlost dopravního proudu

Rychlost dopravního proudu je určena aritmetickým průměrem bodových rychlostí jednotlivých vozidel.

$$u = \frac{1}{N} \sum_{i=1}^{n} v_i \tag{2.2}$$

Kde u je rychlost dopravního proudu, N je počet vozidel a v_i jsou jednotlivé bodové rychlosti vozidel.

2.1.3 Hustota dopravního proudu

Hustota dopravního proudu je definována jako aktuální počet vozidel na daném úseku pozemní komunikace. Jednotkou dopravní hustoty je počet vozidel za jednotku vzdálenosti (nejčastěji *voz/km*). Dopravní hustota se značí písmenem k a lze ji dopočítat pomocí rychlosti u a intenzity q pomocí vzorce:

$$k = \frac{q}{u} \tag{2.3}$$

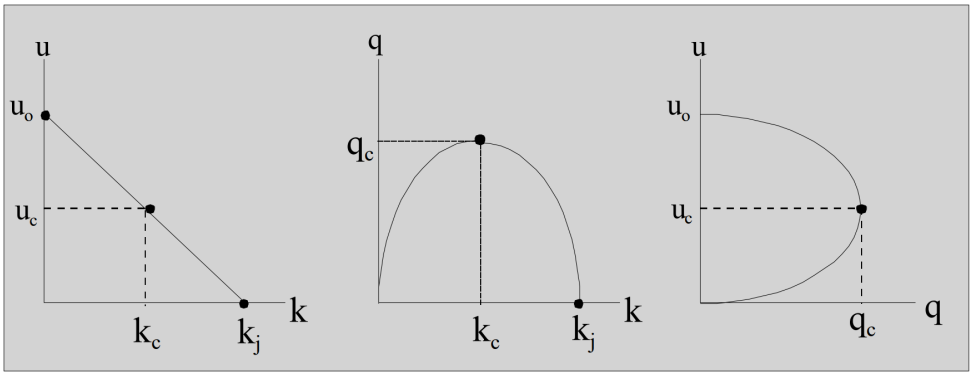
2.1.4 Stupeň dopravy

Pomocí intenzity, rychlosti a hustoty dopravního proudu lze určit dopravní stupeň zkoumané silniční komunikace. Stupeň dopravy popisuje dopravní situaci na zvolené škále (většinou A-F, nebo 1-5).

1. Stupeň dopravy - volný dopravní proud
2. Stupeň dopravy - částečně ovlivněný provoz
3. Stupeň dopravy - plynulý provoz
4. Stupeň dopravy - tvorba kongescí, plynulý provoz je narušen
5. Stupeň dopravy - dopravní kolaps

2.2 Fundamentální grafy dopravy

Vztah mezi dopravní intenzitou, hustotou dopravy a rychlostí dopravního proudu (matematický vzorec 2.3 v kapitole 2.1.3) lze podle teorie dopravního proudu zobrazit pomocí fundamentálních grafů dopravy (Obr. 2.1).



Obrázek 2.1: Fundamentální grafy dopravního proudu, Převzato z Traffic Detector Handbook[1]

První graf na Obrázku 2.1 znázorňuje vztah mezi rychlostí a hustotou dopravního proudu. Na ose u se vyskytují body u_0 a u_c, u_0 značí rychlost volného průjezdu (volný průjezd nastává, když je hustota i intenzita rovná nule) a u_c značí kritickou rychlost (pod kterou začíná tvorba kongescí). Na ose u se vyskytují body k_c a k_j, kde k_c značí kritickou hustotu a k_j značí maximální možnou hustotu při dopravní kongesci. Z grafu je vidět že s klesající rychlostí roste hustota dopravního proudu, tento vztah nemusí být přímo lineární (v reálném provozu často nebývá).

Druhý graf na Obrázku 2.1 znázorňuje vztah mezi intenzitou a hustotou dopravního proudu. Na ose q se vyskytuje pod q_c značící kritickou intenzitu (maximální možnou). Z grafu je vidět že pro nízkou intenzitu může být nízká nebo vysoká hustota. Při nízké hustotě i intenzitě je dopravní proud volný až částečně ovlivněný (dopravní stupně 1 a 2), při nízké intenzitě a vysoké hustotě nastává tvorba kongescí až dopravní kolaps (dopravní stupně 4 a 5) a v kritickém bodě hustoty a intenzity je dopravní proud plynulý (dopravní stupeň 3).

Třetí graf na Obrázku 2.1 znázorňuje vztah mezi rychlostí a intenzitou. Při vysoké rychlosti a nízké intenzitě je dopravní proud volný, kdežto při nízké intenzitě a nízké rychlosti se začínají vytvářet dopravní kongesce.

2.3 Popis strategických dopravních detektorů

Strategické dopravní detektory jsou zařízení určená k monitorovaní a sběru dopravních dat. Tyto detektory jsou umístěny na infrastruktuře silničních komunikací. Existuje několik druhů strategických dopravních detektorů, jednotlivé druhy se liší použitou technologií. Strategické dopravní detektory lze rozdělit na dvě skupiny podle jejich vztahu k vozovce na intruzivní a neintruzivní.

2.3.1 Intruzivní dopravní detektory

Intruzivní dopravní detektory zasahují svojí konstrukcí do vozovky. Zde jsou stručně popsány indukční smyčky a pneumatické detektory.

Indukční smyčka

Indukční smyčka je nejčastěji používaný druh detektoru pro sběr aktuálních dopravních dat. Detektor se skládá z drátěné smyčky, která je zakopaná ve vozovce. Smyčka má obvykle obdélníkový nebo kruhový tvar a je tvořena několika závity drátu. K indukční smyčce je připojeno elektronické zařízení, které do smyčky dodává střídavý proud, čímž indukuje její závity. Při průjezdu vozidla přes indukční smyčku se změní její indukčnost. Změna indukčnosti je poté zaznamenána jako průjezd vozidla. [2][3]

Pneumatické detektory

Pneumatický dopravní detektor pro detekci vozidel používá princip změny tlaku v gumové hadici umístěné příčně k silniční komunikaci. Podle změny tlaku v trubce určí měřící zařízení na základě definovaných kritérií a algoritmů průjezd vozidla. Jedná se o nízkonákladový a lehce přenosný způsob detekce vozidel. Značnou limitací této technologie je malý počet možných sledovaných pruhů a nepřesnosti způsobené změnami teploty a nákladními vozidly s větším počtem náprav. [4] [3] [5]

2.3.2 Neintruzivní dopravní detektory

Instalace neintruzivních detektorů nevyžaduje zásah do vozovky. Patří mezi ně mikrovlnný radar, kamerové systémy a infračervené nebo akustické detektory. Níže je stručně uveden jejich princip.

Mikrovlnný radar

Mikrovlnný radar je schopen detekovat přítomnost, rychlost a zjednodušenou klasifikaci projíždějících vozidel. Radar vysílá mikrovlnné signály s konstantní frekvencí směrem k silniční komunikaci. Při průjezdu vozidla sledovanou zónou radar spočítá rychlost vozidla pomocí Dopplerova jevu. Dopplerův jev udává, že změna frekvence mezi odeslaným a přijatým signálem je proporční rychlosti vozidla. [6][3]

Kamerové systémy

Videodetekční kamerové systémy fungují díky spolupráci videokamer umístěných nad silniční komunikací a softwarovou aplikací. Videokamery snímají obraz, který je přenesen

do softwarové aplikace. Ta pomocí virtuální detekce vozidel identifikuje účastníky dopravního provozu. Identifikace vozidel probíhá pomocí strojové detekce poznávacích značek, nebo pomocí virtuálních indukčních smyček. [3]

Infračervené detektory

Infračervené dopravní detektory se dělí dle typu jejich senzoru na pasivní a aktivní. Aktivní infračervené detektory jsou umístěny nad vozovkou ozařují detekční oblast na vozovce infračerveným zářením o nízkém výkonu pomocí laserových diod. Paprsek se od vozovky odráží a detektor dopočítá dobu mezi vysláním a přijetím infračerveného paprsku. Při průjezdu vozidla se tato doba změní, čímž je vozidlo detekováno. Běžně jeden detektor vysílá na detekční oblast několik paprsků, aby mohl detekovat rychlost a délku vozidla.

Pasivní infračervené detektory nevysílají žádnou energii pouze jí detekují. Senzor pasivního infračerveného detektoru snímá teplotu vozovky, při průjezdu vozidla snímanou oblastí se naměřená teplota změní, čímž se detekuje vozidlo. [7] [3]

Akustické detektory

Akustické detektory jsou méně používaným druhem dopravních detektoru. Akustické detektory jsou většinou umístěny po stranách silniční komunikace v místech s rychle pohybujícími se vozidly. Tyto detektory analyzují zvukové vlny (zvuk motoru a pneumatik), vysílané projíždějícími vozidly. Vozidla jsou poté detekována na základě zvukové analýzy. [8][4]

2.4 Popis dat z plovoucích vozidel

Data z plovoucích vozidel, neboli Floating Car Data (FCD) označuje data vytvářená vozidly, osazenými jednotkami, které přenáší informace o poloze a času datového centra. [9] Datům se říká plovoucí (floating), protože jednotlivými senzory jsou samotná vozidla "plovoucí"v dopravním proudu. Tím se liší od dat ze strategických senzorů umístěných na infrastruktuře pozemní komunikace.

2.4.1 Popis technologie FCD

Technologie FCD lze rozdělit na tři kategorie, GFCD, CFCD a kooperativní FCD. Tyto tři kategorie se dělí podle technologie použité ke sběru dat, princip plovoucích vozidel je zachován.

GPS Floating Car Data

Technologie GFCD využívá ke sběru dat OBU jednotky ve vozidlech, které obsahují GPS modul. GPS modul pravidelně zaznamenává polohu a čas zaznamenání polohy. Tyto informace jsou následně přeneseny do datového centra pomocí mobilních sítí. GFCD je možné získávat z vozidel integrovaných do managementu vozového parku, jako jsou například vozidla taxi služby, vozidla doručovacích společností nebo nákladní vozy. [3]

Za GFCD lze považovat i data získávána z chytrých mobilních telefonů pomocí zabudovaného GPS modulu. Tento způsob používají společnosti jako například Google, Apple a Waze, k určení dopravního stavu na silnicích pro jejich mapové navigační aplikace. [10][11]

Cellular Floating Car Data

Technologie Cellular Floating Car Data (CFCD) využívá princip triangulace základnových stanic mobilních sítí k určení polohy mobilního telefonu. Tato metoda funguje dobře v místech s větším počtem základnových stanic (antén) jako například ve městech. Značnou nevýhodou této metody je nepřesnost dat, způsobená problémem s oddělením účastníků dopravního provozu a osob vyskytujících se v okolí pozemní komunikace. [3]

Kooperativní FCD

Ke sběru dat z plovoucích vozidel lze využít i kooperativní inteligentní systémy (CITS). CITS fungují na systému výměny dat mezi vozidly navzájem (tzv. V2V komunikace) a mezi vozidly a infrastrukturou (tzv. V2I komunikace). Komunikace je založená na standardu IEEE 802.11p.

Sběr dat probíhá na základě identifikace vozidla při průjezdu v blízkosti jednotky umístěné na infrastruktuře (RSU). Pomocí průjezdů u RSU jednotky lze zjistit dopravní intenzitu v daném bodě. Párováním průjezdů vozidel u různých RSU jednotek lze vytvořit dráhu vozidla a dopočítat dobu jízdy.

Tento způsob sběru FCD je značně limitován momentálně slabým rozšířením CITS. S průběžným růstem vozidel schopných zapojení do kooperativních systémů a výstavbou potřebné infrastruktury bude tato technologie pro sběr dopravních dat lépe využitelná. [12][13][14]

2.4.2 Vývoj trendů měření dopravního proudu

S nástupem 5G sítí, CITS a dalších technologií ITS a Smart City se více prosazuje instalace neintruzivních dopravních detektorů. Konkrétně kamerových systémů, které díky velkým pokrokům v oblasti strojového čtení obrazu, jsou nyní schopny přesně detekovat vozidla

a číst poznávací značky. Do budoucna je možné očekávat kamerové systémy s LIDAR (Light Detection And Ranging) pro přesnou identifikaci vozidel, cyklistů i chodců [15].

Další velice aktuální technologií je FCD. Díky velkému zastoupení řidičů, používající chytré mobilní telefony k navigaci, je dostupné velké množství dat o aktuální dopravní situaci.

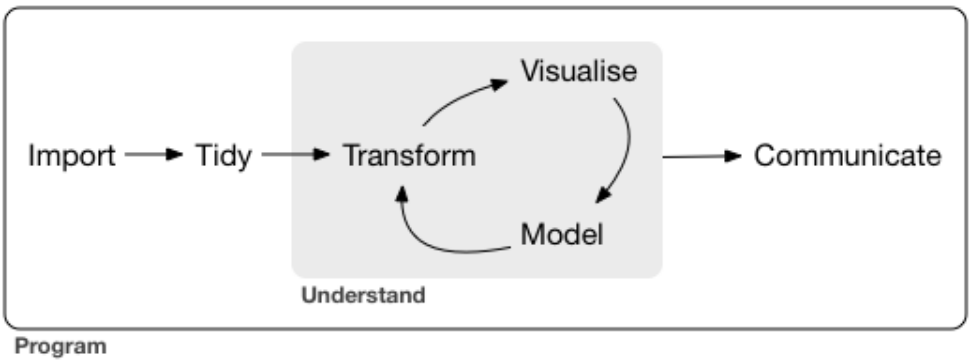
Kapitola 3

Explorativní datová analýza

V této kapitole je vysvětlen proces explorativní datové analýzy, pomocí něhož jsou popsány jednotlivé datové soubory.

3.1 Proces explorativní datové analýzy

Explorativní datová analýza je proces v datové vědě, který slouží k analyzování, vizualizaci a sumarizaci datových sad. [16] [17] Celý obecný proces explorativní datové analýzy je rozdělen do několika částí, které na sebe vzájemně navazují. Jako vzor procesu EDA bude použita definice z knihy "R for data science"[18]. Ke zpracování EDA bude použit programovací jazyk R spolu s open source vědecko-technickým publikačním systémem Quarto [19] a potřebnými knihovnami pro jazyk R.



Obrázek 3.1: Proces EDA Převzato z R for Data Science [18]

Na Obrázku 3.1 je proces EDA graficky znázorněn. Nejprve jsou data importována do prostředí ve kterém je analýza prováděna. Následně jsou data vyčištěna do definované podoby. Poté začíná koloběh porozumění, kdy jsou pomocí transformace, vizualizace a modelování data analyzována. V poslední části je správná komunikace výsledku z již provedeného procesu EDA.

3.1.1 Import dat

Prvním krokem EDA je import datového souboru. Nejčastější formáty jsou formáty tabulových procesorů jako XLSX nebo jiné strukturované datové formáty jako CSV a XML. Další možností importu dat je spojení s databázovým serverem a získání dat pomocí SQL dotazu nebo přes webové API.

3.1.2 Čištění dat

Po importu dat do zvoleného prostředí, ve kterém je EDA zpracovávána, je potřeba data vyčistit. To znamená dostat data do takové podoby, kde každý sloupec je proměnná, každý řádek je pozorování, a v každé buňce je právě jedna hodnota. Tato čistá podoba dat je analogická formátu 3. normálové formy v relačních databázových systémech. Zároveň tato část zahrnuje zpracování chybějících hodnot, překlepů v datech a odlehlých hodnot. [20]

3.1.3 Transformace dat

Transformace dat zahrnuje výběr pozorování, která vás zajímají (například chceme zobrazit data pouze z určité oblasti nebo z určitého časového úseku), vytvoření nových atributů, které jsou funkcemi stávajících atributů (například výpočet rychlosti ze vzdálenosti a času), a výpočet sady souhrnných statistik. [18]

3.1.4 Vizualizace dat

Vizualizace dat je velice důležitou komponentou EDA. Zahrnuje vytváření grafických výstupů, které pomáhají odhalit vzory, vztahy a anomálie, které mohou být obtížně rozpoznatelné pouze z dat v číselné podobě. [18] [21]

3.1.5 Modelování dat

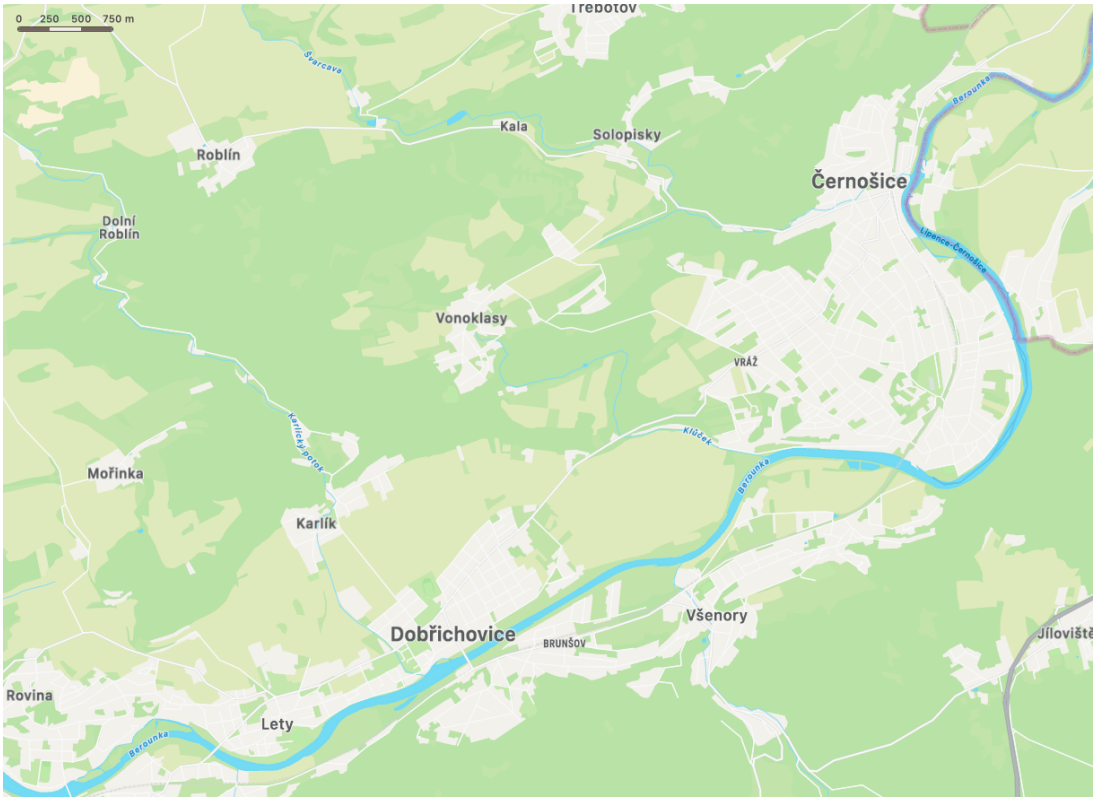
Modelování dat je dalším důležitým komponentem EDA. Zahrnuje vytváření statistických modelů. Modelování může pomoci identifikovat vzory a vztahy v datech, předpovídat budoucí chování a testovat hypotézy o základních procesech, které jsou daty popsány. [18] Mezi často používané modely patří: lineární regrese, klasifikace (logistická regrese, k-nejbližší sousedé, support vector machines), klastrování (k-means, hierarchické klastrování), analýza časových řad a mnoho dalších.

3.1.6 Komunikace výsledků

Posledním krokem procesu EDA je komunikace výsledků. Komunikace zahrnuje prezentaci výsledků v jasné a pochopitelné podobě. Nejedná se pouze o ukázku výsledných vizualizací a tabulek, je nutné veškeré výstupy popsat a sumarizovat nejdůležitější poznatky. [18] [16]

3.2 Popis zkoumané oblasti

Zkoumané datové sady pochází z oblasti města Dobřichovic. Tato lokalita byla vybrána z důvodu dobré dostupnosti dopravních dat, díky projektům na kterých Ústav dopravní telematiky z Fakulty dopravní spolupracoval.



Obrázek 3.2: Mapa Dobřichovic. Převzato z Apple map

Na Obrázku 3.2 je vidět město Dobřichovice nacházející se přibližně 22 kilometrů jihozápadně od Prahy. Jedná se o malé město s přibližně čtyřmi tisíci obyvateli. Pro účely předkládané diplomové práce je ještě důležitá obec Lety, která leží jihozápadně od Dobřichovic a město Černošice ležící přibližně tři kilometry severovýchodně od Dobřichovic. [22] Lety, Dobřichovice a Černošice jsou spojeny silnicí s názvem Pražská, jedná se o komunikaci druhé silniční třídy.

3.3 Analýza dat ze strategických detektorů v Dobřichovicích

Nejprve budou analyzována data ze strategických detektorů v Dobřichovicích. Data poskytlo město Dobřichovice. Jedná se o datovou sadu přibližně 2.1 milionu záznamů ze čtyř mikrovlnných radarů za období 19. 6. 2020 - 26. 1. 2021. Detektory se nachází na následujících místech:

- Ulice Pražská, z Dobřichovic do Černošic
- Ulice Pražská, z Černošic do Dobřichovic
- Ulice Pražská, z Dobřichovic do Let
- Ulice Pražská, z Let do Dobřichovic

Tabulka 3.1 popisuje jednotlivé atributy získaného datového souboru. Atributy, které jsou relevantní pro analýzu dopravního stavu byly následně analyzovány.

Název atributu	Popis atributu
id_detektor	Identifikační číslo detektoru s rozlišením směru a pruhu
datum_cas	Datum a čas měření
intenzita	Počet vozidel ze časový interval
intenzita_n	Normovaná intenzita
obsazenost	Obsazenost detektoru v procentech
rychlost	Rychlost vozidla v km/h
stav	Technický stav detektoru (funkční, nefunkční)
typ_vozidla	Typ vozidla (osobní, nákladní, motocykl)
trvani100	Časový rozestup mezi daty v historii rychlosti ve stovkách milisekund
rychlost_historie	Historie rychlostí k dané detekci vozidla
typ_vozidla10	Přesnější detekce typu vozidla

Tabulka 3.1: Popis atributů datové sady strategický detektorů z Dobřichovic

Datový soubor se skládá z 95% osobních a 5% nákladních vozidel. Nejdůležitější atributy pro zjištění informací o dopravním proudu jsou: *id_detektor* pro identifikaci detektoru, *datum_cas* pro zavedení měření do časové řady, *rychlost* pro výpočet rychlosti dopravního proudu, *typ_vozidla* pro rozlišení skladby dopravního proudu. Sloupce *intenzita*, *intenzita_n* a *obsazenost* nebudou potřeba, protože každé měření obsahuje právě jedno vozidlo, takže se intenzita dá dopočítat jako počet řádků.

3.3.1 Data ze všech detektorů

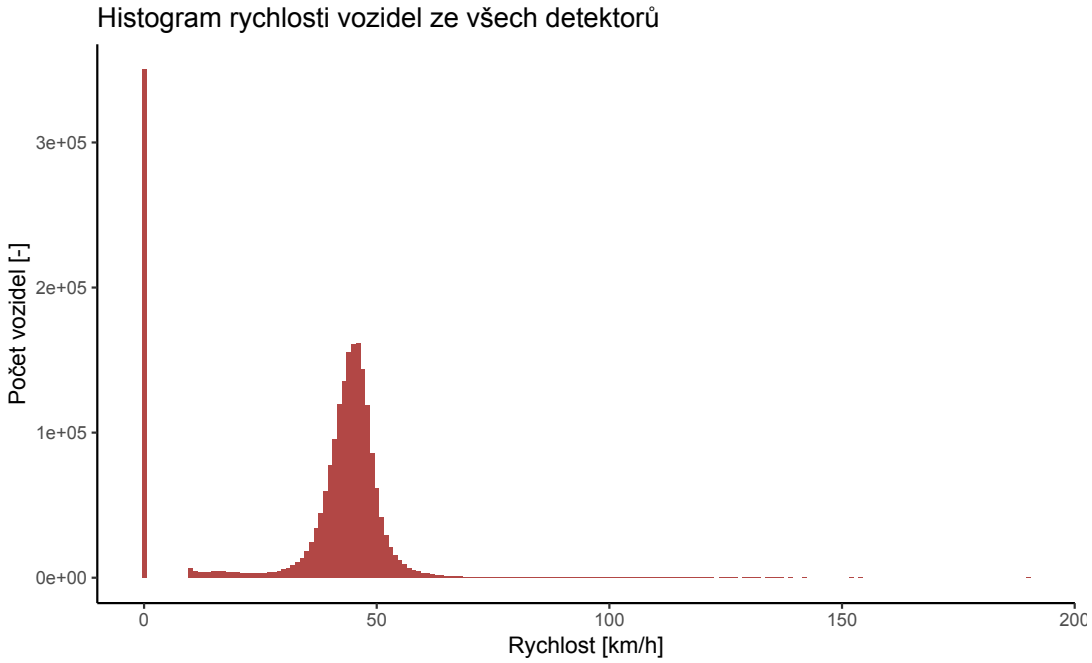
Nejprve jsou zkoumána data ze všech detektorů dohromady.

