Czech Technical University in Prague

Faculty of Electrical Engineering

Department of Computer Science

Diploma thesis

Machine Learning Models for Car Accident Site Analysis

Adam Rak

Supervisor: Ing. Jan Drchal, Ph.D.

January 2018

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Computer Science

DIPLOMA THESIS AGREEMENT

Student: Rak Adam

Study programme: Open Informatics
Specialisation: Artificial Intelligence

Title of Diploma Thesis: Machine Learning Models for Car Accident Site Analysis

Guidelines:

1) Familiarize yourself with an available spatio-temporal dataset on car accidents (Jednotná dopravní vektorová mapa, Ministerstvo dopravy ČR, http://www.jdvm.cz/).
2) Examine available research results on car accident analysis. Focus on machine learning approaches. Also study machine learning methods dealing with imbalanced datasets.
3) Develop, implement and evaluate machine learning models of car accident sites. Model attributes such as accident severity or site hazard. Focus on a selection and design of appropriate features as well as on selection of machine learning techniques and their hyperparameters.
4) Enrich the dataset using external data such as OpenStreetMap to improve the models.

Bibliography/Sources:

[1] Chong, Miao, Ajith Abraham, and Marcin Paprzycki. "Traffic accident analysis using machine learning paradigms." Informatica 29.1 (2005).

[2] Krishnaveni, S., and M. Hemalatha. "A perspective analysis of traffic accident using data mining techniques." International Journal of Computer Applications 23.7 (2011): 40-48.

[3] Chawla, Nitesh V. "Data mining for imbalanced datasets: An overview." Data mining and knowledge discovery handbook. Springer US, 2005. 853-867.

Diploma Thesis Supervisor: Ing. Jan Drchal, Ph.D.

Valid until the end of the winter semester of academic year 2018/2019

prof. Dr. Michal Pěchouček, MSc.
Head of Department

prof. Ing. Pavel Ripka,CSc.
Dean

Prague, June 21, 2017

Acknowledgments

First of all, I would like to thank my supervisor, Ing. Jan Drchal, Ph.D. for his advice, supervision, endless patience and time he dedicated me while creating this thesis. I would like to express my great gratitude to my family, friends and especially my parents, who supported me, advised me and helped me during whole study and preparation of this thesis.

Declaration

I hereby declare that I have completed this thesis independently and that I have listed all the literature and publications used within the research and work. I have no objections to usage of this work in compliance with Act No 121/2000 Sb. (the Copyright Act), as amended, and in compliance with copyright-related rights currently in force.

8. January 2018, Prague ....

Abstract

The aim of this work is to analyze traffic accidents using machine learning methods. Using accident records in the Czech Republic, machine learning methods are able to find patterns and determine key factors which are specific to the traffic accident. The focus is put on the identification of the severity of the accidents and identification of potentially hazardous sites. Accidents records are enriched with map data to improve the accuracy of identification. Obtaining information about factors responsible for traffic accidents can lead to a decrease in the number of traffic accidents.

Keywords traffic accidents, machine learning, accident severity, site hazard

Abstrakt

Cieľom tejto práce je analyzovať dopravné nehody metódami strojového učenia. Pomocou záznamov o dopravných nehodách v Českej republike dokážu metódy strojového učenia nájsť vzory a určovať kľúčové faktory, ktoré sú špecifické pre dopravné nehody. Pozornosť je postavená na identifikáciu vážnosti dopravných nehôd a určovanie potenciálne nebezpečných miest. Záznamy o dopravných nehodách sú doplnené informáciami z máp na zvyšovanie presnosti identifikácii. Získavanie informácii o faktoroch spôsobujúcich dopravné nehody môže viesť k znižovaniu počtu dopravných nehôd.