Rychlost

Graf na Obrázku 3.3 zobrazuje rozložení rychlostí ze všech čtyř detektorů v oblasti, za celé období měření. Na grafu jsou vidět tři zajímavosti.

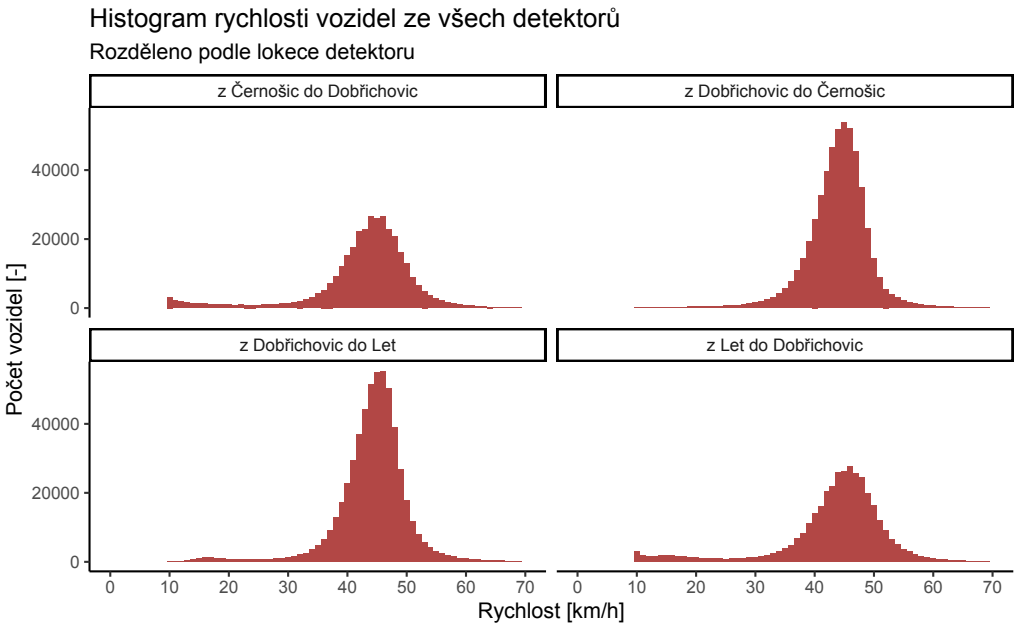
1. Velké množství měření s rychlostí 0 km/h
2. Normální rozdělení se střední hodnotou těsně pod hodnotou maximální povolené rychlosti
3. Dlouhou osu x, což značí že se v datech vyskytují měření, která značně překračují maximální povolenou rychlost

Měření s nulovou rychlostí jsou označena za chybná. Vyskytují se v nočních hodinách na úsecích *z Černošic do Dobřichovic* a *z Let do Dobřichovic*.



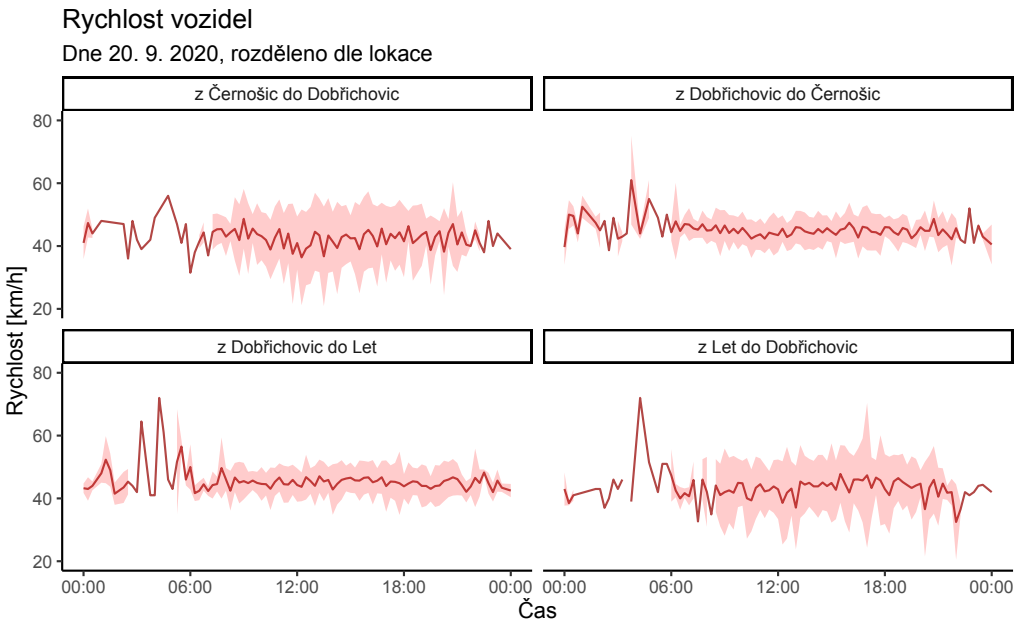
Obrázek 3.3: Histogram naměřených rychlostí ze strategických detektorů v Dobřichovicích

Grafy na Obrázku 3.4 zobrazují rozložení rychlostí za celé období měření, rozděleny dle umístění detektoru a bez odlehlých hodnot. Je zde vidět, že střední hodnota na všech úsecích zůstává podobná. Na úsecích *z Černošic do Dobřichovic* a *z Let do Dobřichovic* došlo k chybě měření, kdy byla vozidla detekována, ale rychlost byla chybně zaznamenána, proto je počet vozidel na těchto úsecích značně menší.



Obrázek 3.4: Histogramy naměřených rychlostí z Dobřichovic

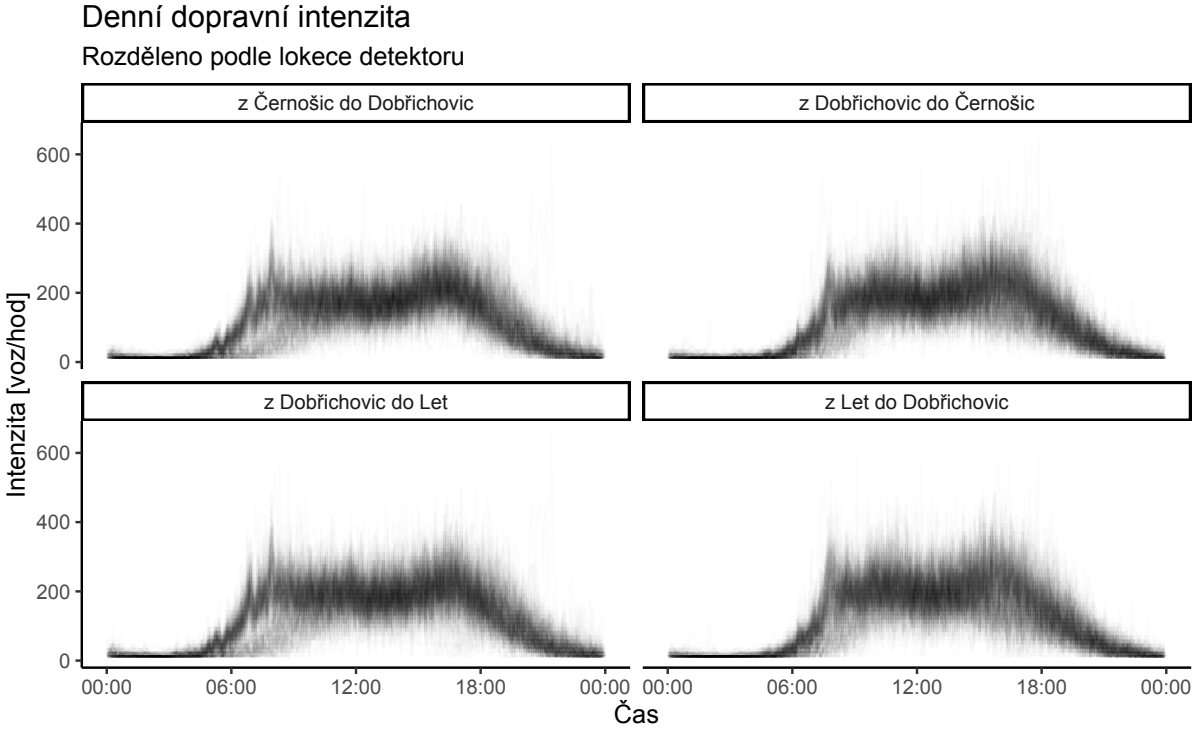
Na Obrázku 3.5 jsou grafy znázorňující rychlost vozidel v náhodně vybraný den, rozděleno dle lokace. Rychlost je do grafů zanesena po patnácti minutových intervalech, jako průměr všech měření v intervalu, světle červeně je podél křivky znázorněna směrodatná odchylka. Na všech úsecích jsou vidět případy překročení maximální povolené rychlosti 50 km/h v ranních hodinách. Při porovnání s grafem na Obrázku 3.8 lze usoudit, že k překročení maximální povolné rychlosti dochází v době, kdy je nízká dopravní intenzita.



Obrázek 3.5: Rychlost vozidel ze strategických detektorů v Dobřichovicích

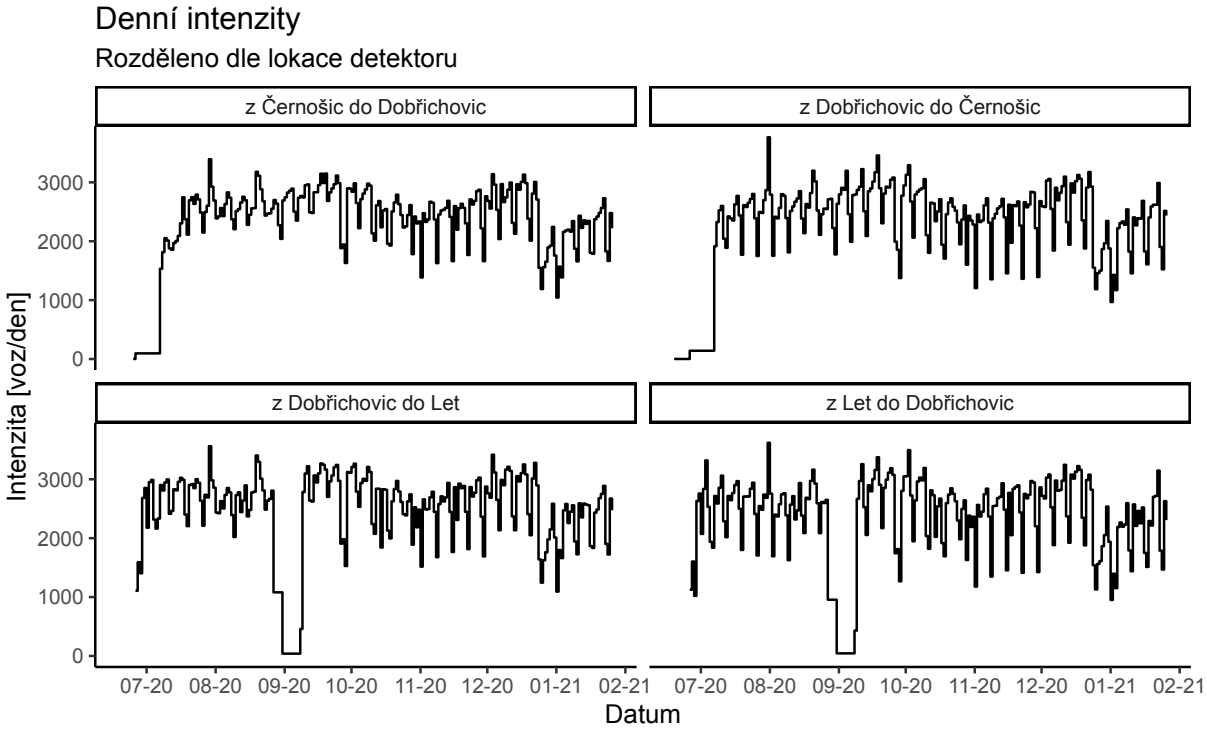
Počty vozidel

Na Obrázku 3.6 jsou znázorněny grafy počtů vozidel na jednotlivých stacionárních detektorech v Dobřichovicích. Každý graf zobrazuje data za celé období. Jednotlivé dny jsou do grafu zaneseny velice průhlednou křivkou (alpha = 0.01). Při překrytí většího množství těchto křivek se barva výsledného obrazce postupně vyplňuje, díky čemuž lze lépe identifikovat obecné dopravní trendy.



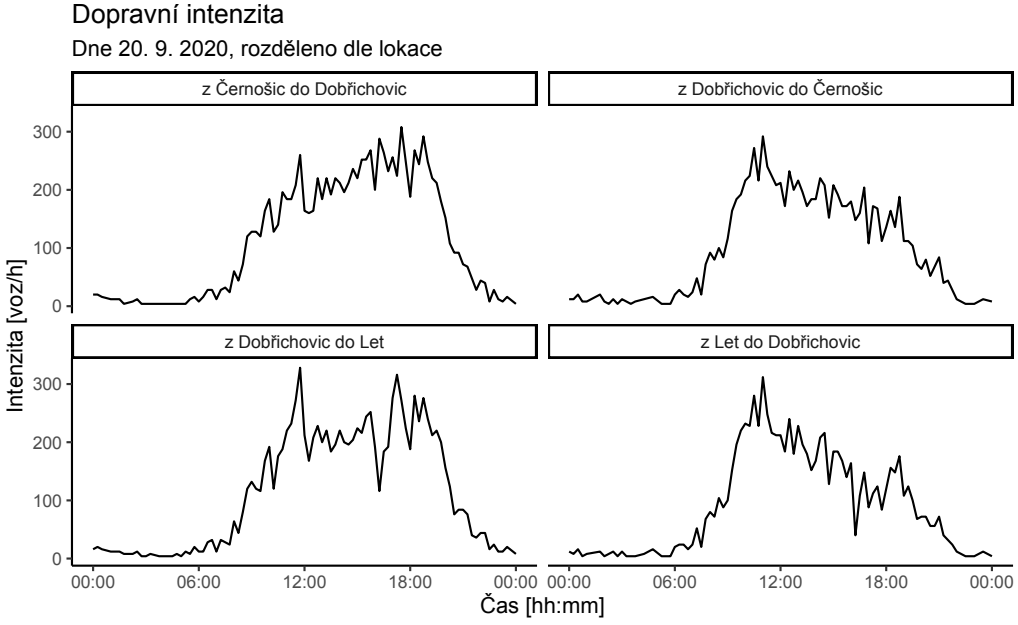
Obrázek 3.6: Dopravní intenzita ze strategických detektorů v Dobřichovicích

Na Obrázku 3.7 jsou zobrazeny denní sumy počtů vozidel z jednotlivých detektorů v Dobřichovicích. Z grafů lze vyčíst dva důležité poznatky, v datech se vyskytují výpadky měření a výskyt sezónních trendů. Výpadky měření se vyskytují na všech úsecích, na úsecích *z Černošic do Dobřichovic* a *z Dobřichovic do Černošic* je výpadek na počátku měření, ve zbylých dvou úsecích je výpadek přibližně 6 týdnů od začátku měření. Ke konci prosince lze v datech zpozorovat vliv Vánočních svátků na dopravní intenzitu.



Obrázek 3.7: Denní počty vozidel ze strategických detektorů v Dobřichovicích

Na Obrázku 3.8 jsou vyjádřeny dopravní intenzity z jednoho náhodně zvoleného dne na vybraném úseku. Z grafů je vidět, že v době ranní špičky se pohybuje více vozidel ve směru *z Dobřichovic do Černošic* než *z Černošic do Dobřichovic*. Na úseku *z Let do Dobřichovic* je vidět nižší odpolední dopravní špička oproti ostatním usekům.



Obrázek 3.8: Počty vozidel ze strategických detektorů v Dobřichovicích

3.4 Analýza FCD z Dobřichovic

Získaná FCD jsou ukládána v standardizovaném datovém formátu DATEX II. Jednotlivé atributy datové sady jsou zde stručně popsány dle dokumentace Ředitelství silnic a dálnic (ŘSD) [23] a předchozích zkušeností. Detailnější popis jednotlivých charakteristik dat je dostupný v mé bakalářské práci [24].

Název atributu	Popis atributu
tmc_id	Identifikační kód TMC segmentu
komunikace	Kategorie a označení silniční komunikace
Oblast	Oblast ve které se TMC segment nachází
Směr	Určuje počáteční a koncový bod TMC segmentu
Datum	Časová stopa pořízení záznamu
Počet osobních vozidel	Počet osobních vozidel (FCD) na segmentu
Počet nákladních vozidel	Počet nákladních vozidel (FCD) na segmentu
Typická rychlost volného průjezdu	Průměrná rychlost (km/h) při volném průjezdu
Typická doba volného průjezdu	Průměrná doba volného průjezdu
Aktuální rychlost	Vypočtená rychlost dopravního proudu
Aktuální čas průjezdu	Průměrná doba průjezdu vozidla segmentem
Lokalizace kolony	Určuje výskyt dopravní kongesce
Délka kolony	Délka dopravní kongesce v metrech
Lokace kolony	Určuje vzdálenost čela kolony od počátku segmentu
Míra spolehlivosti	Kvalitativní parametr charakterizující kvalitu dat
Stupeň dopravy	Průměrná doba průjezdu vozidla segmentem

Tabulka 3.2: Popis atributů datové sady FCD z Dobřichovic

Každý záznam v datovém souboru označuje aktuální stav dopravního proudu s přesností na minuty (data z FCD vozidel jsou agregována po minutách) na daném TMC segmentu. V dostupném datovém souboru z Dobřichovic jsou záznamy z následujících TMC segmentů:

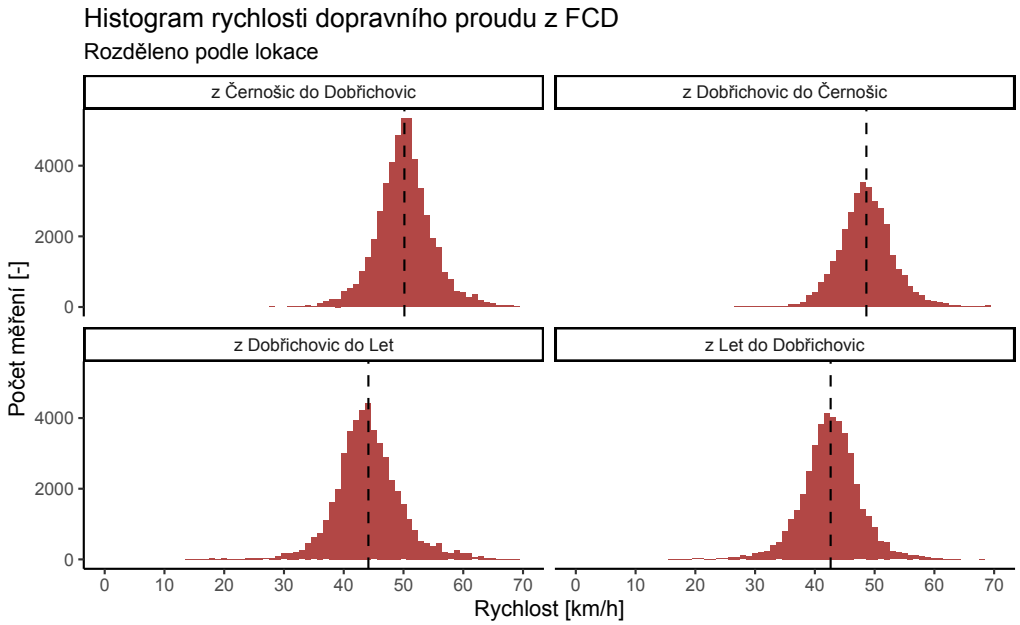
- z Letů směr Dobřichovice
- z Hlásné Třebáně směr Lety
- z Černošic směr Dobřichovice
- z Letů směr Hlásná Třebáň
- z Dobřichovic směr Lety

- z Letů směr Řevnice
- z Dobřichovic směr Černošice
- Od Řevnic směr Lety

Z dat jsou vybrány pouze směry, které jsou dostupné i ze strategických detektorů (Dobřichovice - Černošice a Dobřichovice - Lety).

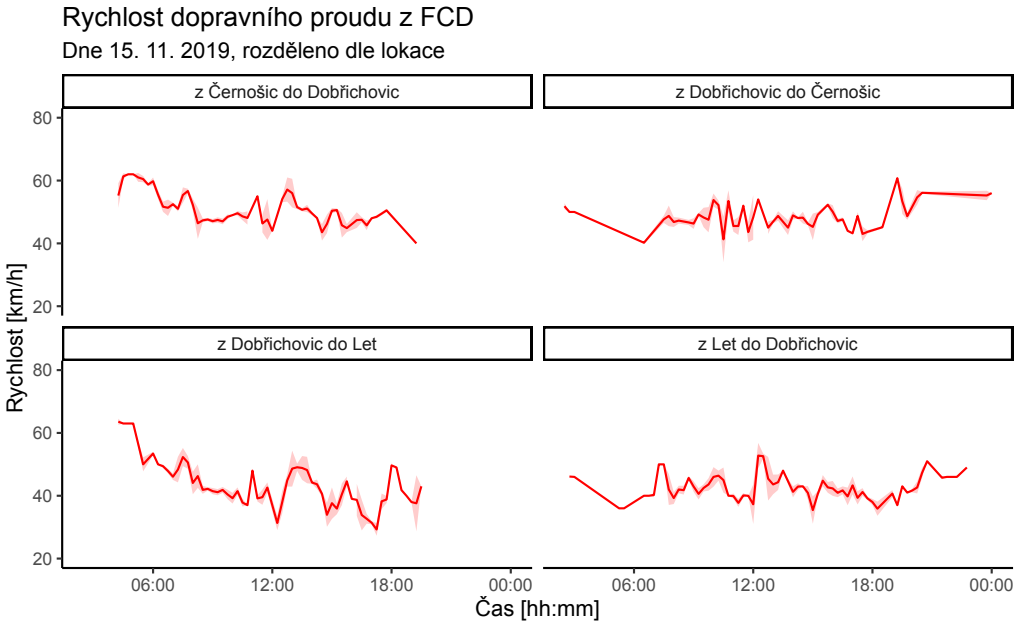
Rychlost

Grafy na Obrázku 3.9 zobrazují histogramy rychlosti dopravního proudu rozděleny podle silničního segmentu. Přerušovaná černá čára značí střední hodnotu u každé skupiny měření. Narozdíl od histogramu rychlostí pro měření ze stacionárních senzorů (Obrázek 3.4) se střední hodnoty mezi jednotlivými úseky výrazněji liší. Na úsecích *z Černošic do Dobřichovic* a *z Dobřichovic do Černošic* se střední hodnota blíží *50 km/h*, kdežto na zbylých úsecích je střední hodnota blíže *40 km/h*.



Obrázek 3.9: Histogramy rychlostí dopravního proudu z FCD

Na Obrázku 3.10 jsou zobrazeny rychlosti dopravního proudu v jeden náhodně zvolený den. Měření jsou průměrována po patnácti minutách se směrodatnou odchylkou vyznačenou světle červenou barvou. Hodnoty rychlosti se skokově mění s malou odchylkou, což je způsobeno malým počtem měření.

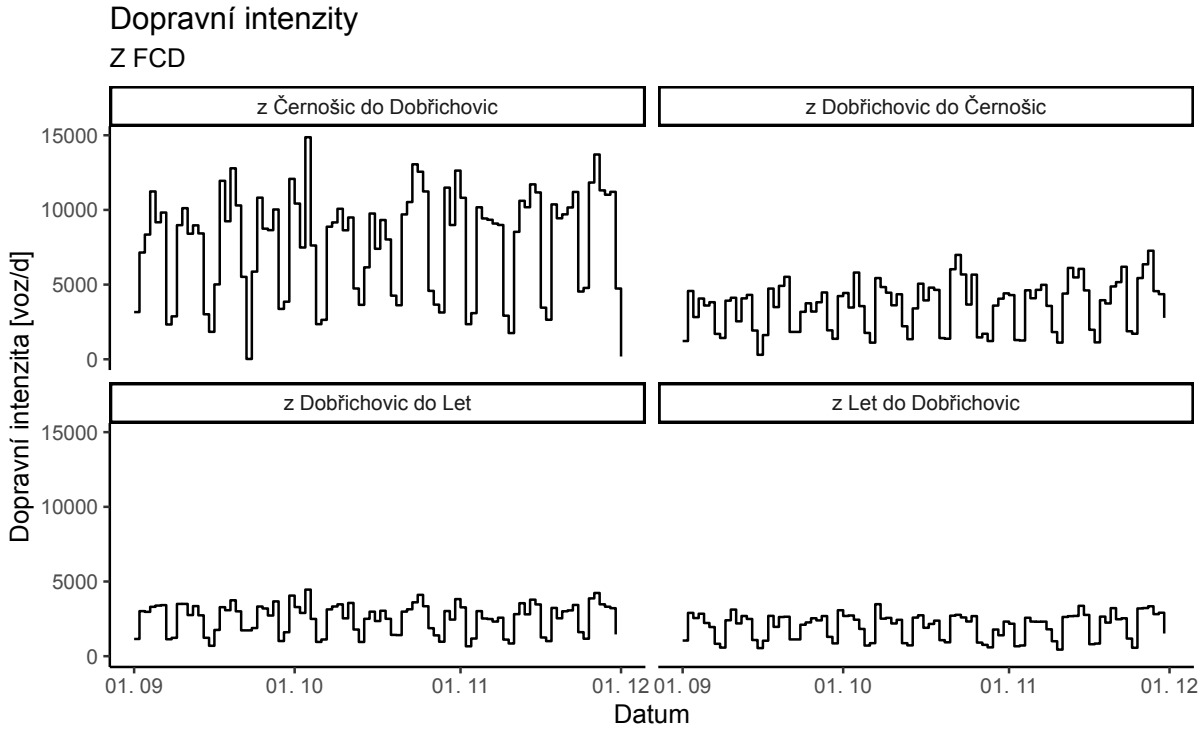


Obrázek 3.10: Rychlost dopravního proudu z FCD v Dobřichovicích

Dopravní intenzita

Dopravní intenzita zjištěná z FCD není přímo rovnocenná intenzitě zjištěné ze stacionárních detektorů. Při sběru FCD se data anonymizují a agregují na interval jedné minuty, takže není možné zpětně identifikovat konkrétní vozidla. Také musí být brána v úvahu penetrace dopravního proudu plovoucími vozidly (podíl plovoucích vozidel a všech vozidel v dopravním proudu). Z FCD lze tedy určit dopravní trend, ale hodnoty dopravní intenzity přímo neodpovídají realitě.

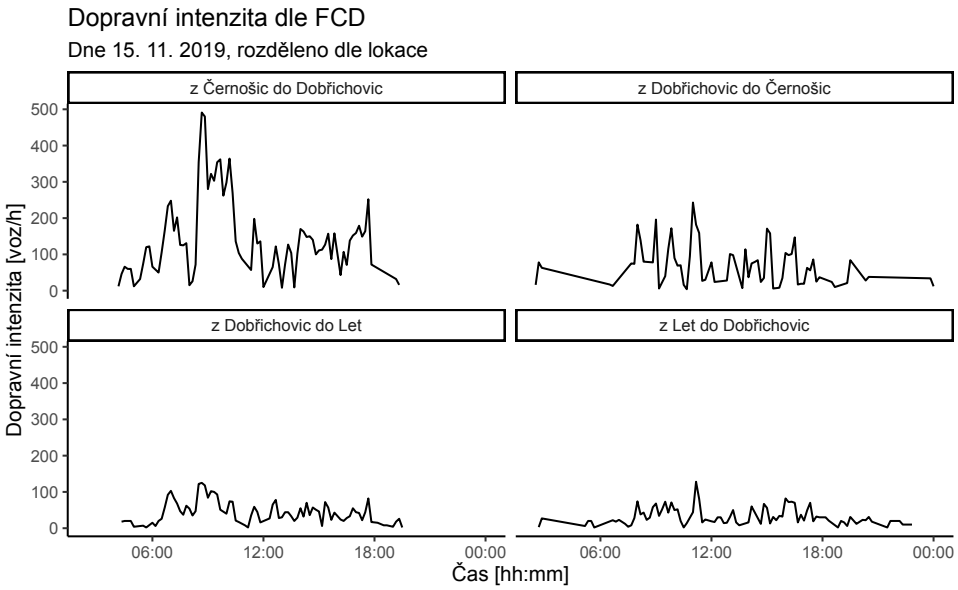
Počet vozidel FCD tedy vlastně není určen z bodového měření (jako u stacionárních detektorů), ale jako počet vozidel vyskytujících se na segmentu silniční komunikace. Pokud vozidlo projíždí daným segmentem více než jednu minutu, je zaznamenáno v datech vícekrát (je přičteno do každé minutové agregace).



Obrázek 3.11: Počty vozidel z FCD v Dobřichovicích

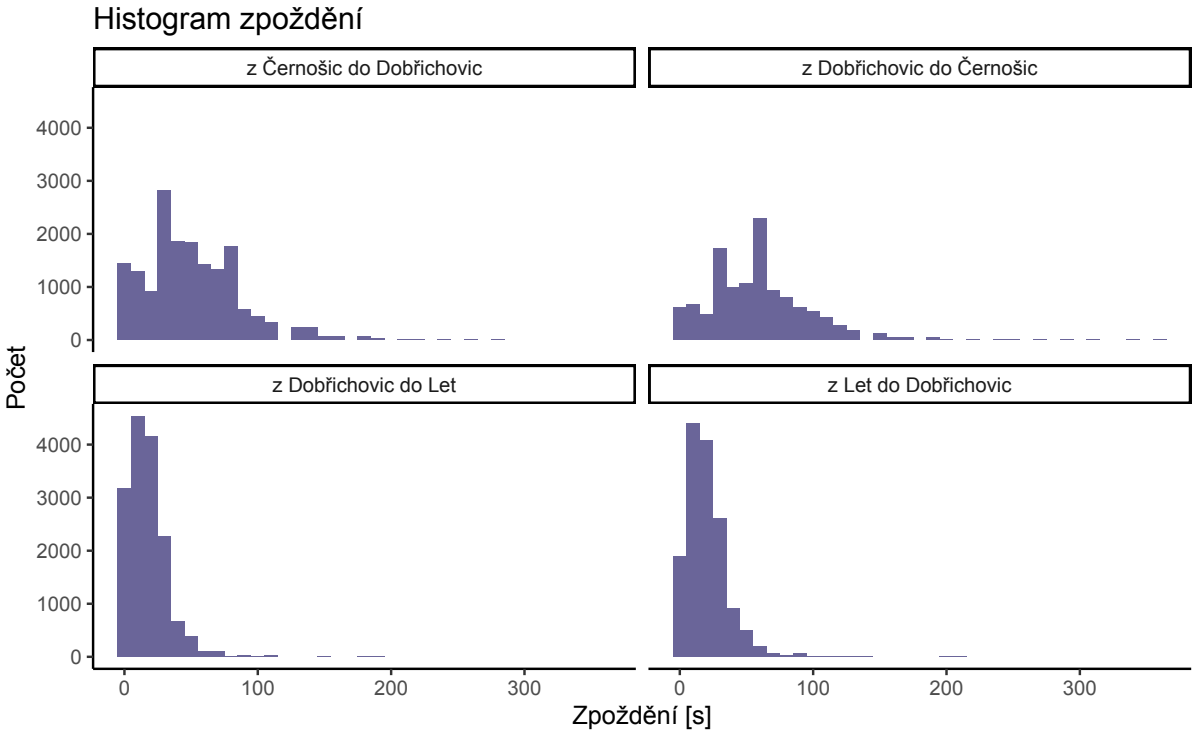
Na Obrázku 3.11 jsou zobrazeny sumy počtů naměřených plovoucích vozidel jako denní intenzity. Na úseku *z Černošic do Dobřichovic* jsou hodnoty značně navýšeny oproti zbylým segmentům. Takovýto stav je značně nepravděpodobný, jedná se o chybu měření, jejíž příčina je neznámá.

Na Obrázku 3.12 jsou zobrazeny dopravní intenzity dopočteny pomocí FCD, z jednoho náhodně vybraného dne. Oproti dopravním intenzitám zjištěných ze stacionárních detektorů je mnohem těžší nalézt v datech jasné dopravní trendy.



Obrázek 3.12: Počty vozidel z FCD v Dobřichovicích

FCD neměří pouze intenzitu a rychlost, ale i další veličiny informující o stavu dopravního proudu, jako jsou výskyt dopravní kongesce a aktuální zpoždění na silničním segmentu. Na Obrázku 3.13 jsou zobrazeny histogramy zpoždění na vybraných segmentech pozemní komunikace. Z grafů je vidět, že zpoždění mezi Dobřichovicemi a Lety bývá kratší než mezi Černošicemi a Dobřichovicemi, což odpovídá kratší vzdálenosti mezi Dobřichovicemi a Lety.



Obrázek 3.13: Zpoždění vozidel z FCD v Dobřichovicích

3.5 Volba datové sady pro predikční modely

Pro trénování predikčních modelů jsou použity dvě datové sady. První datová sada je ze stacionárního detektoru v Dobřichovicích. Jedná se o detektor na ulici Pražská ve směru z Dobřichovic do Černošic. Za období 19. 6. 2020 - 26. 1. 2021 je naměřeno 500 tisíc záznamů. Z grafů na Obrázku 3.7 vyplývá, že za celé období na tomto úseku nedošlo k výpadku dat. Tento úsek byl zvolen, především kvůli dobré kvalitě dat.

Druhá datová sada je ze stejného silničního úseku na ulici Pražská ve směru z Dobřichovic do Černošic. Zvolené období je od září do prosince roku 2019. Bohužel data z roku 2020 jsou v nedostatečné kvalitě, proto jsou použita dat z roku 2019.

Kapitola 4

Popis predikčních modelů a jejich použití pro predikci dopravy

Tato kapitola nejprve popisuje a poté aplikuje predikční modely používané v dopravě. Modely jsou aplikovány na datové sady popsány v předešlé kapitole.

4.1 Obecný popis predikčních modelů v dopravě

Pro zlepšování efektivity silničního provozu pomocí ITS je klíčové usnadnit řidičům výběr optimální trasy. Nestačí jen umět využívat informace o provozu v reálném čase, ale je nutné také předpovědět, parametry dopravního provozu v nějakém časovém horizontu. To je nezbytným předpokladem pro zlepšení efektivity dopravy. [25]

Predikční modely v dopravě lze rozdělit dvěma způsoby. Podle doby predikce a podle druhu predikčního modelu.

4.1.1 Predikční modely podle doby predikce

Predikční modely podle doby predikce se dělí na dvě kategorie, modely pro krátkodobou a pro dlouhodobou predikci. Krátkodobé predikční modely předpovídají stav dopravního proudu s předstihem 5 až 15 minut dopředu. Pro krátkodobou předpověď je potřeba velké množství dat.

Dlouhodobé predikční modely jsou schopny predikovat s předstihem dnů, týdnů a i déle. Jsou využívány hlavně k identifikaci dopravních špiček, týdenních i sezonních změn. Také mohou sloužit jako porovnání ke krátkodobé predikci, nebo v kombinaci s krátkodobou predikcí. [25]

4.1.2 Predikční modely podle druhu predikčního modelu

Statistické učení označuje sadu nástrojů pro tvorbu statistických modelů pro odhad výstupu na základě jednoho nebo více vstupů. Tyto nástroje lze klasifikovat nejprve podle druhu učení na nástroje učením s učitelem (supervised) a učením bez učitele (unsupervised). Tato klasifikace je tříděna podle potřeby trénovacích dat. [26]

Pro predikci veličin dopravního proudu je vždy nutné znát historické hodnoty sledovaných veličin. Všechny zde popsané metody patří mezi metody učením s učitelem. Predikční modely je možné následně rozdělit podle metody jejich tvorby. Od jednodušších statistických modelů po složité modely strojového a hlubokého učení.

Statistické modely

Statistické modely lze použít k popisu vztahu mezi jednotlivými veličinami, k identifikaci vzorců a trendů v datech a k odhadu parametrů výchozí distribuce. Statistický model je obvykle stanoven jako matematický vztah mezi náhodnou a nenáhodnou proměnnou, nebo mezi více náhodnými a nenáhodnými proměnnými. V dopravě se používají statistické modely časových řad pro analýzu stavu dopravního proudu.

Pomocí statistických metod lze rozpoznat chování dopravy v různých měřítkách, jako je denní, týdenní, sezónní atd. Ve srovnání s metodami strojového učení jsou statistické metody obvykle jednodušší, rychlejší a nákladově efektivnější. Jejich přesnost je však relativně nižší, protože nedokážou zpracovat tolik vícerozměrných dat. [27]

Metody strojového učení

Metody strojového učení jsou podskupinou umělé inteligence. Jsou založeny na schopnosti počítačů (strojů) učit se na základě zkušeností z trénovacích dat (v případě modelů s učitelem). Každý model strojového učení obsahuje učící algoritmus, který převede vstupní trénovací data na zkušenost. Na základě získané zkušenosti je poté možné automaticky vykonávat nějakou určitou činnost. [28]

Neuronové sítě a metody hlubokého učení

Hluboké učení (deep learning) je podkategorií strojového učení. Popularita neuronových sítí začala v posledním desetiletí stoupat díky pokrokům ve výpočetní technice a dostupnosti většího množství dat. Mezi metody hlubokého učení patří neuronové sítě s třemi a více vrstvami. K jejich popularitě také přispěla dostupnost softwaru jako je *tensorflow* od společnosti Google. [26] V dopravě se používají rekurentní neuronové sítě pro predikci dopravního proudu.

4.1.3 Výběr predikčních modelů

Predikční modely použité v předkládané práci byly zvoleny na základě několika přístupů. V první řadě byly použity modely, které se aktuálně používají pro predikci veličin dopravního proudu. Dále byly použity modely, které jsou často využívané pro obecnou predikci časových řad. A nakonec jsou vyzkoušeny state-of-the-art modely.

Model ARIMA je jedním z nejznámějších a nejrozšířenějších modelů predikce časových řad. Billings, et al. [29] použili model ARIMA pro predikci dojezdové doby na úseku dálnice v Minnesotě, USA. Jejich dlouhodobý predikční model použil variaci ARIMA pro ne-stacionární data a dosáhl výsledků s přiměřenou přesností.

Alghmadi, et al. [30] zkoumali využití modelu ARIMA pro predikce dopravní intenzity, nad tříměsíční datovou sadou z dálnic ve státě California, USA. Porovnávali také možnosti manuálního výběru parametrů modelu s automatickým generováním optimálních parametrů modelu pomocí algorimu *auto.arima* dostupném v programovacím jazyce R. Výsledný model, vytvořený algoritmem *auto.arima*, dosáhl nejlepších výsledků v porovnání s ostatními.

Dong, et al. [31] použili model ARIMA pro predikci dopravní intenzity u dat s různou periodou. Autoři dospěli k uspokojivým výsledkům predikce a přepokládají, že zahrnutí prostorových vlastností sledovaných pozemních komunikací přinese lepší výsledky.

Jain, et al. [32] použili kombinaci modelů Prophet a XGBoost pro předpověď provozu v telekomunikačních sítích. Autoři zvolili tyto modely, protože jsou jednou z nejpokročilejších a nejúspěšnějších metod používaných pro strojové učení, které byly vyvinuty v posledních letech.

Zhang, et al.[33] použili pro krátkodobou predikci dopravní intenzity model k-nejbližších sousedů (KNN). Přesnost modelu byla dobrá, protože byl model trénován na velkém vzorku dat.

Leshem, et al. [34] použili algoritmus AdaBoost s modelm Random Forest pro predikci výskytu kongescí v daný čas na daných křižovatkách. Autoři navrhli metodu predikce, která měla velice slibné výsledky na simulacích a i na reálných datech.

Dong, et al. [35] porovnali využití XGBoost modelu a modelu Support Vector Machine pro predikci dopravní intenzity a jako lepší se ukázal XGBoost.

Fu, et al. [36] použili model neuronové sítě pro predikci dopravní intenzity a porovnávali jej s modelem ARIMA. Jejich model neuronové sítě predikoval dopravní intenzitu trochu lépe než model ARIMA.

Wang, et al. [37] použili predikční model Prophet pro předpověď intenzit v námořní dopravě. Model Prophet zde ukazoval dobré výsledky. Autoři také navrhli kombinaci modelu Prophet a *discrete wavelet decomposition*, která měla o trochu lepší výsledky než samostatný model Prophet.

Vybrané modely

Zde je předložen seznam vybraných modelů na základě přehledu literatury, možnostech programovacího jazyka R a zastoupení všech druhů predikčních modelů.

- Naivní modely
- ARIMA
- Prophet
- K-NN
- Random Forest
- XGBoost
- Prophet s XGBoost
- Autoregresivní neuronová síť

Naivní modely byly použity pro srovnání s ostatními modely. Naivní modely mohou sloužit pro výpočet metrik přesnosti (MASE). Může se tedy stát, že metrika přesnosti hodnotí model Naivní lépe než posuzovaný model. Jedná se o velice jednoduchý model, v této práci jsou použity dvě varianty tohoto modelu. Prostý naivní model se zpožděním jednoho časového kroku (použit pouze pro výpočet MASE) a sezónní naivní model se zpožděním jedné sezóny.

Model ARIMA byl vybrán, na základě své velké popularity mezi modely pro predikci časových řad. Tento model je často používá pro predikci dopravní intenzity a také jako model se kterým se srovnávají nově navržené modely. Tento model je zástupcem statistických modelů.

Model Prophet byl vybrán jako novodobá alternativa modelu ARIMA. Tento model se od doby jeho uvedení (2017) začíná čím dál častěji používat pro predikci časových řad. Nicméně v oblasti predikce silniční dopravy je dosud méně využíván, což je dalším důvodem pro prozkoumání jeho využitelnosti na dostupná data. Tento model je také zástupcem statistických modelů.

Model KNN je jednoduchý model strojového učení, který lze použít na klasifikaci i na regresi. Tento model vyžaduje velké množství dat. A je spíše vhodný pro krátkodobou predikci (v řádu několika hodin).

Model Random Forest je často používaná metoda strojového učení převážně pro klasifikační a regresní problémy. Tento model lze použít i pro predikci časových řad, v dopravě byl také využit k predikci dopravní intenzity. Je však možné jej využít i na jiné klasifikační či regresní problémy v dopravě.

Model XGBoost je nová pokročilejší alternativa modelu Random Forest. Jedná se také o model strojového učení, který lze využít pro regresi a klasifikaci. Tento model se rovněž používá i pro predikci časových řad. Uplatnění tohoto modelu pro predikci dopravní intenzity v silniční dopravě nebylo doposud popsáno. O to je zajímavější zhodnotit využití XGBoost na dostupná dopravní data.

Hybridní model Prophet a XGBoost vznikl kombinací dvou zmíněných modelů s cílem přesnější a kvalitnější predikce. Tento model zatím nebyl využit pro predikci v dopravě. Což je dobrým důvodem pro jeho zhodnocení na dostupných datech v předkládané práci.

Model autoregresivní neuronové sítě (NNAR) byl zvolen jako zástupce metod hlubokého učení a neuronových sítí. Jedná se o jednoduchý model neuronové sítě, který je vhodný pro predikci časové řady. Modelů neuronových sítí pro predikci v dopravě existuje mnoho, tento byl vybrán kvůli snadné implementaci v programovacím jazyce R.

4.2 Naivní sezónní model

Jak jeho jméno napovídá naivní model je jednoduchý model, který přiřadí předpovědi hodnotu posledního pozorování. Model lze zapsat matematicky:

$$\hat{y}_{t+h} = y_t \tag{4.1}$$

kde $\hat{y}_{t+h}$ je predikovaná hodnota a y_t je předchozí hodnota opožděná o y_h. Tato jednoduchá metoda v určitých situacích poskytuje překvapivě dobré výsledky. Proto se používá pro výpočet metriky přesnosti MASE pro ostatní predikční modely. Metrika MASE je popsána v kapitole 5.

Sezónní naivní model je obdobná metoda pro sezónní data. Oproti klasickému naivnímu modelu jsou hodnotám predikce přiřazeny poslední naměřené hodnoty z poslední sezóny. [38] [39] Matematicky lze tento model zapsat:

$$\hat{y}_{t+h} = y_{t+h-T} \tag{4.2}$$

kde T je sezonní perioda.

4.3 Použití Naivního sezónního modelu na dostupná data

Aplikace modelu sezónního naivního modelu, i všech nadcházejících modelů jsou provedeny v programovacím jazyce R [40] doplněném o balíčky potřebné pro tvorbu predikčních

modelů (tidymodels [41], modeltime [42], tidyverse [43], lubridate [44], timetk [45]). Programovací jazyk R a všechny použité balíčky jsou volně dostupné z The Comprehensive R Archive Network.

Tvorba většiny modelů následuje podobný rámec. Prvním krokem tvorby modelu je transformace čistých dat (která byla připravena v rámci EDA), tak aby v jednom řádku bylo jedno měření a ve sloupcích byla časová stopa měření a zkoumaná proměnná (dopravní intenzita).

Následně se data rozdělí na testovací a trénovací sadu. Toto rozdělení proběhlo tak, aby následovalo metodiku křížové validace, která je popsána v kapitole 5.2. Poté se trénovací sada použije na natrénování modelu.

```
model_fit_naive <- naive_reg() %>%
    set_engine("snaive") %>%
    fit(intenzita ~ ., data = training(d_c_data_split))
```

Ukázka kódu 1: Trénování modelu sNaive

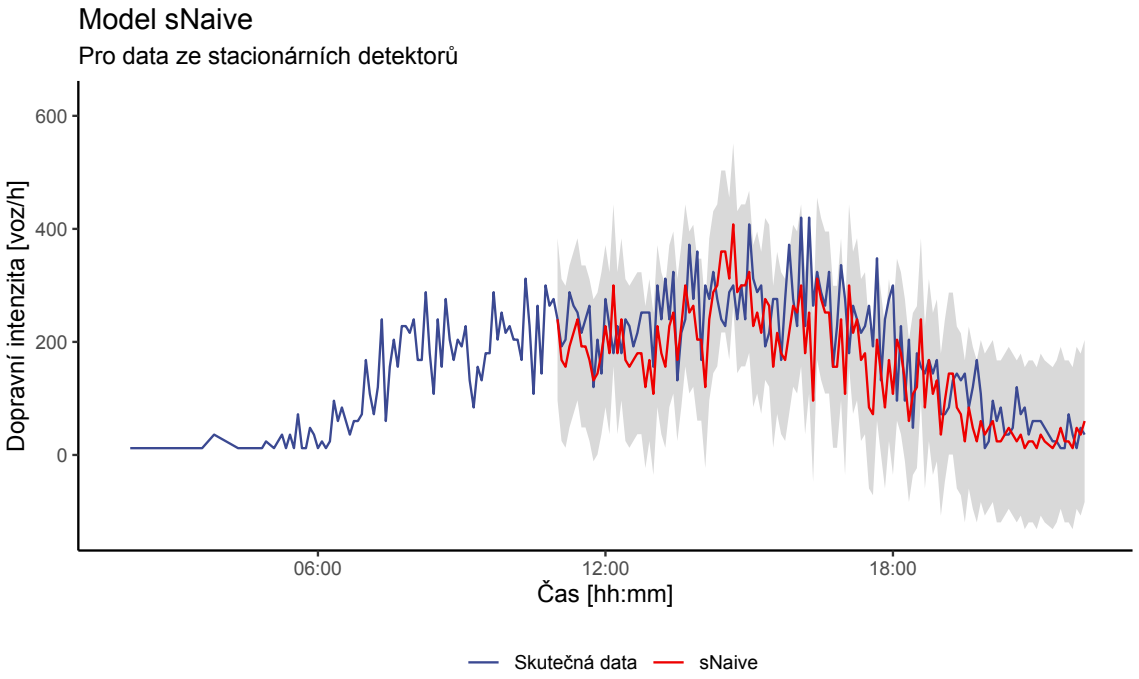
Trénovací sada je v Ukázce kódu 1 využita pro natrénování modelu. Nejprve je vybráno obecné rozhraní pro regresní model (zde konkrétně rozhraní pro naivní modely *naive_reg()* z balíčku *modeltime*), poté je použita funkce *set_engine()* pro zvolení konkretního modelu (v tomto případě je vybrán model *snaive*). Nakonec je specifikována regresní proměnná (v tomto příkladě se jedná o dopravní intenzitu) v obecné modelovací funkci *fit()*, která odhadne parametry pro daný model na základě dat (v tomto případě pouze specifikuje regresní proměnnou, naivní model nemá žádné parametry). Za datový soubor je dána trénovací část dat.