Kľúčové slová dopravné nehody, strojové učenie, vážnosť nehôd, nebezpečné miesta

Contents

Figures

Tables

1 Introduction

In the past years, traffic accidents have been one of the most common causes of injuries and deaths. Reducing the number of accidents has a huge value to the society. Modification of traffic policies has potential to reduce the number of accidents. Finding patterns that arise in event of the traffic accident can improve our understanding of factors that are responsible for the accidents and flaws in traffic policies. Previous studies started to analyze traffic accidents using machine learning methods.

This work proposes two approaches in traffic accidents analysis using machine learning methods. The goal of the first approach is to determine the severity of the accidents. The severity of the accident is identified by a combination of factors provided in accident record. Elevated factors are calculated to determine possible reasons that cause severe accidents. The second approach focuses on identification of sites with a higher frequency of accidents. This approach has potential to have world-wide usage in the identification of potentially dangerous places on traffic network. Identifying hazardous sites can lead to significant reduction of the number of accidents. Information about yet unknown factors can be helpful in designing new traffic networks or modification of the existing traffic network resulting in increase safety on the roads.

Accidents records were obtained from Unified Transport Vector Map (UTVM) from Ministry of Transport of the Czech Republic. Approximately 600 000 records contain various spatial, temporal and other specific information about the accident such as vehicle characteristics, driver characteristics, weather conditions, direction of impact, number of injured people and fatalities etc. Data from OpenStreetMap provide additional information about surroundings of the accidents. To preserve generality of hazardous sites identification, specific information from UTVM was not taken into account.

Both approaches were analyzed using well-known Random Forest and Gradient Boosting methods. Significant factors responsible for severe accidents or high frequency of the accidents are calculated from models. Features that frequently occur in trained decision trees have potential to be the key factors.

Next chapter of this thesis overviews some of the state of the art approaches in identifying key factors responsible for severe accidents and key factors responsible for the high frequency of accidents in certain locations. Chapter 3 is divided into three parts. First part overviews two presented approaches in analyzing traffic accidents (severity and site hazard) and their goals. The second part describes datasets used, selecting and preprocessing of potentially interesting features. Third part briefly defines machine learning methods and evaluation methods of the models. Chapter 4 describes a process of obtaining datasets and its preparation for learning and implementation of ensembled models. Lastly, in chapter 5 both approaches are experimentally evaluated. Performance of particular models is compared with respect to the various dataset selections and representations. Chapter 5 also contains discussion about achieved results and possible improvements of the traffic accidents analysis.

2 Traffic accidents analysis

In the past years, there is a huge growth in traffic in the world. Traffic accidents are one of the most common causes of injuries and deaths. For this reason, researchers began to analyze traffic behavior. Finding patterns in traffic behavior can significantly increase safety in the traffic network. Analyzing patterns in data of the traffic accidents can help to modify traffic policy to decrease number of accidents and the severity of accidents. Identifying factors that are responsible for injuries and fatalities can bring a huge value to the society. Previous researches focused on using machine learning methods to identify causes of accidents and provide valuable information for mitigating the number of accidents and its severity.[1]

2.1 Accident severity

Collecting information related to the traffic accidents can be useful in analysis the severities of the accidents and minimizing injuries and deaths caused by traffic accidents.[1] Researchers used various methods from statistical analysis, logistic regressions to machine learning to identify environmental, vehicle and driver factors, predict accidents, reduce the number of accidents and reduce the severity of the accidents.[1][2] The most popular machine learning methods used for traffic accidents analysis are neural networks, decision trees, support vector machines and ensemble methods. Studies show that neural networks, decision tress and ensemble methods provide the best results in accident analysis. [1][3] Severity analysis in [1] shows the Random Forest Classifier achieved better results than other methods (Naïve Bayes, J48, AdaBoostM1, PART).