Poté se model otestuje na testovací části dat, vypočtou se metriky přesnosti (popsány v kapitole 5) a vykreslí se grafy. Kompletní kód je dostupný v příloze.

Všechny vytvořené modely jsou v následující kapitole 5 hodnoceny pomocí křížové validace a metrik přesnosti.

4.3.1 Data ze stacionárních detektorů v Dobřichovicích

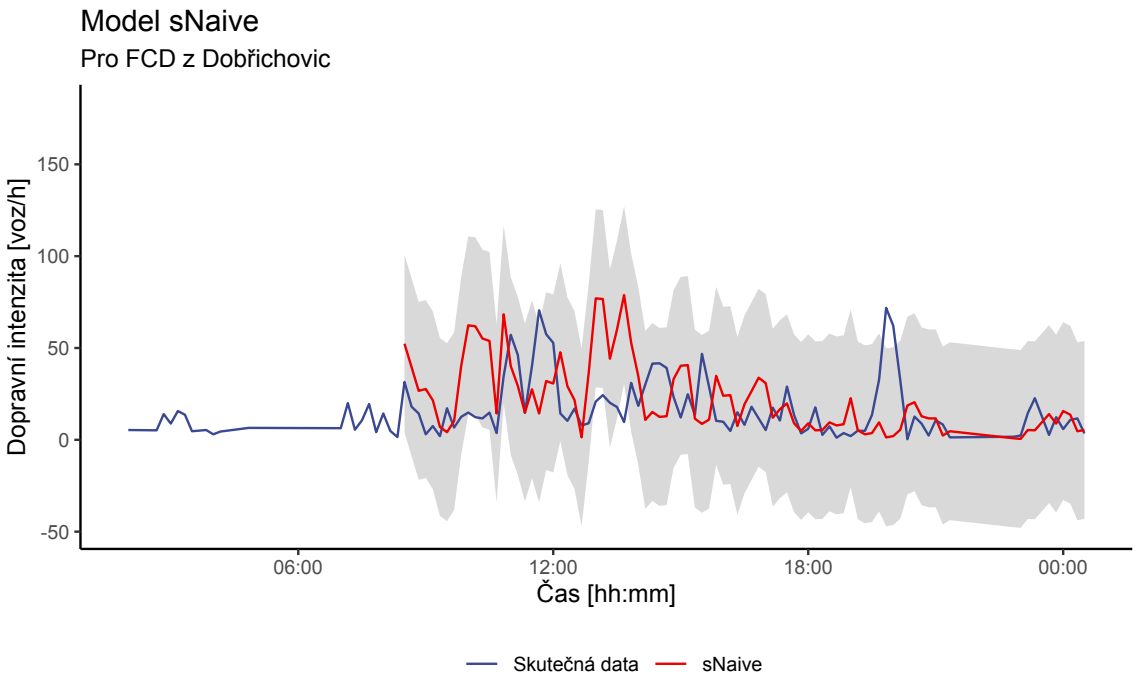
Model sNaive pro data ze strategických dopravních detektorů je ukázán na Obrázku 4.1. Predikované hodnoty celkem odpovídají trendu skutečných dat. Kdyby tento model predikoval například dopravní intenzitu v sobotu na základě dat z předchozího dne, výsledky by vypadaly podstatně hůř, což bude dokázáno pomocí křížové validace. Šedý pruh podél predikovaných hodnot značí 95% konfidenční interval (v tomto i ve všech následujících grafech, které znázorňují predikční křivky modelů).



Obrázek 4.1: Model sNaive

4.3.2 FCD v Dobřichovicích

Na datech získaných pomocí plovoucích vozidel se tomuto modelu vůbec nepodařilo držet skutečných dat, což je vidět na Obrázku 4.2. Je vidět, že aby tento model mohl fungovat, je zapotřebí, aby se postupné sezónní sekce navzájem moc nelišily.



Obrázek 4.2: Model sNaive

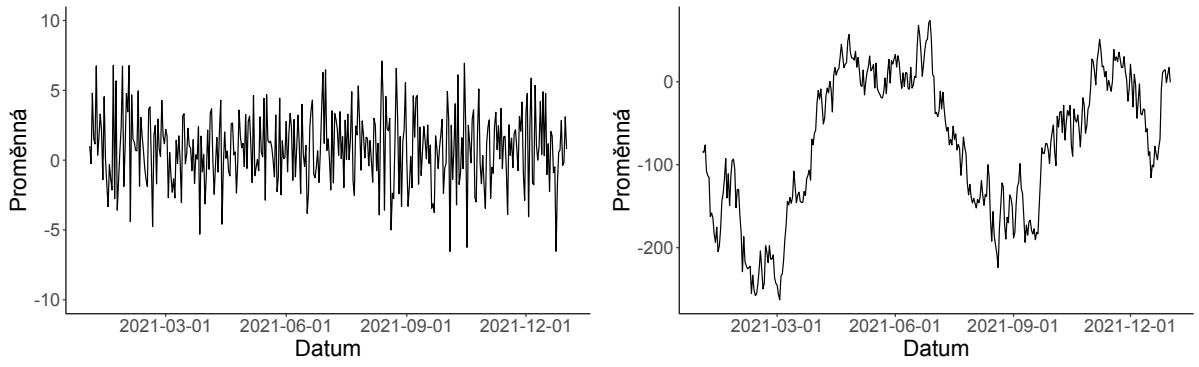
4.4 Model ARIMA

Model časové řady je takový model, kde jsou pozorování měřena postupně v čase. Pozorování v časové řadě mají pořadí a většinou existuje vztah mezi předchozí a následující hodnotou. Hlavním cílem analýzy časových řad je pochopit základní proces, který je zodpovědný za generování pozorovaných dat, a následně předpovědět budoucí hodnoty řady. [29] Od 70. let 20. století jsou autoregresní integrované modely klouzavých průměrů (Autoregressive integrated moving average) široce využívány pro předpovídání dopravy díky své snadné implementaci a vyšší přesnosti ve srovnání s jinými statistickými metodami. [27]

4.4.1 Předpoklady pro tvorbu modelu

Pro porozumění modelu ARIMA je nutné nejprve definovat koncept stacionarity dat a techniky diferencování časových řad. [38] Stacionární datové řady jsou takové, které mají stabilní střední hodnotu a rozptyl, tedy nejsou závislé na čase (neovlivňují je trendy ani sezónní vlivy). Z těchto důvodů je mnohem snazší je modelovat. Metody statistického modelování předpokládají nebo vyžadují, aby časové řady byly stacionární. [46]

Naopak v nestacionárních datech se efekty sezónních vlivů a trendů projevují. Nestacionaritu dat lze zjistit vizuálně z grafu časové řady, nebo pomocí Dickey-Fuller statistického testu. [46] Vizuální rozdíl mezi stacionární a nestacionární časovou řadou je znázorněn na Obrázku 4.3.



(a) Stacionární časová řada (b) Nestacionární časová řada

Obrázek 4.3: Stacionární a nestacionární časová řada

Transformace z nestacionární na stacionární časovou řadu je možná pomocí diferencování. Diferencování odstraní z časové řady časovou závislost (trendy a sezónnost). Diferencování probíhá jako rozdíl předchozího pozorování od aktuálního (difference(t) = pozorování(t) - pozorování($t-1$))). Může se stát, že po procesu diferencování se časová řada bude nadále jevit jako nestacionární. V takovém případě proces diferencování opakujeme dokud řada

není stacionární. Počtu opakování procesu diferencování se říká řád diferencování a značí se d.

4.4.2 Autoregrese

Autoregresní model spočívá v předpovědi proměnné, která nás zajímá, pomocí lineární kombinace jejích předešlých hodnot. Termín *autoregrese* odkazuje na skutečnost, že tento model zahrnuje regresi proměnné vůči jejím vlastním minulým hodnotám. [38] Podmínkou AR jsou stacionární data. Pokud je y_t modelováno pomocí AR modelu, je zapsáno jako:

$$y_t = \delta + \psi_1 y_{t-1} + \psi_2 y_{t-2} + ... + \psi_p y_{t-p} + \epsilon_t \tag{4.3}$$

Kde: y_t je modelovaná proměnná, δ je intercept, ψ jsou regresní koeficienty, y_{t-p} jsou regresory (opožděná data) a ϵ je chyba (šum). Řád modelu je určen maximálním zpožděním p. Pro odhad regresní koeficientů lze použít několik metod, například metodu nejmenších čtverců nebo metodu maximální věrohodnosti (Maximum likelihood estimation). [47] [38] [48]

4.4.3 Klouzavý průměr

Model klouzavého průměru (Moving average) je metoda analýzy časových řad, která se používá k predikci budoucích hodnot proměnné na základě jejího minulého chování. V modelu klouzavého průměru se místo minulých hodnot samotné proměnné používá lineární kombinace minulých chyb v předpovědích. [47] Pokud je proměnná y_t modelována pomocí MA modelu, lze jí zapsat jako:

$$y_t = \mu + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + ... + \theta_q \epsilon_{t-q} + e_t \tag{4.4}$$

Kde: y_t je modelovaná proměnná, μ je intercept, θ jsou regresní koeficienty, ϵ_{t-q} jsou regresory (chyby v předchozí predikci) a ϵ_t je chyba (šum). Řád modelu je určen zpožděním q.

4.4.4 ARIMA

Kombinací Autoregrese, Moving average a diferencování je získán ne-sezónní model Autoregressive integrated moving average (ARIMA). [38] Model ARIMA(p, d, q), kde p je řád autoregrese, d je řád diferencování a q je řád klouzavého průměru lze vyjádřit matematicky:

$$y_t = \delta + \{\psi_1 y_{t-1} + \psi_2 y_{t-2} + ... + \psi_p y_{t-p}\} + \{\theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + ... + \theta_q\} + \epsilon_t \tag{4.5}$$

Model ARIMA lze aplikovat i na sezónní data přidáním sezónních parametrů.

$$\text{ARIMA}(p,d,q)(P,D,Q)_m \tag{4.6}$$

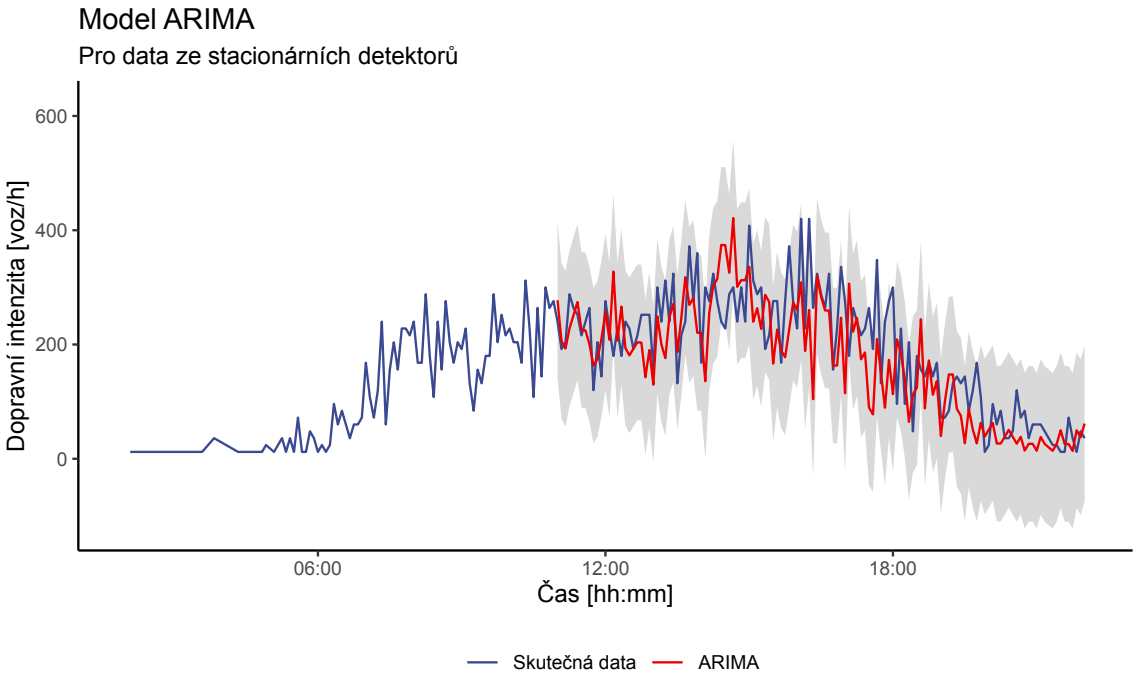
Kde (P, D, Q) jsou sezónní parametry a m je počet pozorování za sezónu. [38] [39]

4.5 Použití modelu ARIMA na dostupná data

Pro volbu optimálních hodnot (p, d, q) a (P, D, Q)$_m$ je použita funkce *auto.arima()*, která algoritmicky vybere nejlepší možné hodnoty. Pro odhad regresních koeficientů (θ a ψ) používá *auto.arima()* metodu Maximum likelihood estimation (MLE). Kompletní kód tvorby tohoto modelu je v příloze.

4.5.1 Data ze stacionárních detektorů v Dobřichovicích

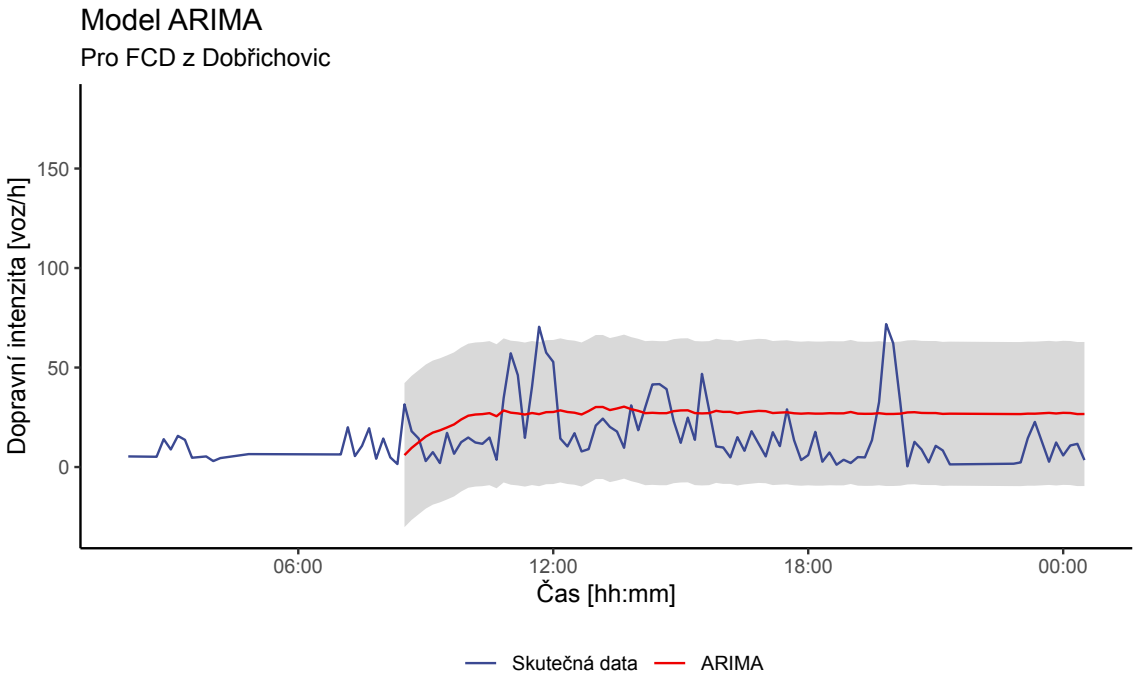
Nejprve byl model ARIMA použit na data ze strategických detektorů v Dobřichovicích. Na Obrázku 4.4 je vidět, že model ARIMA je vizuálně podobný testovacím datům. To je způsobeno převážně autoregresní částí tohoto modelu, která předpovídá hodnoty pomocí lineární kombinace předešlých hodnot.



Obrázek 4.4: Model ARIMA

4.5.2 FCD v Dobřichovicích

Nad FCD daty se modelu ARIMA nepodařilo předpovědět jakýkoli trend. Výsledná křivka připomíná spíše střední hodnotu.



Obrázek 4.5: Model ARIMA

4.6 Model Prophet

Prophet je model sloužící pro predikci dat časové řady navržen datovými analytiky firmy Meta (dříve Facebook) v roce 2017. Prophet je určen pro předpovídání dat se sezónními vlastnostmi, funguje nejlépe s několika sezónami historických dat. Model lze zapsat:

$$y_t = g(t) + s(t) + h(t) + \epsilon_t \tag{4.7}$$

Kde $g(t)$ je funkce trendu znázorňující neperiodické změny v časové řadě, $s(t)$ je funkce reprezentující periodické změny (např. týdenní, měsíční nebo roční sezónnost), $h(t)$ vyjadřuje efekty prázdnin a svátků (Vánoce, státní svátky) a ϵ_t je šum (s předpokladem normálního rozdělení). [38] [49] Konkrétní popis jednotlivých funkcí je dostupný v [49].

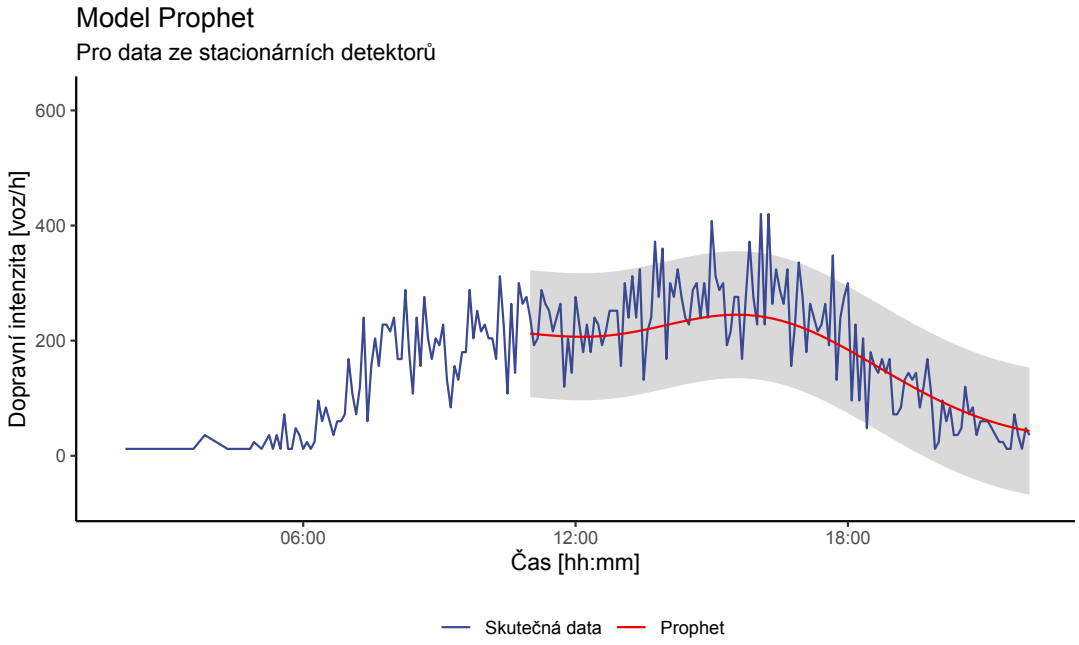
4.7 Použití modelu Prophet na dostupná data

Použití modelu Prophet na dostupná data proběhlo analogicky jako u předchozích modelů. Pro tvorbu modelu byl využit balíček Prophet [50] volně dostupný pro programovací jazyk

R. Kompletní kód je dostupný v příloze.

4.7.1 Data ze stacionárních detektorů v Dobřichovicích

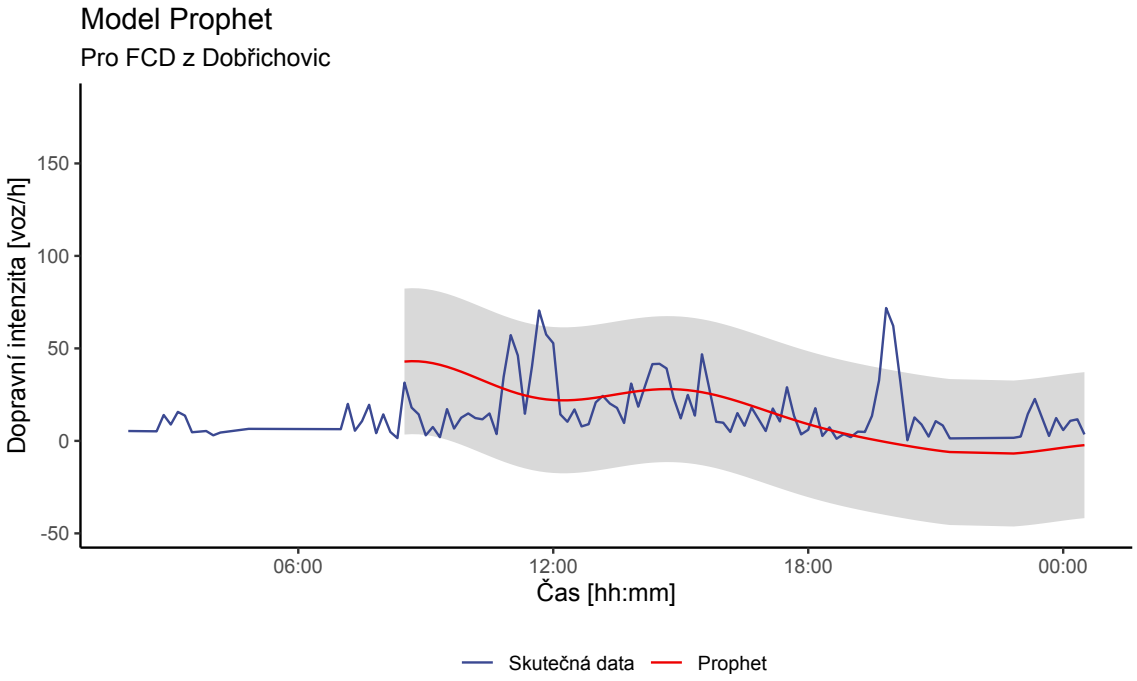
Nejprve jsou analyzována data ze strategických detektorů v Dobřichovicích. Na Obrázku 4.6 je vidět trend stanovený modelem Prophet v porovnání se skutečnými daty. Výsledná predikční křivka velice hezky detekuje odpolední dopravní špičku a většina skutečných hodnot spadá do 95% intervalu spolehlivosti. Narozdíl od modelu ARIMA (Obr. 4.4) je regresní křivka mnohem hladší.



Obrázek 4.6: Model Prophet

4.7.2 FCD v Dobřichovicích

Model Prophet si s datovou sadou FCD poradil lépe než modely předchozí, povedlo se mu alespoň detekovat odpolední špičku a večerní pokles.



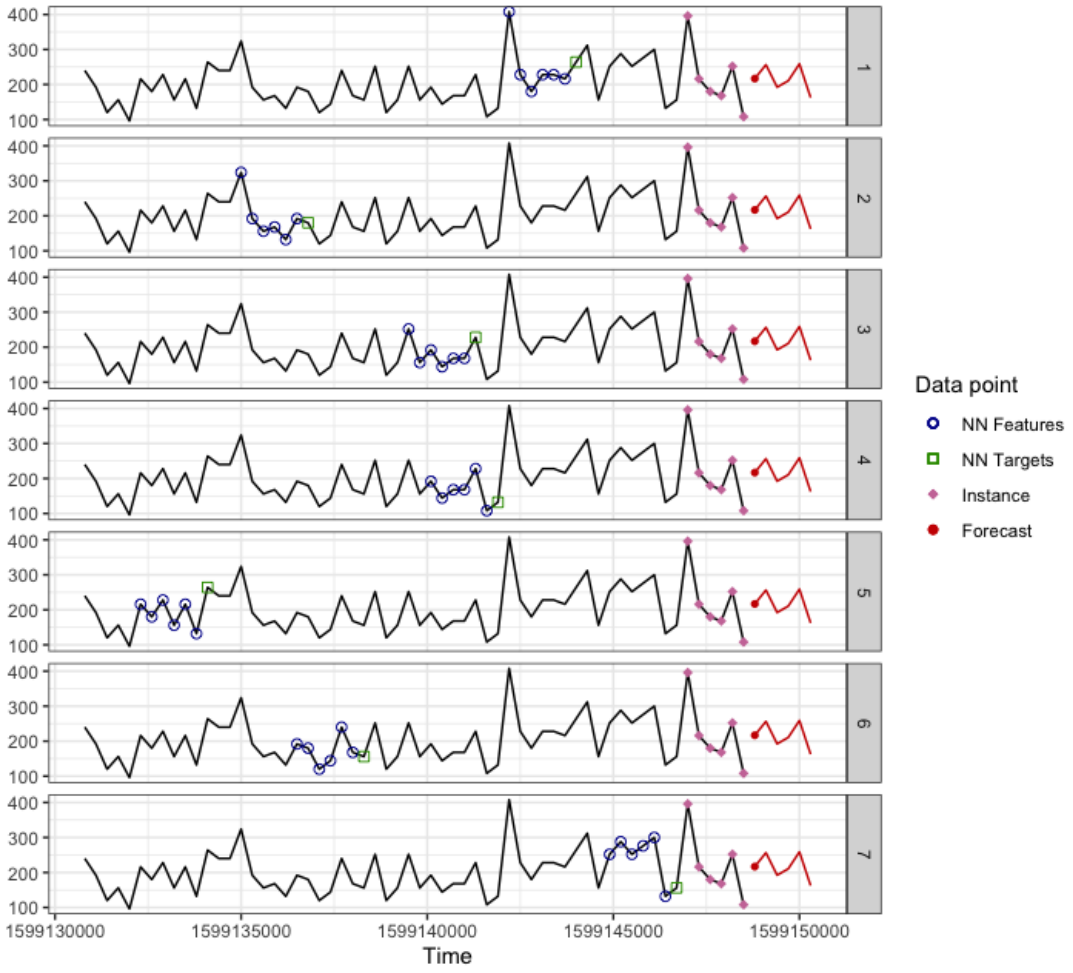
Obrázek 4.7: Model Prophet

4.8 K-nejbližší sousedé

K-nejbližší sousedé (anglicky K-Nearest Neighbors) je model strojového učení s učitelem (je potřeba model trénovat) používaný na klasifikaci a predikci dat. Tento model je vhodný pro krátkodobou predikci. [51]

Princip KNN spočívá v ukládání vzorků trénovací datové sady, kde každý vzorek obsahuje vektor vlastností popisující jej (*Features*) a přiřazenou hodnotu (*Target*). Pro nový vzorek najde KNN k nejbližších vzorků pomocí Eukleidovské vzdálenosti a predikuje novou hodnotu (*Forecast*) na základě agregace *Target* hodnot (většinou pomocí střední hodnoty). [51] [52].

Na Obrázku 4.8 je princip KNN znázorněn graficky. Na grafu je uveden příklad se zpožděním 6 kroků, s horizontem předpovědi 6 kroků a $k = 7$. Zpoždění udává, že výběr vzorku bude proveden na základě šesti posledních hodnot (v grafu označeny růžovým kosočtvercem *Instance*), pro vzorek tvořený z šesti posledních hodnot se vyhledá k nejbližších vzorků (v grafu *NN Features*, značeny modře). Predikovaná hodnota (*Forecast*) je pak průměrem *Target* hodnot (značeny zeleně) nejbližších vzorků. Proces se opakuje po dobu horizontu předpovědi s tím, že se z nově predikované hodnoty stane první hodnota *Instance* vzorku a poslední hodnota *Instance* je vyřazena.



Obrázek 4.8: Příklad principu KNN

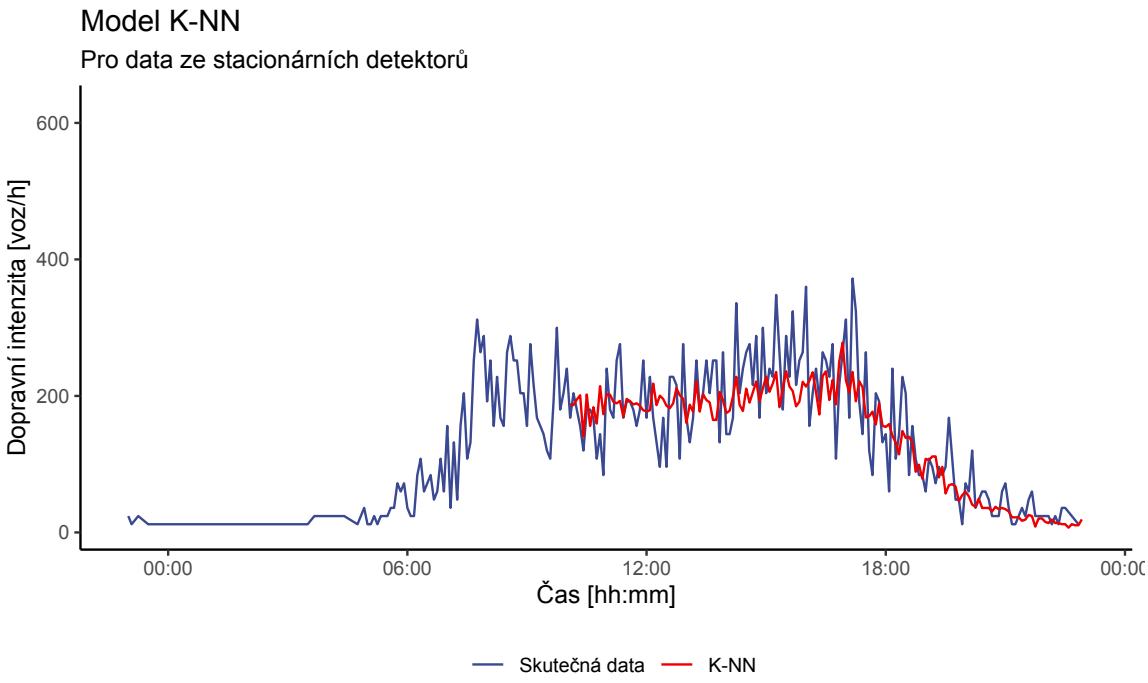
4.9 Použití modelu KNN na dostupná data

Model KNN pro krátkodobou predikci dostupných dat byl vytvořen v programovacím jazyce R. Narozdíl od předchozích modelů nebyl použit framework balíčků *tidymodels* a *modeltime*, protože model KNN nebyl těmito balíčky podporován.

Pro tvorbu modelu byl využit balíček *tsfknn* (Time series forecasting using KNN regression) [53], implementace tohoto modelu na dostupná data je v kódu v příloze.

4.9.1 Data ze strategických detektorů v Dobřichovicích

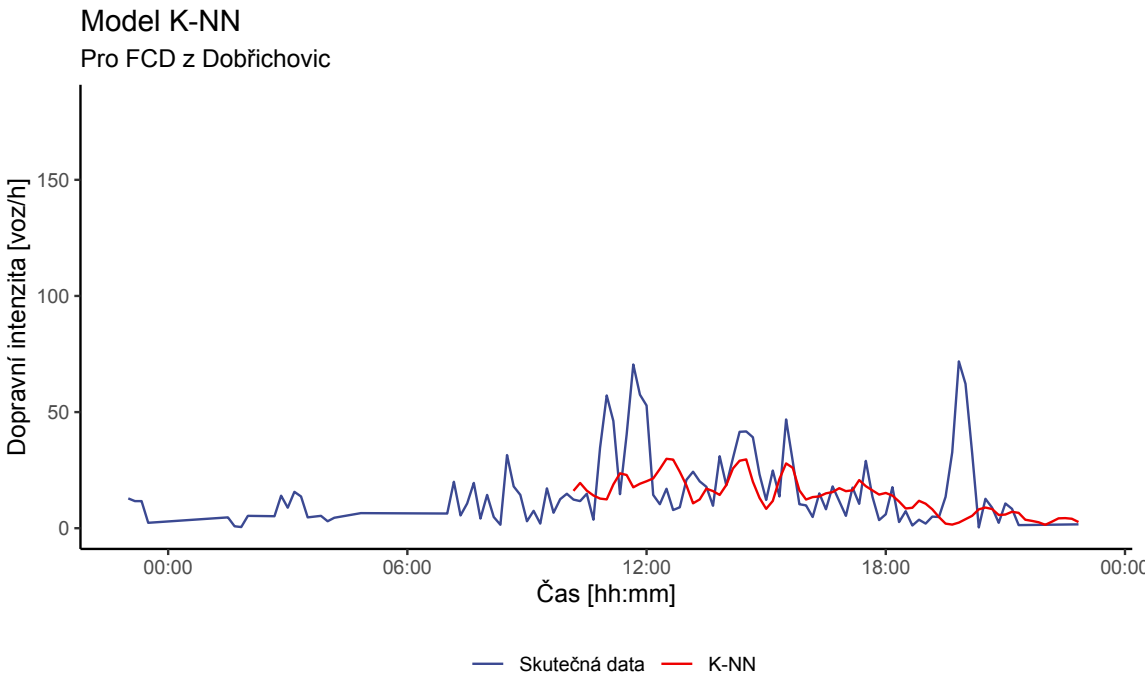
Na grafu na Obrázku 4.9 je vidět porovnání mezi skutečnými a predikovanými hodnotami intenzit. Z grafu je vidět že modelu se pro danou testovací sadu daří celkem přesně predikovat dopravní trend. Trénovaní tohoto modelu bylo díky jednoduchosti KNN velice rychlé a výpočetně nenáročné.



Obrázek 4.9: K-NN v Dobrichovicich

4.9.2 Data z FCD v Dobřichovicích

KNN pro data z FCD z Dobřichovic vypadá vizuálně hůře než pro data ze stacionárních dopravních senzorů. Model se několikrát neshoduje se skutečným trendem z FCD. Důvodem je nejspíše nízká kvalita dat, malá trénovací sada a velikost predikčního horizontu.



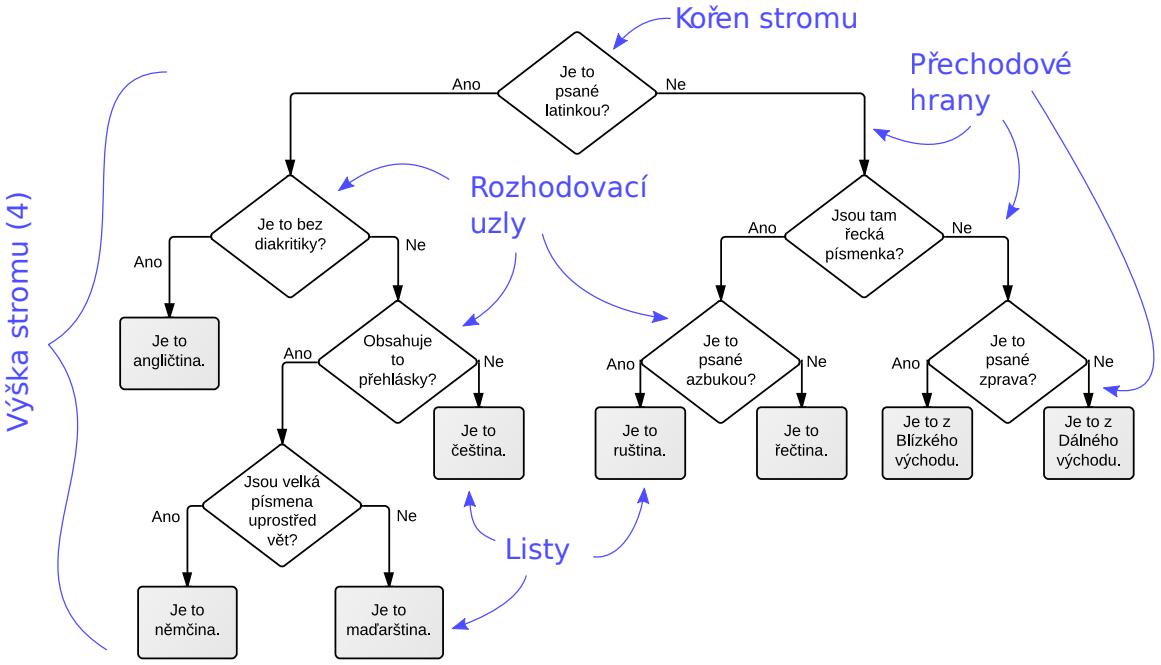
Obrázek 4.10: K-NN v Dobrichovicich

4.10 Random Forest

Ansámblové metody strojového učení jsou stavěny na principu kombinace mnoha jednoduchých modelů s cílem získání jednoho potenciálně velmi kvalitního modelu. Jednoduché modely, ze kterých se ansámblový model tvoří, se nazývají *weak learners*. V případě modelu Random Forest je za *weak learner* považován rozhodovací strom. [26]

4.10.1 Rozhodovací strom

Rozhodovací strom je neparametrická metoda strojového učení používaná pro klasifikaci a regresi. Princip této metody spočívá v tvorbě modelu, který předpovídá hodnotu vybrané proměnné pomocí učení jednoduchých rozhodovacích pravidel, odvozených z vlastností datové sady. [54] Jedná se o stromově strukturovaný model, jehož rozhodovací uzly představují vlastnosti datové sady, přechodové hrany představují rozhodovací pravidla a listy představují výsledek.



Obrázek 4.11: Příklad rozhodovacího stromu, převzato z [55]

Na Obrázku 4.11 je uveden příklad rozhodovacího stromu pro klasifikaci jazyka textu.

Pro konstrukci klasifikačního rozhodovacího stromu je nejprve nutné zvolit rozhodovací pravidlo pro kořen stromu. Rozhodovací pravidlo je určeno na základě míry čistoty. Míru čistoty lze stanovit pomocí *Gini impurity*, nebo *Entropie*. Zde je uveden příklad *Gini impurity*, tato míra měří jak často by byl náhodně vybraný prvek nesprávně zařazen, kdyby byl vybrán náhodně a nezávisle. Kořen stromu je zvolen tak, že je vypočteno *Gini impurity* pro všechny proměnné v datovém souboru a vybrána ta proměnná která má nejmenší

míru celkové *Gini impurity*. Matematický zápis *Gini impurity* vypadá následovně:

$$\text{Gini} = 1 - \sum_{i=1}^{K} p_i^2 \tag{4.8}$$

Kde p_i je pravděpodobnost zařazení prvku třídy i, pro K tříd. Celková *Gini impurity* je poté vypočtena jako vážený průměr *Gini impurity* listů rozhodovacího uzlu, kde váhy jsou dány počtem prvků zařazených do listu.

Po zvolení proměnné kořenu stromu jsou postupně identickým způsobem vybrány proměnné rozhodovacích uzlů, dokud není dosaženo zastavovacího kritéria. Zastavovací kritérium bývá definováno jako snížení *Gini impurity* na minimum, nebo dosažení předem definované maximální výšky stromu. Výsledný strom lze použít pro predikci nových dat zasazením nového pozorování do vytvořeného rozhodovacího stromu. [26]

Regresní strom

Doposud byly popsány rozhodovací stromy pro klasifikační úlohy (diskrétní data), rozhodovací stromy pro regresní úlohy (spojitá data) se nazývají regresní stromy. Konstrukce regresního stromu je velice podobná konstrukci klasifikačního rozhodovacího stromu. Hlavním rozdílem je že, místo hledání minimální míry *Gini impurity* je hledána minimální suma čtverců residuí (*RSS*)

V každém rozhodovacím uzlu u regresních stromů je podmínka určena na základě vybrané dělicí hodnoty (R_j) (např. vybraná proměnná > 10, kde dělicí hodnota je 10). Tato dělicí hodnota je určena algoritmem, který pro všechny R_j vypočte *RSS* a vybere tu, která je minimální. [26]

$$\text{RSS} = \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j}) \tag{4.9}$$

4.10.2 Random Forest

Samostatné rozhodovací stromy fungují velice dobře na trénovacích datech, ale často mají problémy správně klasifikovat nová pozorování. Tento problém je řešen pomocí metody Random Forest.

Pro tvorbu Random forest je nejprve vytvořenou potřebné množství bootstrapovaných datových sad. Bootstrapping je metoda resamplingu (opakovaného vzorkování), kdy je bootstrapovaná sada vytvořena náhodným výběrem s opakováním. Pro každou bootstrapovanou datovou sadu je poté vytvořen rozhodovací strom. Při tvorbě rozhodovacího stromu používáme pouze náhodnou podmnožinu proměnných při každém kroku tvorby stromu.

Výsledný model Random forest vznikne agregací všech rozhodovacích stromů, které

jsou jeho součástí. Tato metoda se nazývá *bagging* (Bootstrap + agregace). Výsledný model lze nadále optimalizovat pomocí optimalizace hyperparametrů (počet stromů, výška stromu, velikost náhodné podmnožiny proměnných). [56]

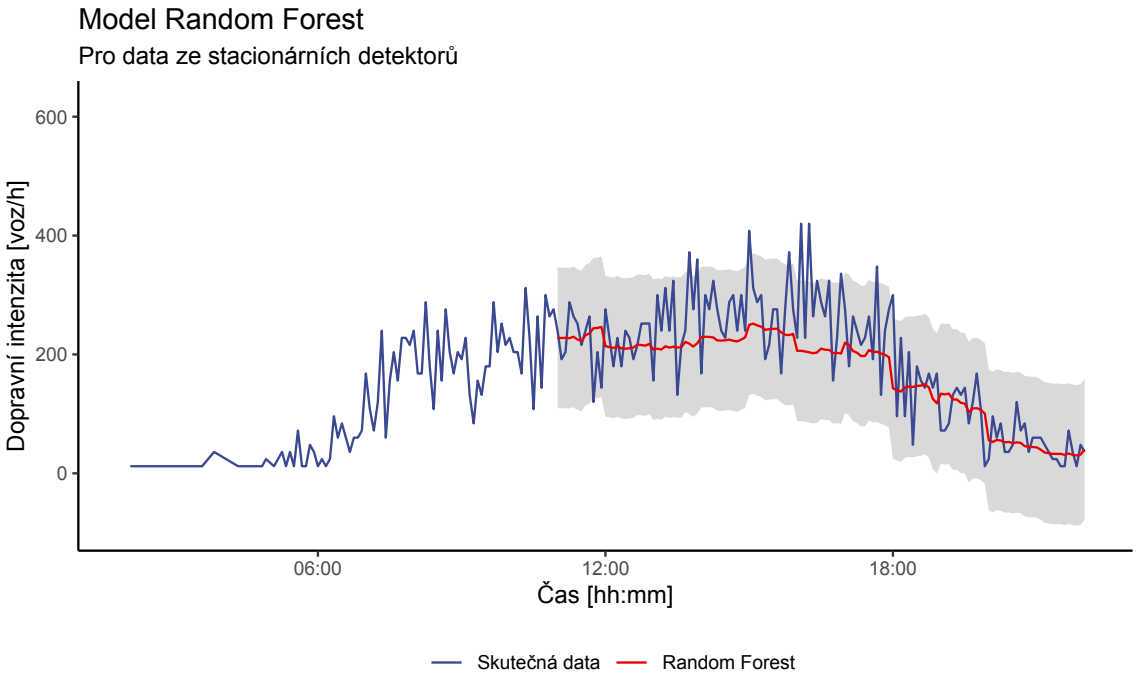
4.11 Použití modelu Random Forest

Model Random Forest je použit analogicky jako předchozí metody, s jedním zásadním rozdílem. Aby byl tento model schopen kvalitně predikovat budoucí stav dopravy je nutné rozdělit proměnnou datum měření na více komponent. Komponenty data měření jsou například: den v týdnu, den v měsíci, hodina dne, apod..

Pro tvorbu modelu byl využit balíček *randomForest* [57] volně dostupný pro programovací jazyk R.

4.11.1 Data ze stacionárních detektorů v Dobřichovicích

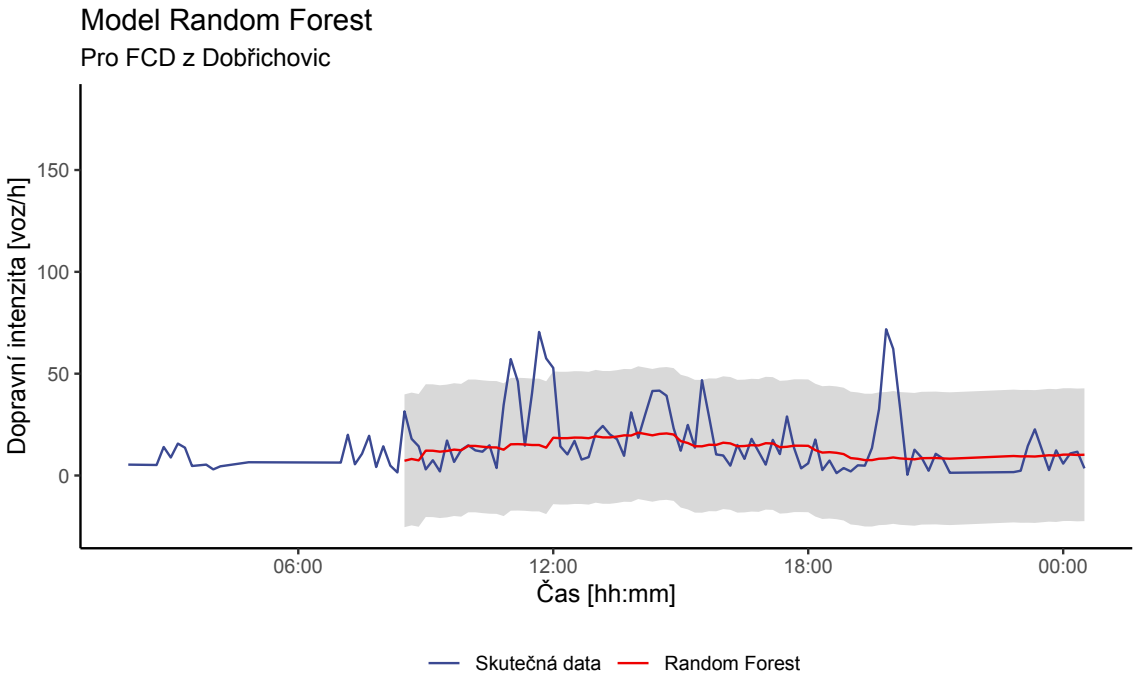
Na Obrázku 4.12 je vidět výstup modelu Random Forest v porovnání se skutečnými daty. Predikční křivka se v průběhu času skokově mění a dosahuje schodovitého vzhledu, což je způsobeno samotným principem regresních stromů, ze kterých je model složený.



Obrázek 4.12: Model Random Forest

4.11.2 Data z FCD v Dobřichovicích

Pro FCD se modelu Random Forest dle Obrázku 4.13 celkem přesně podařilo určit trend na základě dostupných dat. S výjimkou pár náhlých nárůstů intenzity ve skutečných datech.



Obrázek 4.13: Model Random Forest

4.12 XGBoosting

Velice populární metodou strojového učení se v posledních letech stal eXtreme Gradinet Boosting (XGBoost). [58] Jedná se o ansámblovou metodu pro řešení regrese a klasifikace. XGBoost je implementací Gradient Boosted Decision Trees alogritmu. Zahrnuje iterativní přidávání jednodušších predikčních modelů (nazývané weak learners, v případě XGBoost se jedná o rozhodovací stromy) do ansámbl modelu [59][60]

Boosting

Boosting je je ansámblová metoda používající kombinaci *weak learners* ke tvorbě modelu, podobně jako u metody Random forest. Narozdíl od Random forest, boosting výsledný model vytváří sekvenčně, kdy se každý *weak learner* snaží opravit chybu aktuálního modelu. Pro tvorbu výsledného modelu se používá jeden ze dvou algoritmů, adaptive boosting (adaboost), nebo gradient boosting. [56] Pro model XGBoost se používá algoritmus gradient boosting.

Gradient descent

Pro popis algoritmu gradient boosting je vhodné nejprve popsat optimalizační algoritmus gradient descent. Gradient descent hledá minimum *Loss* funkce, což je funkce, která kvantifikuje nepřesnosti mezi skutečnými a predikovanými hodnotami při trénování modelu (např. suma čtverců residuí). Algoritmus gradient descent postupuje iterativně po následujících krocích:

1. Výpočet gradientu *Loss* funkce (parciální derivace každého parametru)
2. Výběr náhodných hodnot pro parametry
3. Dosazení hodnot parametru do gradientu *Loss* funkce
4. Výpočet přírůstku
5. Výpočet nových parametrů
6. Kroky 3-5 se opakují dokud není dosažen maximální počet iterací, nebo velikost kroku je menší než zvolená tolerance

Výpočet přírůstků probíhá podle vzorce:

$$\hat{p}_t = -\eta \nabla \mathcal{L}(\hat{\theta}_{t-1}) \tag{4.10}$$

Kde $\hat{p}_t$ je nový vypočtený přírůstek v iteraci t, η je míra učení (*learning rate*) a $\nabla \mathcal{L}(\hat{\theta}^{t-1})$ je gradient *Loss* funkce s dosazenými parametry z předchozí iterace. *Learning rate* rozhoduje jak velký bude krok ve směru záporného gradientu (směrem k minimu). Velikost *Learning rate* určuje jak rychle algoritmus konverguje k optimálnímu řešení, pokud je *Learning rate* moc malý může se algoritmus zaseknout v lokálním minimu. Naopak pokud je *Learning rate* moc velký, může algoritmus přestřelit optimální řešení a nekonvergovat.

Po výpočtu přírůstků následuje výpočet nových parametrů. Výpočet nových parametrů vznikne součtem všech předchozích přírůstků.

$$\hat{\theta}_t = \sum_{i=0}^{t-1} \hat{p}_i \tag{4.11}$$

Kde $\hat{\theta}_t$ je nově vypočtený parametr a p_i jsou přírůstky z předchozích iterací. Pro jeden parametr m lze výpočet nového parametru zapsat jako:

$$\hat{\theta}_t^{(m)} = \hat{\theta}_{t-1}^{(m)} - \eta \frac{\partial \mathcal{L}}{\partial \hat{\theta}_{t-1}^{(m)}} \tag{4.12}$$

Správná volba *Learning rate* je pro algoritmus zásadní. Běžně se nejprve zvolí relativně velká hodnota, která se postupně s každou iterací zmenšuje. [61] [62]

Gradient boosting

Gradient boosting je algoritmus podobný algoritmu gradient descent s tím že, místo optimalizace parametrů optimalizujeme funkce představující *weak learners*. Oba optimalizační algoritmy hledají minimum *Loss* funkce, gradient descent hledá minimum výpočtem nových parametrů, kdežto gradient boosting hledá minimum přidáváním nových modelů (*weak learners*).

Pro vstupní data $\{x_i, y_i\}_{i=1}^n$ a diferencovatelnou *Loss* funkci $\mathcal{L}(y_i, F(x))$ algoritmus gradient boosting, ve kterém je *weak learner* regresní strom, probíhá iterativně po následujících krocích. [63]

1. Inicializace modelu s konstantní hodnotou

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} \mathcal{L}(y_i, \gamma)$$

2. Cyklus vytvářející regresní stromy od $m = 1$ do M

   2.1 Výpočet pseudo-residuí

$$r_{im} = -\Big[\frac{\partial \mathcal{L}(y_i, F(x_i))}{\partial F(x_i)}\Big]_{F(x)=F_{m-1}(x)}, \text{ pro } i = 1, ..., n$$

   2.2 Natrénování regresního stromu s vlastnostmi x na pseudo-reziduích r a tvorba *terminal regions* (konečné listy regresního stromu) $R_{j,m}$, pro $j = 1, .., J_m$

   2.3 Pro každý list stromu $j = 1, .., J_m$ je vypočtena výstupní hodnota γ_{jm}

$$\gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{ij}} \mathcal{L}(y_i, F_{m-1}(x_i) + \gamma)$$

   2.4 Aktualizujeme model

$$F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$$

XGBoost

XGBoost upravuje algoritmus gradient boostingu tak, že místo optimalizace, která funguje jako gradient descent v prostoru funkcí používá aproximaci pomocí Taylorova polynomu druhého řádu (Newton-Raphson metoda). Toto není jediný rozdíl mezi XGBoost a gradient boosting, XGBoost používá několik dalších metod pro zpřesnění a zrychlení algoritmu. Zde je uvedeno pár vlastností, které XGBoost odlišují. [64]

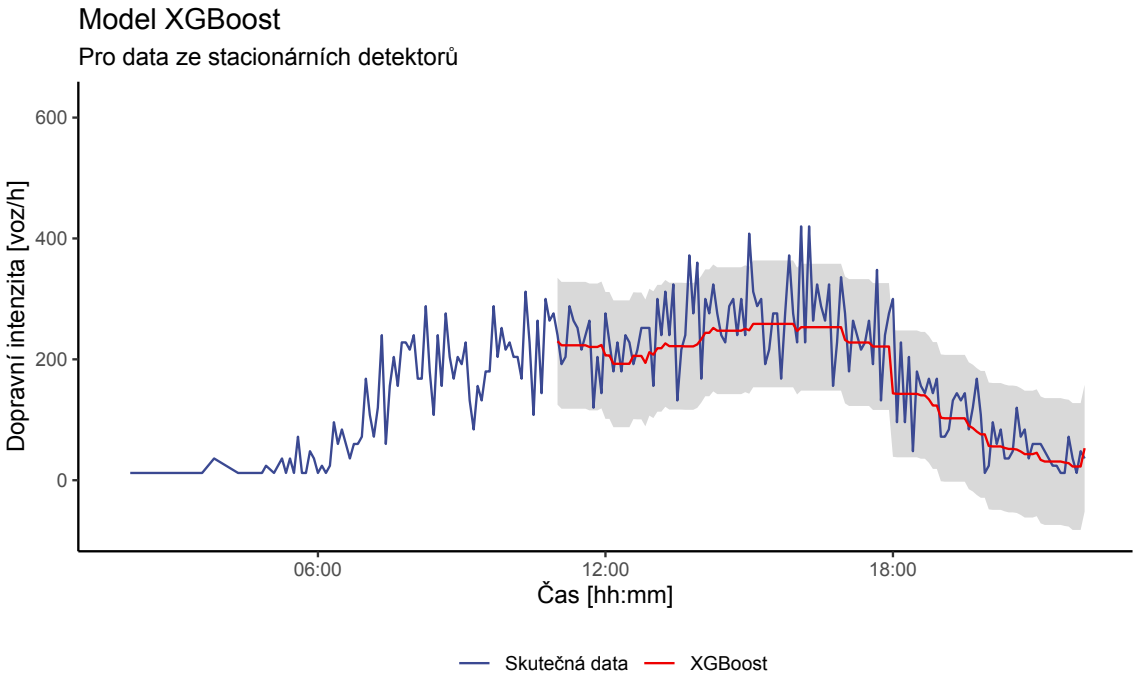
- Chytrá penalizace stromů
- Proporční zmenšování listů rozhodovacích stromů
- Newton Boosting
- Extra randomizační parametr

4.13 Použití modelů s XGBoost

Použití modelu XGBoost proběhlo velice podobně jako u modelu Random Forest. Pro tvorbu modelu byl použit oficiální XGBoost balíček [65] volně dostupný pro programovací jazyk R.

4.13.1 Data ze stacionárních detektorů v Dobřichovicích

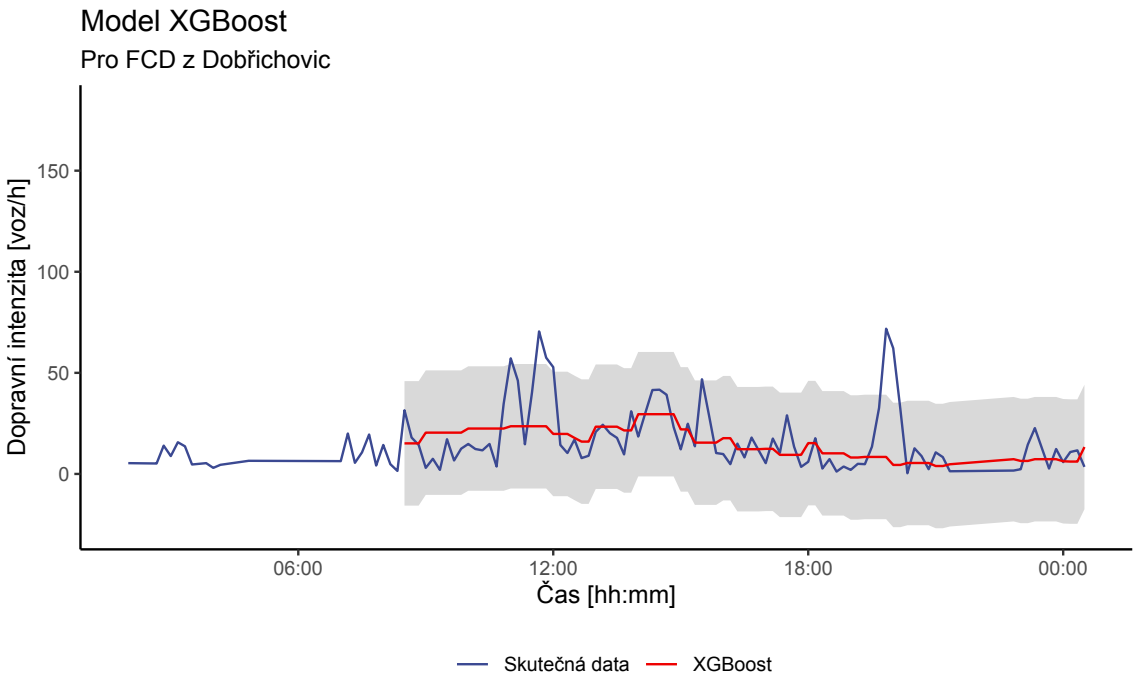
Predikční výstup modelu XGBoost na Obrázku 4.14 připomíná předchozí model Random Forest, což je způsobeno tím, že oba predikční modely používají regresní strom za *weak learner*. Tomuto modelu se podařilo dopravní model odhadnout velmi dobře. V predikcích je jasně vidět polední sedlo a odpolední špička následovaná večerním poklesem intenzity.



Obrázek 4.14: Model XGBoost

4.13.2 Data z FCD v Dobřichovicích

Na Obrázku 4.15 je znázorněna predikční křivka modelu XGBoost v porovnání se skutečnými daty. Predikční křivka se drží skutečných hodnot velice přesně, až na pár vyjímek.



Obrázek 4.15: Model XGBoost

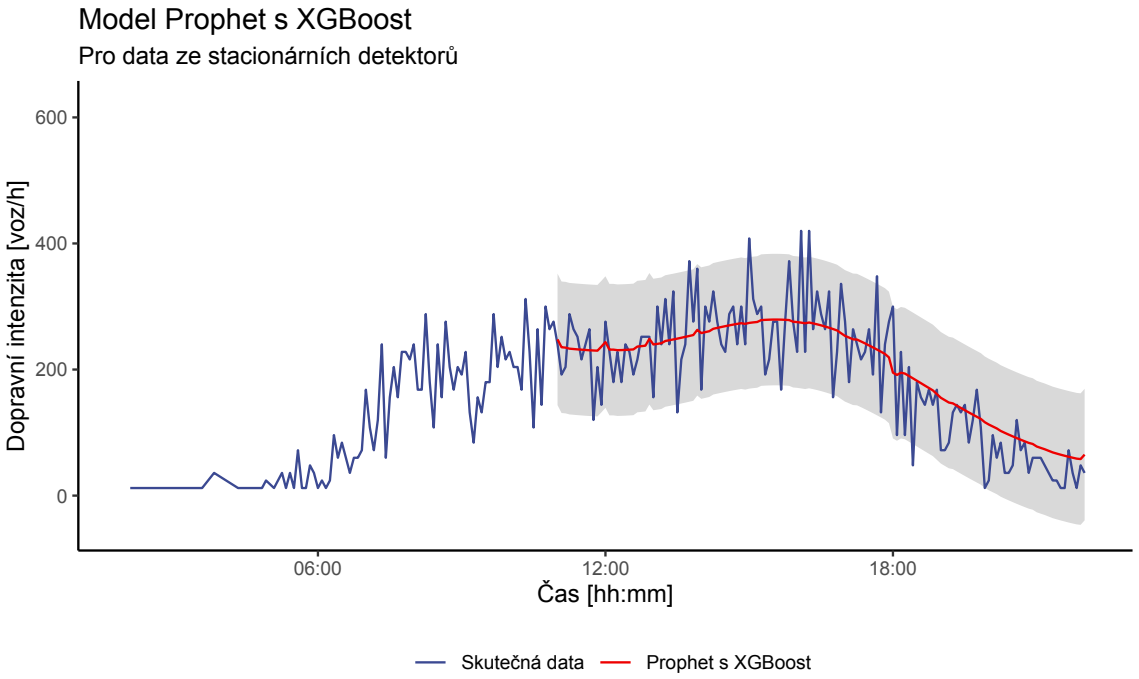
4.14 Hybridní modely s XGBoost

Statistické parametrické modely jako ARIMA a Prophet lze optimalizovat pomocí XGBoost. Hybridní model využije XGBoost pro modelování reziduí parametrického modelu.

Nejprve je vytvořen parametrický model (ARIMA, nebo Prophet), následně jsou vypočteny rezidua pro parametrický model. Rezidua společně s komponenty časové řady jsou následně modelována pomocí XGBoost. Výsledné predikce jsou vypočteny jako součet predikce parametrického modelu a predikce rezidua.

4.14.1 Data ze stacionárních detektorů v Dobřichovicích

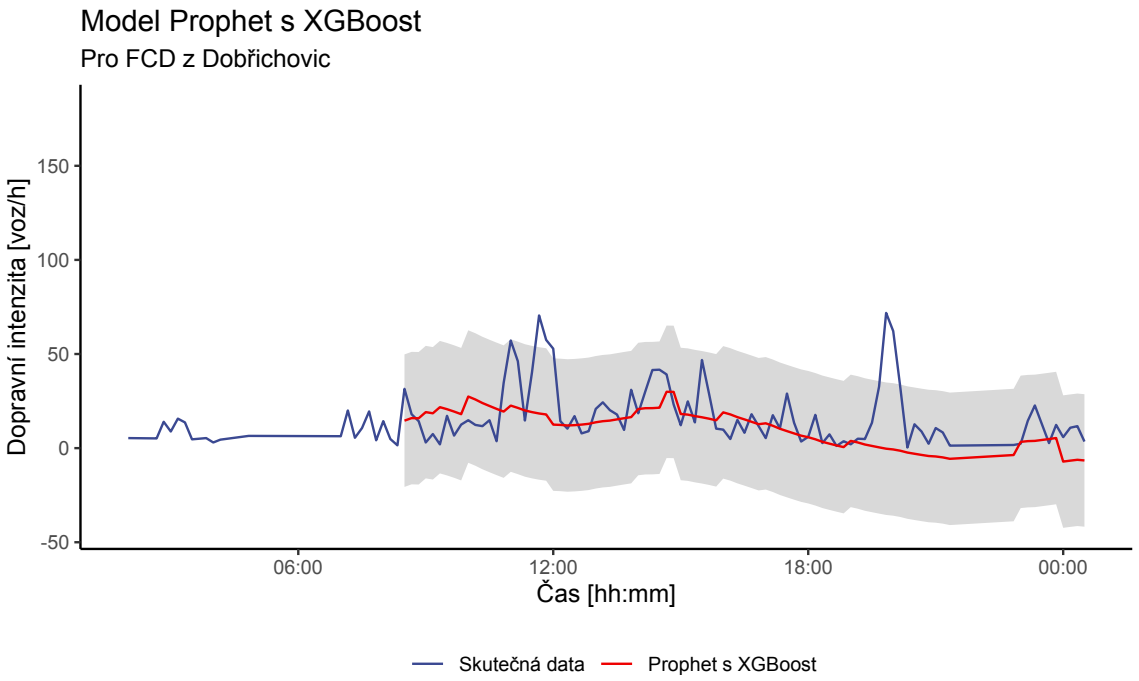
Kombinace modelu XGBoost a Prophet pro predikci dat ze stacionárních detektorů je znázorněna na Obrázku 4.16. Predikční křivka se moc neliší od predikcí samotného modelu Prophet (Obr. 4.6). Pouze v pár místech je vidět působení modelu XGBoost skokovými změnami.



Obrázek 4.16: Model Prophet s XGBoost

4.14.2 Data z FCD v Dobřichovicích

Predikce modelu Prophet s XGBoost znázorněny na obrázku 4.17 pro datovou sadu FCD se výrazně liší od samotného modelu Prophet. Je zde vidět výrazný vliv modelu XGBoost a výrazné zlepšení oproti samotnému modelu Prophet (Obr. 4.7).



Obrázek 4.17: Model Prophet s XGBoost

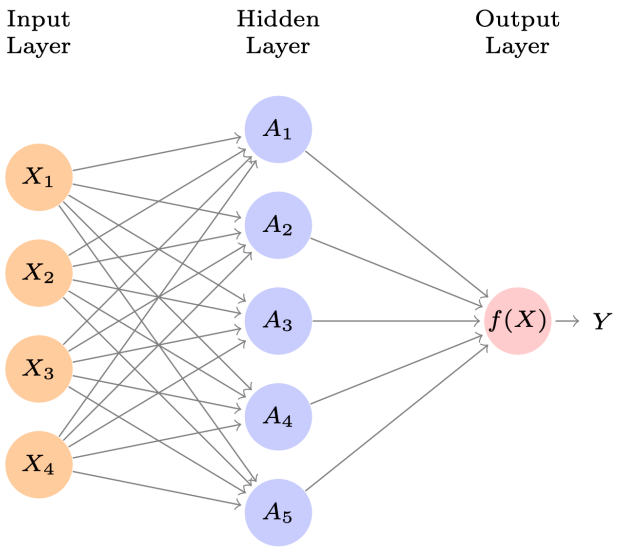
4.15 Neuronové sítě

Neuronové sítě obecně jsou metodou hlubokého učení založenou na matematickém modelu, který vzdáleně připomíná model neuronů v mozku. Modely napodobují způsob, kterým si biologické neurony předávají signály. Skládají se z neuronů uspořádaných ve vrstvách. Cílem neuronové sítě je na základě vstupního vektoru proměnných

$$X = (X_1, X_2, ..., X_p) \tag{4.13}$$

vytvořit nelineární funkci $f(X)$, která bude predikovat výstup Y.

Vrstvy neuronových sítí se třídí na vstupní vrstvu, skryté vrstvy a výstupní vrstvu. Prvky (neurony) vstupní vrstvy jsou tvořeny ze složek vstupního vektoru proměnných X. Ze vstupní vrstvy poté vedou vazby do skryté vrstvy (může být jedna, nebo více než jedna) a výstupy ze skrytě vrstvy vedou do výstupní vrstvy. [38] [26]



Obrázek 4.18: Neuronová síť, převzato z [26]

Na Obrázku 4.18 je uveden příklad dopředné (feed-forward) neuronové sítě s jednou skrytou vrstvou. Ve vstupní vrstvě jsou 4 prediktory $X_1, X_2, ..., X_4$, z každého prvku poté vede vazba do každého prvku skryté vrstvy. Ve skryté vrstvě je K prvků (K je volitelné, v tomto příkladu $K = 5$). Pro každý prvek skryté vrstvy jsou spočteny aktivace $A_K = h_K(X)$, kde funkce $h_K()$ je vytvořena tréninkem sítě. Výstupní vrstvu lze poté zapsat jako lineární model:

$$f(X) = \beta_0 + \sum_{K=1}^{K} \beta_K A_K \tag{4.14}$$

Parametry $\beta_0, ..., \beta_K$ jsou odhadnuty z dat. [38] [26] Tento příklad slouží pouze jako ukázka

principu jednoduché jednovrstvé dopředné neuronové sítě, reálné neuronové sítě mohou obsahovat více skrytých vrstev s více neurony a s jiným uspořádáním neuronů.

Existují různé specializace neuronových sítí pro specifické problémy. Například konvoluční neuronové sítě jsou často využívány pro rozpoznávání obrazu. [66] Pro zpracování sekvenčních dat, jako například časové řady, text a data ze senzorů se využívají rekurentní neuronové sítě. Tato třída neuronových sítí je navržena právě na zpracování sekvenčních dat. [67]

4.15.1 Autoregrese s pomocí neuronové sítě

Pro tvorbu autoregresního modelu časové řady za pomoci neuronových sítí byl použit model Neural network autoregression (NNAR). Tento model využívá dopřednou neuronovou síť s jednou skrytou vrstvou a opožděnými vstupy pro předpověď jednorozměrné časové řady. Obecně lze tento model zapsat:

$$\mathrm{NNAR}(p, P, k)_m \tag{4.15}$$

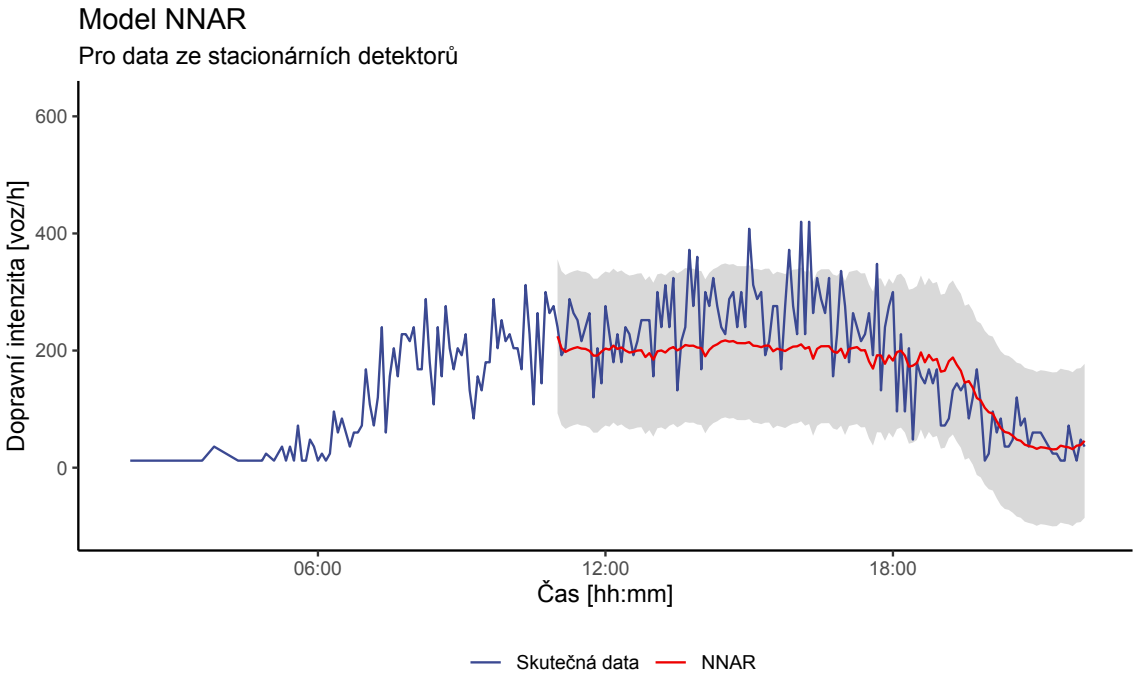
Kde p je dimenze ne-sezonního zpoždění, P je dimenze sezónního zpoždění, k je počet uzlů (neuronů) ve skryté vrstvě a m je počet pozorování za sezónu.

4.16 Použití neuronové sítě na dostupná data

Model autoregresní neuronové sítě byl vytvořen pomocí modelu *nnetar* z balíčku *modeltime*, který je volně dostupný pro programovací jazyk R. Model *nnetar* automaticky zvolí optimální hodnoty p, P, k a m na základě vstupních dat a poté se model natrénuje.

4.16.1 Data ze stacionárních detektorů v Dobřichovicích

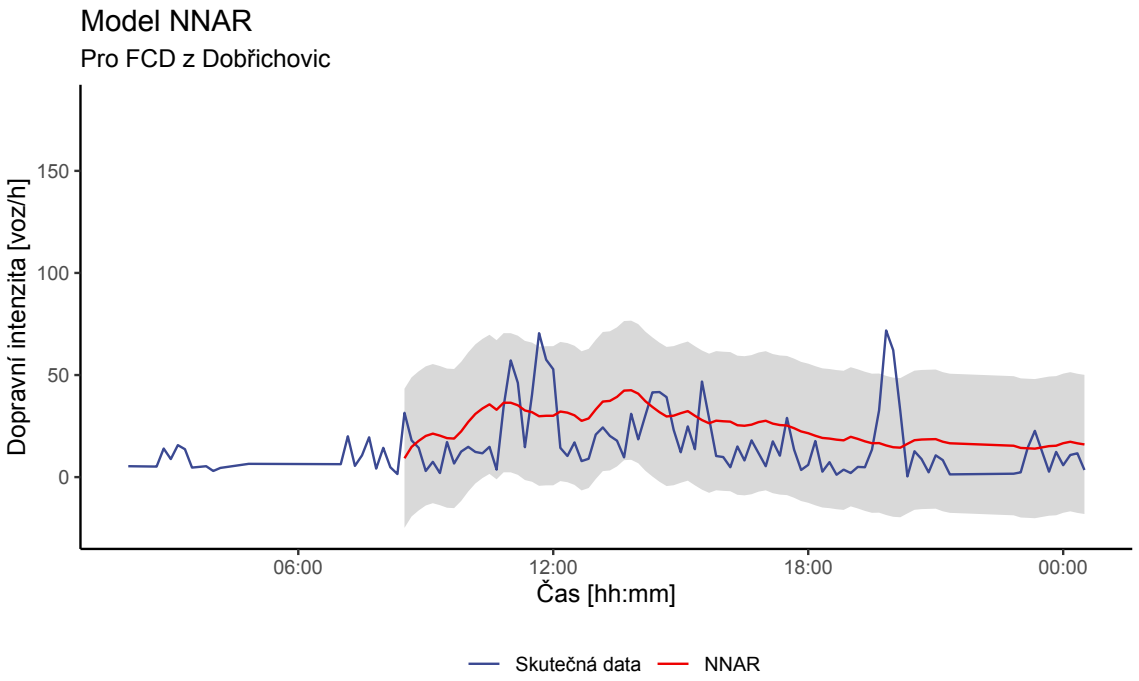
Na Obrázku 4.19 jsou zobrazeny výsledky modelu NNAR. Modelu NNAR se nepodařilo předpovědět odpolední dopravní špičku, pouze večerní pokles.



Obrázek 4.19: Model NNAR

4.16.2 Data z FCD v Dobřichovicích

Na Obrázku 4.20 jsou zobrazeny výsledky modelu NNAR pro datovou sadu FCD. Model v porovnání s testovacími daty vyprodukoval křivku, která je více hladká, ale její hodnoty většinou překračují skutečné naměřené hodnoty.



Obrázek 4.20: Model NNAR

Kapitola 5

Porovnání přesnosti použitých predikčních modelů

V této kapitole jsou nejprve obecně popsány metriky sloužící pro určení přesnosti predikčních modelů. Poté budou porovnány přesnosti modelů na dostupných datech.

5.1 Obecný popis metrik přesnosti

Aby bylo možné určit přesnost predikčního modelu a případně jej porovnat s jinými predikčními modely je nutné definovat obecné metriky určující přesnost modelu.

5.1.1 Střední absolutní chyba

Střední absolutní chyba (anglicky Mean absolute error) je metrika popisující chybu mezi skutečnou a predikovanou hodnotou. MAE je vypočtena jako suma absolutních rozdílů mezi skutečnou a predikovanou hodnotou podělena počtem vzorků:

$$\mathrm{MAE} = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t| \tag{5.1}$$

MAE lze jednoduše interpretovat, protože má stejné jednotky jako zkoumaná proměnná. Při evaluaci modelu je cílem aby tato hodnota byla co nejnižší.

5.1.2 Střední absolutní procentuální chyba

Střední absolutní procentuální chyba (Mean absolute percentage error) popisuje chybu predikce v procentech.

$$\mathrm{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right| \tag{5.2}$$

5.1.3 Střední absolutní škálovaná chyba

Střední absolutní škálovaná chyba (anglicky Mean absolute scaled error) je metrika chyby určená pro popis přesnosti modelu. Jedná se o podíl MAE a MAE naivního modelu. Naivní model je takový model, který přiřazuje aktuální předpovědi výstup z předchozího kroku. [68] Vyjádřeno matematicky kde m je délka kroku zpoždění:

$$\mathrm{MAE}_{naive} = \frac{1}{n-m}\sum_{t=1}^{n}|y_t - \hat{y}_{t-m}| \tag{5.3}$$

Poté lze MASE vyjádřit matematicky jako:

$$\mathrm{MASE} = \frac{\mathrm{MAE}}{\mathrm{MAE}_{naive}} \tag{5.4}$$

MASE ukazuje kvalitu predikčního modelu vůči naivnímu modelu. Pokud je hodnota MASE větší než 1, znamená to, že výsledky predikčního modelu vychází hůře, než výsledky modelu naivního. [68]

5.1.4 Symetrická střední absolutní procentuální chyba

Symetrická střední absolutní procentuální chyba (anglicky Symmetric mean absolute percentage error), narozdíl od MAPE normalizuje relativní chyby podílem skutečných i predikovaných hodnot. [69] Matematicky vyjádřeno:

$$\mathrm{SMAPE} = \frac{1}{n}\sum_{t=1}^{n}\frac{|y_t - \hat{y}_t|}{|y_t| + |\hat{y}_t|} \tag{5.5}$$

5.1.5 Střední kvadratická chyba

Střední kvadratická chyba (anglicky Root mean square error) lze chápat jako směrodatnou odchylku, ale místo odmocniny rozptylu je vypočtena odmocnina průměru čtverce reziduí (reziduál je vzdálenost mezi skutečnou a predikovanou hodnotou). Matematicky vyjádřeno:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \tag{5.6}$$

Hodnota RMSE udává průměrně jak moc jsou skutečné hodnoty vzdáleny od predikovaných. Druhá mocnina dává větší váhu větším chybám, proto je RMSE užitečné v případech, kdy jsou velké chyby obzvláště nežádoucí.

5.1.6 Koeficient determinace

Koeficient determinace známý také jako *R-kvadrát* (anglicky *R-squared*, nebo R^2),je statistická míra, která vyjadřuje podíl rozptylu závislé proměnné, který je vysvětlen nezávislými proměnnými v regresním modelu. Matematicky lze koeficient determinace zapsat jako:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{5.7}$$

Kde SS_{res} je suma čtverců residuí a SS_{tot} je suma čtverců odchylek predikované hodnoty od střední hodnoty vzorku. Matematický zápis lze poté rozepsat:

$$R^2 = 1 - \frac{\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}{\sum_{t=1}^{n}(y_t - \bar{y})^2} \tag{5.8}$$

Hodnoty *R-squared* se pohybují od 0 do 1, kde nízké hodnoty značí slabou korelaci mezi skutečnou a predikovanou hodnotou a vysoké hodnoty značí, že rozptyl predikovaných hodnot je podobný rozptylu skutečných hodnot. [70] [47]

5.2 Metodika křížové validace

Validace je klíčovým procesem při tvorbě jakéhokoli predikčního modelu. Umožňuje popis kvality modelu pomocí metrik přesnosti. Základním konceptem validace je rozdělení dostupné datové sady na trénovací a testovací část. Model je natrénován na testovací sadě dat a poté je jeho kvalita ověřena na testovací sadě dat.

Nejjednodušší metodou validace je prosté rozdělení celé datové sady dle určeného poměru (často je dán poměr: 80% trénovací a 20% testovací). Tento způsob je nazýván *Hold-out* validace. Je vhodný v počátečních fázích tvorby modelu, kdy chceme rychle a jednoduše zobrazit kvalitu modelu. Jedná se o výpočetně méně náročnou metodu, vhodnou u větších datových souborů. [71]

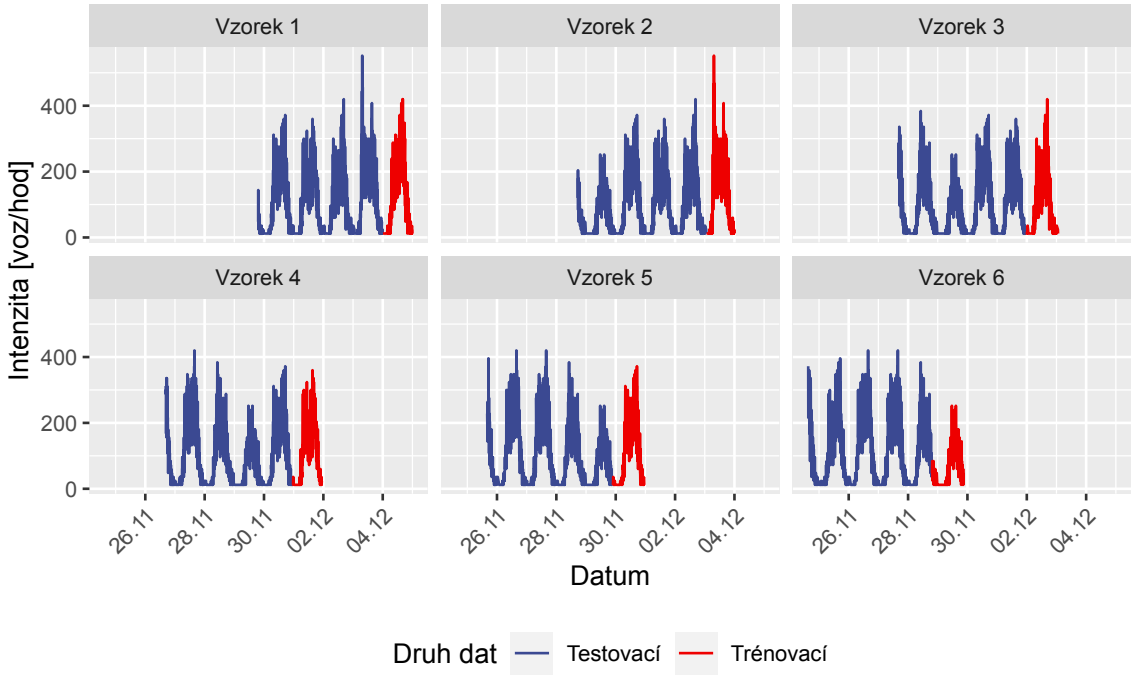
Vizuálním příkladem *Hold-out* jsou grafy použití predikčních modelů v kapitole 4, například na Obrázku 4.4 jsou červeně ukázány predikce modelu ARIMA v porovnání s testovací sadou dat (skutečnými hodnotami dopravní intenzity).

Více robustním přístupem k hodnocení modelů je křížová validace (Cross-validation). Principem křížové validace je rozdělení celé datová sady na několik vzorků, kde každý vzorek má testovací a trénovací část. Výslednou metrikou přesnosti je poté průměr metrik přesností jednotlivých vzorků. Existuje několik metod křížové validace, s tím, že různé metody vzorkují datovou sadu různým způsobem. Pro křížovou validaci časové řady je nutné postupovat tak, aby byla zohledněna časová závislost dat. [72]

Křížová validace předkládaných modelů je provedena následujícím způsobem. Datová sada je nejprve rozdělena na n vzorků o stejné délce. Každý vzorek je rozdělen na testovací

a trénovací sadu ve stejném poměru. Model je poté trénován a testován n-krát a výsledné metriky přesnosti jsou získány jako průměr individuálních metrik přesností jednotlivých vzorků.

Rozdělení vzorků pro křížovou validaci je znázorněno na příkladu na Obrázku 5.1.



Obrázek 5.1: Ukázka principu tvorby vzorků pro křížovou validaci

Na Obrázku 5.1 je uveden příklad tvorby vzorků pro křížovou validaci. Je vybráno $n = 6$ vzorků o délce 5 dní a vzdáleností mezi vzorky jeden den (Vzorek 2 začíná o jeden den dříve než Vzorek 1 atd.). Každý vzorek je rozdělen na trénovací sadu o délce 4 dny a testovací sadu o délce 1 den.

5.3 Porovnání přesnosti modelů

Pro porovnání použitých modelů jsou použity metriky přesnosti popsány v kapitole 5.1 a metoda křížové validace popsána v kapitole 5.2. Modely jsou porovnávány zvlášť na datech ze stacionárních detektorů v Dobřichovicích a datech z FCD v Dobřichovicích. Na základě výsledků metrik přesnosti budou modely zhodnoceny v Kapitole 6.

5.3.1 Stacionární detektory v Dobřichovicích

Nejprve jsou v Tabulce 5.1 ukázány výsledky metrik přesnosti pro modely natrénované na datech ze stacionárního detektoru bez křížové validace. Tyto výsledky patří k modelům,

které byly graficky znázorněny v Kapitole 6. Modely byly trénovány na třech měsících dat a otestovány na šestnácti hodinách.

Model	MAE	MAPE	MASE	SMAPE	RMSE	R^2
ARIMA	52.34	44.02	0.92	43.12	69.40	0.66
Prophet	41.56	47.12	0.73	31.29	56.20	0.78
sNaive	55.89	44.42	0.98	46.09	72.99	0.65
K-NN	39.42	31.77	0.72	30.12	51.45	0.68
Random Forest	43.90	40.46	0.77	31.70	60.24	0.75
XGBoost	40.10	41.32	0.70	31.86	53.44	0.79
Prophet s XGBoost	40.59	65.59	0.71	34.71	53.13	0.79
NNAR	50.30	46.79	0.88	35.44	66.96	0.70

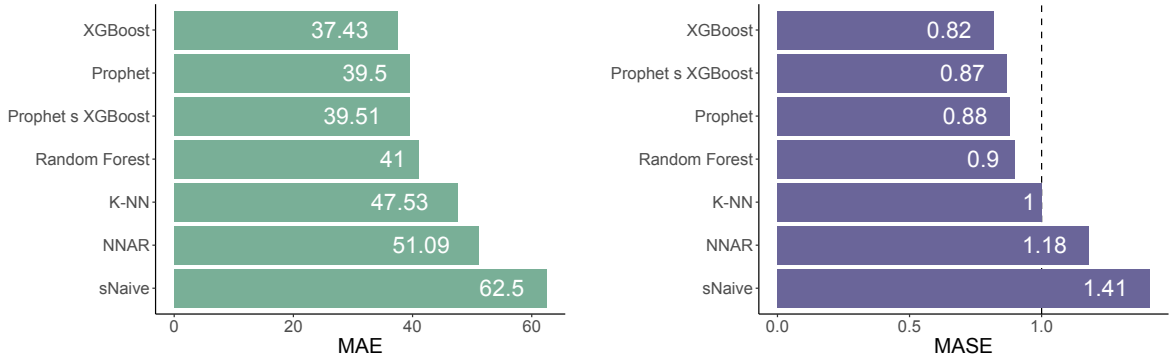
Tabulka 5.1: Porovnání modelů pro strategické detektory v Dobřichovicích

Následná Tabulka 5.2 ukazuje křížově validované metriky přesnosti pro modely natrénované na datech ze stacionárního detektoru. Dvouměsíční datová sada byla rozdělena na 7 vzorků, kde každý vzorek má délku tři měsíce mínus sedm dni, časová mezera mezi vzorky je jeden den a na testování byl zvolen celý jeden den.

Model	MAE	MAPE	MASE	SMAPE	RMSE	R^2
Prophet	39.50	62.57	0.88	44.70	53.94	0.68
sNaive	62.50	81.30	1.41	55.30	87.57	0.40
K-NN	47.53	65.55	1.00	40.40	62.57	0.56
Random Forest	41.00	67.97	0.90	41.52	55.35	0.68
XGBoost	37.43	50.00	0.82	37.17	52.09	0.69
Prophet s XGBoost	39.51	61.48	0.87	42.46	53.50	0.70
NNAR	51.09	85.48	1.18	47.81	68.88	0.54

Tabulka 5.2: Resampling

Pro přehledné porovnání predikčních modelů je uveden Obrázek 5.2, ve kterém jsou zobrazeny výsledné metriky MAE a SMAPE.



(a) MAE (b) MASE

Obrázek 5.2: Grafické znázornění vybraných metrik z Tabulky 5.2

5.3.2 FCD v Dobřichovicích

Tabulka 5.3 popisuje metriky přesnosti natrénované na tříměsíční datové sadě a otestované na šestnácti hodinách jednoho dne. Toto rozmezí je dáno, tak aby simulovalo reálnou situaci, kdy je na základě historických dat vytvořena predikce pro budoucí den. Dalším důvodem je, nízká kvalita dat. Tato tabulka ukazuje přesnosti modelů, znázorněných na grafech v Kapitole 4. Vzhledem k velkému rozptylu, který je vidět v grafech, pro získání lepší představy o přesnosti jednotlivých modelů je nutné pro tuto datovou sadu provést křížovou validaci.

Model	MAE	MAPE	MASE	SMAPE	RMSE	R^2
ARIMA	16.31	355.32	1.52	83.65	18.42	0.00
Prophet	15.10	171.80	1.41	98.91	20.07	0.03
sNaive	18.44	214.68	1.72	87.54	24.64	0.01
K-NN	12.28	125.73	1.06	68.48	18.45	0.03
Random Forest	10.61	129.17	0.99	64.64	16.33	0.07
XGBoost	10.68	122.06	0.99	66.71	15.66	0.12
Prophet s XGBoost	12.24	109.38	1.14	86.74	17.91	0.08
NNAR	14.81	258.66	1.38	79.44	17.39	0.08

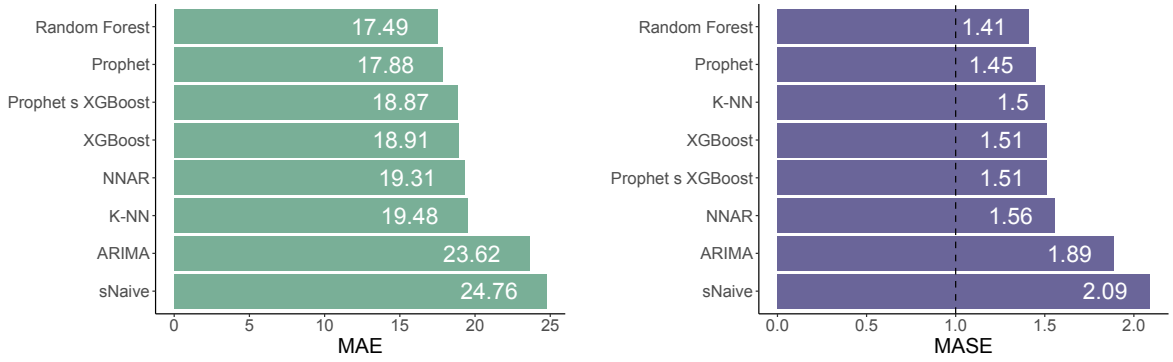
Tabulka 5.3: Porovnání modelů pro FCD v Dobřichovicích, bez cv

V Tabulce 5.4 jsou výsledky metriky přesnosti křížově validovaných modelů pomocí metodiky popsány v kapitole 5.2. Pro křížovou validaci bylo zvoleno sedm vzorků o délce 80 dní, s jedním dnem braným jako testovací a časovým posunem jeden den mezi vzorky.

Model	MAE	MAPE	MASE	SMAPE	RMSE	R^2
ARIMA	23.62	333.74	1.89	81.47	29.32	0.10
Prophet	17.88	232.56	1.45	70.23	23.39	0.39
sNaive	24.76	213.08	2.09	83.64	35.26	0.10
K-NN	19.48	181.77	1.50	66.03	27.33	0.28
Random Forest	17.49	196.04	1.41	61.89	23.76	0.32
XGBoost	18.91	259.06	1.51	66.32	24.46	0.35
Prophet s XGBoost	18.87	228.45	1.51	70.20	24.67	0.37
NNAR	19.31	241.05	1.56	69.65	25.48	0.27

Tabulka 5.4: Porovnání modelů pro FCD v Dobřichovicích pomocí křížové validace

Výsledné metriky MAE a SMAPE jsou graficky znázorněny v grafech na Obrázku 5.3.



(a) MAE (b) MASE

Obrázek 5.3: Grafické znázornění vybraných metrik z Tabulky 5.4

Kapitola 6

Zhodnocení výsledků a vyvození obecných možnosti využití predikčních modelů

V této kapitole jsou nejprve zhodnoceny jednotlivé predikční modely na základě metrik přesnosti. Poté jsou vybrány nejvhodnější modely pro data ze stacionárních detektorů a pro data z plovoucích vozidel.

Poté jsou vyvozeny obecné možnosti využití predikčních modelů v dopravě. Je navrženo pro jaké predikční sady a horizonty predikce je vhodné zvolit jaký model.

6.1 Zhodnocení modelů individuálně

V této sekci jsou popsány výsledky jednotlivých použitých predikčních modelů individuálně.

Model ARIMA

Přesnost modelu ARIMA je lepší než použití prostého naivního modelu. Nicméně přesnost modelu nesplnila očekávání vyplývající z popularity používání tohoto modelu pro predikci intenzity dopravy. Výsledky šestnáctihodinové predikce natrénované na datech ze stacionárního detektoru měly podobný trend jako testovací data, ale střední absolutní chyba je v porovnání s ostatními modely výrazná.

Při křížové validaci modelu ARIMA na datech ze stacionárního senzoru se vyskytl problém s funkcí *auto.arima()*. Tato funkce naráží na problém nadměrně dlouhé trénovací doby při větším množství dat s větším počtem sezón (pro zvolená data je jedna sezóna jeden den). Tento problém je umocněn křížovou validací, která model trénuje několikrát na různých datech, a neoptimalizovaností správy alokace paměti u některých funkcí v

programovacím jazyce R. Křížová validace na kompletní datové sadě nebyla úspěšná a nedobrala se výsledku.

Výsledky modelu ARIMA natrénovaného na FCD dopadly podstatně hůře, jak pro šestnáctihodinovou, tak i pro denní křížově validovanou predikci. Trend modelu na grafu v Obrázku 4.5 vůbec neodpovídá testovacím datům, připomíná spíše konstantní střední hodnotu. Tento výsledek je způsoben nejednoznačností trendů v FCD.

Model Prophet

Model Prophet je parametrický model, o kterém se dá říci, že konkuruje modelu ARIMA. Model obstál velice dobře s dobrými výsledky z šestnáctihodinové predikce i z denní křížové validace nad daty ze stacionárního detektoru. Nízká hodnota MAE drží krok i s komplikovanými modely strojového učení. Jediný nedostatek tohoto modelu je nízká schopnost poradit si s odlehlými hodnotami v datech, což potvrzuje vyšší hodnota MAPE (tato metrika je více ovlivnitelná velkými chybami). Výsledná predikční křivka je mnohem hladší v porovnání s ostatními modely.

Na FCD datech model obstál trochu lépe než model ARIMA, ale pořád hůře než model naivní. Při křížové validaci dosáhl v porovnání velice dobrých výsledků.

Model sNaive

Model sNaive je zdaleka nejjednodušší použitý predikční model, který lze považovat za referenční model podobně jako prostý naivní model využívaný metrikou přesnosti MASE. Výsledky tohoto modelu pro data ze stacionárního detektoru, byly tedy očekávaně nejhorší. Při šestnáctihodinové predikci se tento model udržel před prostým naivním modelem, ale při křížové validaci obsadil poslední místo ve všech metrikách kromě MAPE.

U FCD dat si model sNaive vedl ještě hůře, což je překvapující vzhledem k výsledkům MASE u všech modelů, které naznačují že pro tuto datovou sadu by byl naivní model vhodný. Znamená to, že pro danou datovou sadu je mnohem více důležitá předchozí naměřená hodnota, než jakékoli sezónní trendy.

Model K-NN

Model k-nejbližších sousedů (KNN) je prvním zástupcem neparametrických modelů strojového učení. Nad daty ze stacionárního detektoru měl tento model velice dobré výsledky při šestnáctihodinové predikci, ale při křížové validaci nad jedním testovacím dnem byly metriky přesnosti mnohem horší. To naznačuje využití tohoto modelu spíše pro rychlou (trénovaní modelu je výpočetně velice nenáročné) krátkodobou predikci v řádu několika hodin.

Výsledky pro FCD jsou podobné ostatním modelům s výjimkou modelů ARIMA a sNaive, které jsou horší

Model Random Forest

Random Forest je první zástupce ansámblových modelů strojového učení. Tento model dosáhl dobrých výsledků nad daty ze stacionárního detektoru. Výsledné metriky jsou srovnatelné s modelem Prophet, kde model Random Forest nemá takové problémy s odlehlými hodnotami, ale má trochu větší střední absolutní chybu. Křivka predikčních hodnot tohoto modelu (Obrázek 4.12) má narozdíl od předchozích modelů schodovitý průběh, to je způsobeno regresními stromy ze kterých se tento model skládá.

Model Random Forest si nejlépe poradil s datovou sadou FCD dat, ve srovnání s ostatními modely.

Model XGBoost

Druhý zástupce ansámblových modelů je XGBoost, který by teoreticky poskytovat lepší predikce než model Random Forest, díky jeho pokročilejší optimalizaci parametrů. Nad daty ze stacionárních detektorů má tento model opravdu nejlepší výsledky. Tento model má malou střední absolutní chybu a zároveň si dobře poradil s odlehlými hodnotami (má malé MAPE).

Trend predikční křivky se podobá trendu modelu Random Forest, což dává smysl, protože oba modely používají jako *weak learner* variaci regresního stromu. S tím že křivka modelu XGBoost se lépe drží skutečných hodnot.

S FCD daty si tento model poradil trochu hůře, než Random Forest, ale pořád docela dobře.

Model Prophet s XGBoost

Hybridní model Prophet s XGBoost, který modeluje jednorozměrnou řadu pomocí modelu Prophet a poté vykoná regresi residuí modelu Prophet pomocí XGBoost, by teoreticky měl vylepšit hodnoty modelu Prophet a vykompenzovat nedostatky s odlehlými hodnotami. V praxi jsou výstupy hybridního modelu podobné s výstupy samotného modelu Prophet.

Model NNAR

Jediným zástupcem modelů umělé neuronové sítě je model NNAR. Výsledky tohoto modelu nejsou v porovnání moc dobré. Předpokládá se, že komplexnější model rekurentní neuronové sítě by měl lepší výsledky.

6.2 Zhodnocení modelů dle datové sady

Pro porovnání modelů byly dostupné dvě datové sady. První datová sada byla získána pomocí stacionárního mikrovlnného radaru. Data z této datové sady jsou velice kvalitní a vhodná pro porovnávání predikčních modelů.

Druhá datová sada byla získána metodou plovoucích vozidel (FCD). Tato datová sada má podstatně nižší kvalitu dat, což jí činí zajímavou pro posouzení predikčních modelů nad slabšími daty.

Datová sada ze stacionárního detektoru

Pro tuto sadu vychází dle křížové validace nejlépe model XGBoost, s tím ,že modely Prophet a Prophet s XGBoost dosahovaly podobných výsledků, jen o malinko horších.

Datová sada FCD

Pro datovou sadu FCD vychází nejlépe model Random Forest, ale i tento model má metriku přesnosti MASE přesahující 1, což značí, že má větší chyby než naivní model. Velká úspěšnost prostého naivního modelu na FCD datech je způsobena faktem, že v následných záznamech existuje vztah mezi počty vozidel.

Jedno vozidlo se může při průjezdu segmentem silniční komunikace vyskytnout ve více záznamech pokud průjezd segmentem trvá více než jednu minutu. Tento jev může být jedním s faktorů zvyšujícím úspěšnost prostého naivního modelu.

Přispívat k úspěšnosti naivního modelu může také malá míra penetrace plovoucích vozidel dopravního proudu.

6.3 Obecné možnosti využití predikčních modelů v dopravě

Celkově mají největší přesnost ansámblové modely XGBoost a Random Forest. Díky rozkladu data měření na více komponent jsou schopné korektně identifikovat týdenní trendy (víkendy a speciální dny), což jim zaručilo dobré výsledky při křížové validaci.

Predikční modely mohou sloužit jako doplňkový nástroj k některým dopravním senzorům, u kterých může dojít k výpadku dat. Při zpracování explorativní datové analýzy bylo nalezeno několik výpadků dat u stacionárních detektorů v Dobřichovicích. Tyto výpadky by mohly být nahrazeny predikcemi na základě historických dat.

V případě dat z plovoucích vozidel se může stát že na daném segmentu silniční komunikace v určitý moment nevyskytuje žádné plovoucí vozidlo. Tyto momenty lze doplnit

pomocí predikčních modelů. V případě FCD je vhodné mít větší množství dat (kvůli nízké penetraci). Pro FCD by bylo ideální odhadnout skutečné hodnoty dopravní intenzity pro naměřené počty plovoucích vozidel. Pro zpřesnění výsledků lze zde použité predikční modely kombinovat s dalšími datovými zdroji (například informace o počasí).

Kapitola 7

Závěr

Cílem předkládané diplomové práce bylo popsat, použít a posoudit zvolené predikční modely na dostupných dopravních datech z oblasti Dobřichovic. Dále popsat veličiny dopravního proudu, vysvětlit vztahy mezi nimi a uvést dopravní detektory, pomocí nichž se měří.

Základní charakteristiky dopravního proudu, jsou dopravní intenzita, hustota a rychlost dopravního proudu. Mezi těmito třemi veličinami existují vztahy, které jsou znázorněny pomocí fundamentálních grafů dopravy. K zjištění těchto veličin z dopravního proudu slouží dopravní detektory.

Dopravní detektory se dělí na intruzivní a neintruzivní. K intruzivním dopravním detektorům patří například indukční smyčka nebo pneumatický detektor. V dnešní době se čím dál více prosazují neintruzivní dopravní detektory, které svojí instalací nezasahují do vozovky. Za neintruzivní dopravní detektory jsou považovány například infračervené detektory nebo kamerové systémy.

Dalším způsobem sběru dat o dopravním proudu jsou data z plovoucích vozidel FCD. Data FCD jsou vytvářena z informací přenášených přímo z vozidel účastníků dopravního proudu. Tento druh dat se dělí na GFCD, CFCD a kooperativní FCD. GFCD využívají pro zjištění lokace vozidla GPS jednotku ve vozidle, CFCD využíva triangulaci mobilních telefonů pomocí základnových stanic a kooperativní FCD využívá technologie CITS.

V předkládané diplomové práci byla vyhodnocena data ze stacionárních mikrovlnných radarů a data z FCD v Dobřichovicích. Data FCD byla získána od Ředitelství silnic a dálnic. Tyto datové soubory byly popsány pomocí explorativní datové analýzy (EDA). EDA je proces analýzy, vizualizace a sumarizace dat. Pro následnou tvorbu predikčních modelů byl z obou datových souborů vybrán segment silniční komunikace mezi Dobřichovicemi a Černošicemi.

Predikční modely byly nejprve obecně popsány a na základě přehledu literatury byly vybrány tyto predikční modely: sNaive, ARIMA, Prophet, KNN, Random Forest, XG-

Boost, Prophet s XGBoost a NNAR. Všechny zvolené modely byly popsány a následně implementovány na oba datové soubory v programovacím jazyce R.

Výsledky predikčních modelů byly ověřeny pomocí metrik přesnosti, vypočtených pomocí křížové validace. Celkově dosáhly největší přesnosti ansámblové modely strojového učení, konkrétně model XGBoost a model Random Forest, který měl trochu horší výsledky pro data ze stacionárních senzorů. Dopravní trendy predikoval také dobře model Prophet, jak samotný tak v kombinaci s XGBoost.

Všechny použité predikční modely dosahovaly lepších výsledků na datech ze stacionárních detektorů. To bylo pravděpodobně způsobeno zatím malým podílem plovoucích vozidel v dopravním proudu. Se zvyšujícím se počtem plovoucích vozidel budou získána lepší data, která se stanou podkladem pro přesnější predikce.

Bibliografie

[1] URL: https://ocw.tudelft.nl/wp-content/uploads/Chapter-4.-Fundamental-diagrams.pdf.

[2] *Chapter 2, Traffic Detector Handbook: Third edition-volume I*, 2006. URL: https://www.fhwa.dot.gov/publications/research/operations/its/06108/02.cfm.

[3] H. Pavel, L. Martin, S. Dušan a V. Martin, "Zmapování služeb a dat v oblasti FCD (Floating Car Data) pro využití v rámci informačních systémů ŘSD", *ČVUT v Praze Fakulta dopravní, Ústav řídicí techniky a telematiky*, s. 96, 2010.

[4] J. Guerrero-Ibáñez, S. Zeadally a J. Contreras-Castillo, "Sensor Technologies for Intelligent Transportation Systems", *Sensors*, roč. 18, č. 4, s. 1212, 2018. DOI: 10.3390/s18041212.

[5] *What is a pneumatic tube counter? pneumatic tube counter meaning*, 2023. URL: https://www.isarsoft.com/knowledge-hub/pneumatic-tube-counter.

[6] A. Riid, J. Kaugerand, J. Ehala, M. Jaanus a J.-S. Preden, "An Application of a Low-Cost Microwave Radar to Traffic Monitoring", in *2018 16th Biennial Baltic Electronics Conference (BEC)*, 2018, s. 1–4. DOI: 10.1109/BEC.2018.8600962.

[7] 2014. URL: https://www.fhwa.dot.gov/policyinformation/pubs/vdstits2007/05pt2.cfm.

[8] G. Szwoch a J. Kotus, "Acoustic Detector of Road Vehicles Based on Sound Intensity", *Sensors*, roč. 21, č. 23, 2021, ISSN: 1424-8220. URL: https://www.mdpi.com/1424-8220/21/23/7781.

[9] D. Pfoser, "Floating Car Data", in *Encyclopedia of GIS*, S. Shekhar a H. Xiong, ed. Boston, MA: Springer US, 2008, s. 321–321, ISBN: 978-0-387-35973-1. DOI: 10.1007/978-0-387-35973-1_423. URL: https://doi.org/10.1007/978-0-387-35973-1_423.

[10] J. Lau, *Google maps 101: How ai helps predict traffic and determine routes*, 2020. URL: https://blog.google/products/maps/google-maps-101-how-ai-helps-predict-traffic-and-determine-routes/.

[11] J. Fedewa, *Where does google maps get its traffic data from?*, 2022. URL: https://www.howtogeek.com/842637/where-does-google-maps-get-its-traffic-data-from/.

[12] K. Thomas, H. Fouchal, S. Cormier a F. Rousseaux, "C-ITS Communications based on BLE Messages", in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, s. 1–7. DOI: 10.1109/GLOBECOM42002.2020.9322076.

[13] URL: https://www.its-knihovna.cz/cz/knihovna/projekty/c-roads/systemy-c-its/o-systemech-c-its.

[14] D. Budimir, N. Jelušić a M. Perić, "Floating Car Data Technology", *Pomorstvo*, roč. 33, s. 22–32, čvn. 2019. DOI: 10.31217/p.33.1.3.

[15] G. Aoude a K. Jeanbart, *Ai-enabled cameras and lidar can improve traffic today and support the AVS of Tomorrow*, 2022. URL: https://www.smartcitiesdive.com/news/opinion-ai-traffic-cameras-lidar-avs-derq/625494/.

[16] *What is exploratory data analysis?* URL: https://www.ibm.com/topics/exploratory-data-analysis.

[17] H. Tyagi, *Hitchhiker's Guide to Exploratory Data Analysis*, 2020. URL: https://towardsdatascience.com/hitchhikers-guide-to-exploratory-data-analysis-6e8d896d3f7e.

[18] H. WICKHAM, "Introduction", in *R for Data Science: Import, Tidy, transform, visualize, and model data*. O'REILLY MEDIA, 2023.

[19] *Quarto*. URL: https://quarto.org/.

[20] H. Wickham, "Tidy data", *The Journal of Statistical Software*, roč. 59, 10 2014. URL: http://www.jstatsoft.org/v59/i10/.

[21] C. O. Wilke, *Fundamentals of data visualization a primer on making informative and compelling figures*. O'Reilly, 2020.

[22] URL: https://www.czso.cz/documents/10180/142756350/1300722103.pdf/53ded62a-5c7c-45ba-b17f-ba60021e5c54?version=1.1.

[23] *NDIC - DATEX II Elaborated Data Publication - FCD*, 2023. URL: https://registr.dopravniinfo.cz/docs/x-format/cz-ndic_d2-fcd-v1.0-cs-html/concepts.html.

[24] F. Hrubý, "Návrh vyhodnocení kvality dopravy na základě dostupných FCD ve vybrané lokalitě", dis. pr., Czech Technical University in Prague, 2021.

[25] K. Irawan, R. Yusuf a A. S. Prihatmanto, "A Survey on Traffic Flow Prediction Methods", in *2020 6th International Conference on Interactive Digital Media (ICIDM)*, 2020, s. 1–4. DOI: 10.1109/ICIDM51048.2020.9339675.

[26] G. James, D. Witten, T. Hastie a R. Tibshirani, *An introduction to statistical learning: With applications in R*, 2. vyd. Springer, 2022.

[27] *Traffic prediction: How machine learning helps forecast congestions and plan optimal routes*, 2022. URL: https://www.altexsoft.com/blog/traffic-prediction/.

[28] S. Shalev-Shwartz a S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2022.

[29] D. Billings a J.-S. Yang, "Application of the ARIMA Models to Urban Roadway Travel Time Prediction - A Case Study", in *2006 IEEE International Conference on Systems, Man and Cybernetics*, sv. 3, 2006, s. 2529–2534. DOI: 10.1109/ICSMC.2006.385244.

[30] T. Alghamdi, K. Elgazzar, M. Bayoumi, T. Sharaf a S. Shah, "Forecasting Traffic Congestion Using ARIMA Modeling", in *2019 15th International Wireless Communications Mobile Computing Conference (IWCMC)*, 2019, s. 1227–1232. DOI: 10.1109/IWCMC.2019.8766698.

[31] H. Dong, L. Jia, X. Sun, C. Li a Y. Qin, "Road Traffic Flow Prediction with a Time-Oriented ARIMA Model", in *2009 Fifth International Joint Conference on INC, IMS and IDC*, 2009, s. 1649–1652. DOI: 10.1109/NCM.2009.224.

[32] G. Jain a R. R. Prasad, "Machine learning, Prophet and XGBoost algorithm: Analysis of Traffic Forecasting in Telecom Networks with time series data", in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2020, s. 893–897. DOI: 10.1109/ICRITO48877.2020.9197864.

[33] L. Zhang, Q. Liu, W. Yang, N. Wei a D. Dong, "An Improved K-nearest Neighbor Model for Short-term Traffic Flow Prediction", *Procedia - Social and Behavioral Sciences*, roč. 96, s. 653–662, 2013, Intelligent and Integrated Sustainable Multimodal Transportation Systems Proceedings from the 13th COTA International Conference of Transportation Professionals (CICTP2013), ISSN: 1877-0428. DOI: https://doi.org/10.1016/j.sbspro.2013.08.076. URL: https://www.sciencedirect.com/science/article/pii/S1877042813022027.

[34] G. Leshem a Y. Ritov, "Traffic flow prediction using adaboost algorithm with random forests as a weak learner", *International Journal of Mathematical and Computational Sciences*, roč. 1, č. 1, s. 1–6, 2007.

[35] X. Dong, T. Lei, S. Jin a Z. Hou, "Short-Term Traffic Flow Prediction Based on XGBoost", in *2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)*, 2018, s. 854–859. DOI: 10.1109/DDCLS.2018.8516114.

[36] R. Fu, Z. Zhang a L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction", in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 2016, s. 324–328. DOI: 10.1109/YAC.2016.7804912.

[37] D. Wang, Y. Meng, S. Chen, C. Xie a Z. Liu, "A Hybrid Model for Vessel Traffic Flow Prediction Based on Wavelet and Prophet", *Journal of Marine Science and Engineering*, roč. 9, č. 11, 2021, ISSN: 2077-1312. DOI: 10.3390/jmse9111231. URL: https://www.mdpi.com/2077-1312/9/11/1231.

[38] R. J. Hyndman a G. Athanasopoulos, *Forecasting: Principles and practice*. OTexts, 2021.

[39] J. Dancker, *A brief introduction to time series forecasting using statistical methods*, 2023. URL: https://towardsdatascience.com/a-brief-introduction-to-time-series-forecasting-using-statistical-methods-d4ec849658c3.

[40] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2022. URL: https://www.R-project.org/.

[41] M. Kuhn a H. Wickham, *Tidymodels: a collection of packages for modeling and machine learning using tidyverse principles.* 2020. URL: https://www.tidymodels.org.

[42] M. Dancho, *modeltime: The Tidymodels Extension for Time Series Modeling*, R package version 1.2.6, 2023. URL: https://CRAN.R-project.org/package=modeltime.

[43] H. Wickham, M. Averick, J. Bryan et al., "Welcome to the tidyverse", *Journal of Open Source Software*, roč. 4, č. 43, s. 1686, 2019. DOI: 10.21105/joss.01686.

[44] G. Grolemund a H. Wickham, "Dates and Times Made Easy with lubridate", *Journal of Statistical Software*, roč. 40, č. 3, s. 1–25, 2011. URL: https://www.jstatsoft.org/v40/i03/.

[45] M. Dancho a D. Vaughan, *timetk: A Tool Kit for Working with Time Series*, R package version 2.8.3, 2023. URL: https://CRAN.R-project.org/package=timetk.

[46] J. Brownlee, *How to remove trends and seasonality with a difference transform in Python*, 2020. URL: https://machinelearningmastery.com/remove-trends-seasonality-difference-transform-python/.

[47] R. Khan, *ARIMA model for forecasting – Example in R*, 2017. URL: https://rpubs.com/riazakhan94/arima_with_example.

[48] E. Howell, *How to forecast time-series using autoregression*, 2023. URL: https://towardsdatascience.com/how-to-forecast-time-series-using-autoregression-1d45db71683.

[49] S. J. Taylor a B. Letham, "Forecasting at scale", *The American Statistician*, roč. 72, č. 1, 37–45, 2017. DOI: 10.1080/00031305.2017.1380080.

[50] S. Taylor a B. Letham, *prophet: Automatic Forecasting Procedure*, R package version 1.0, 2021. URL: https://CRAN.R-project.org/package=prophet.

[51] M. P. Frias, F. Martinez, F. Charte a A. J. Rivera, *Time Series Forecasting with KNN in R: the tsfknn Package*, 2023. URL: https://cran.r-project.org/web/packages/tsfknn/vignettes/tsfknn.html.

[52] S. Disci, *Time series forecasting: KNN vs. ARIMA*, 2020. URL: https://www.r-bloggers.com/2020/09/time-series-forecasting-knn-vs-arima/.

[53] F. Martinez, M. P. Frias, F. Charte a A. J. Rivera, "Time Series Forecasting with KNN in R: the tsfknn Package", *The R Journal*, roč. 11, č. 2, s. 229–242, 2019.

[54] *1.10. Decision Trees*. URL: https://scikit-learn.org/stable/modules/tree.html.

[55] D. Lessner, M. Lána, M. Podrázká Tomková a J. Haut, *Rozhodovací stromy a chytré otázky*, 2020. URL: https://popelka.ms.mff.cuni.cz/~lessner/mw/index.php/Uebnice/Informace/Rozhodovac_stromy_a_chytr_otzky.

[56] J. Rocca, *Ensemble methods: Bagging, boosting and stacking*, 2021. URL: https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205.

[57] A. Liaw a M. Wiener, "Classification and Regression by randomForest", *R News*, roč. 2, č. 3, s. 18–22, 2002. URL: https://CRAN.R-project.org/doc/Rnews/.

[58] R. Nana, *Kaggle competitions: Why choose keras as framework?: Data Science and Machine Learning*, 2020. URL: https://www.kaggle.com/getting-started/151903.

[59] *XGBoost documentation*. URL: https://xgboost.readthedocs.io/en/stable/.

[60] D. Becker, *XGBoost*, 2018. URL: https://www.kaggle.com/code/dansbecker/xgboost.

[61] R. Kwiatkowski, *Gradient descent algorithm-a deep dive*, 2022. URL: https://towardsdatascience.com/gradient-descent-algorithm-a-deep-dive-cf04e8115f21.

[62] N. Hug, *Understanding gradient boosting as a gradient descent*, 2019. URL: https://nicolas-hug.com/blog/gradient_boosting_descent#fn:2.

[63] J. H. Friedman, "Greedy function approximation: A gradient boosting machine.", *The Annals of Statistics*, roč. 29, č. 5, s. 1189 –1232, 2001. DOI: 10.1214/aos/1013203451. URL: https://doi.org/10.1214/aos/1013203451.

[64] R. Gandhi, *Gradient boosting and xgboost*, 2019. URL: https://medium.com/hackernoon/gradient-boosting-and-xgboost-90862daa6c77.

[65] T. Chen, T. He, M. Benesty et al., *xgboost: Extreme Gradient Boosting*, R package version 1.7.5.1, 2023. URL: https://CRAN.R-project.org/package=xgboost.

[66] M. Mishra, *Convolutional Neural Networks*, 2020. URL: https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939.

[67] F. Ducau, *Text generation with recurrent neural networks (rnns)*, 2020. URL: https://blog.paperspace.com/recurrent-neural-networks-part-1-2/.

[68] A. Ahuja, *Mean absolute scaled error (mase) in forecasting*, 2021. URL: https://medium.com/@ashishdce/mean-absolute-scaled-error-mase-in-forecasting-8f3aecc21968.

[69] D. Sarra, *How to interpret SMAPE just like MAPE*, 2022. URL: https://medium.com/@davide.sarra/how-to-interpret-smape-just-like-mape-bf799ba03bdc.

[70] S. Shabou, *Time Series Analysis with R*, 2020. URL: https://s-ai-f.github.io/Time-Series/index.html.

[71] E. Allibhai, *Holdout vs. cross-validation in machine learning*, 2018. URL: https://medium.com/@eijaz/holdout-vs-cross-validation-in-machine-learning-7637112d3f8f.

[72] E. Howell, *How to correctly perform cross-validation for Time Series*, 2023. URL: https://towardsdatascience.com/how-to-correctly-perform-cross-validation-for-time-series-b083b869e42c.