According to [2][4][5], not using seatbelts, high speed, and driver-side impacts are the main factors causing fatalities. Also, vehicle type, alcohol, age and gender of the affected people play a significant role in the outcome of the accidents. Accidents involving smaller vehicles tend to be more severe. Older people and women tend to suffer more severe injuries in traffic accidents. Accident severity correlates with the combination of factors rather than a single factor.[2] On the other hand, the study [6] states that a single variable like driving speed or light conditions can single-handedly have a huge impact to the number of injuries in the traffic accident. Studies [2][3] found that weather conditions and time of the accidents were not a significant factor in accidents severity.

Studies [7] [8] outline correlation between injuries and types of regions. According to [7], urban areas tend to be more hazardous than rural areas. Also, fatal injuries were more frequent in residential areas. This is apparent due to different density of the population in urban and rural areas.

In a study [9] accidents records were separated according to their severity into five categories: no injury, possible injury, non-incapacitating injury, incapacitating injury and fatal injury. Four different models were used to classify accidents: artificial neural network using hybrid learning, decision trees, support vector machines and hybrid decision trees. For identification non-incapacitating injuries, incapacitating injuries and fatal injuries hybrid decision trees outperformed other three methods. The model performed better on representation classes as fatal accidents class and non-fatal accidents class.

In a study [3], accidents had been labeled with two categories: injury and property damage. In this case, road width, shape of the vehicle or speed appeared to hold more information than weather conditions. Neural networks and decision trees were used. Trained classifiers were subsequently ensembled to improve the results of the classification. Also, the study proposes a method of pre-clustering dataset. A *k*-means clustering algorithm was applied to a large dataset. Classifiers were fed with information from clusters. In this case, this method outperformed other classifiers and ensemble methods.

2.2 Site hazard analysis

Study [10] focused on analyzing the occurrence of traffic accidents at the intersections. The correlation between accidents occurrence and geometric, traffic and control characteristics of signalized intersections can lead to increasing safety on the intersections. Data used in the study were overdispersed (variance is greater than mean). For this reason, a Negative Binomial model with stochastic component was used to describe the relationship between accident occurrence and the geometric design of the intersections. Random Effect Negative Binomial model can deal with spatial and temporal features of the accident. The most significant variables were traffic volume, the number of phases per cycle and uncontrolled left-turn lane.

Study [10] states that with increasing traffic volume, drivers have less opportunities for uncontrolled left-turns and tend to take the risk, which can lead to an accident. Increasing the section available for accelerating before uncontrolled left-turn significantly increases the safety of the intersection. On the other hand, bus stops near the intersection increase hazardousness of the site.

Study [11] used Classification and Regression Tree (CART) as a data mining technique. CART does not need any relationship information between predictors and targets. Study compared CART with the Negative Binomial Regression model, similar to the one used in [10]. The dataset contained information about accidents location, road information (number of lanes, horizontal curvature, slopes and shoulder width), traffic volume, injury levels and environmental characteristics (weather conditions, peak hours and lane distribution). CART model and Negative Binomial Regression model achieved similar results. This research also noted that traffic accidents are mostly caused by the combination of the factors rather than a single factor.

2.3 Imbalance and sparseness in datasets

In the past few years, researchers have begun to focus on application machine learning to real-world problems. Real-world problems are often difficult and are represented by sparse and complex datasets. These datasets are often imbalanced. Mining from imbalanced datasets causes various errors and thus decreases precision of the learning. Previous research stated that natural distribution is often not the best distribution to use machine learning on. Imbalanced datasets can be also characterized by sparseness in the feature space. Imbalance can be defined as imbalance among classes, but also within classes. Data within a single class can have imbalanced distribution which also produces sparseness.[12]

2.3.1 Simple methods dealing with imbalance

To reduce imbalance and sparseness of the datasets various methods have been developed. Popular methods are random and focused oversampling and undersampling, generation of samples based on given information and combination of these methods. Oversampling and undersampling and their variants both present useful points but also some negatives. Random undersampling can remove important samples with significant information. Oversampling can conduce to overfitting.[12]

2.3.2 Advanced methods and synthetic samples

Advancement of undersampling, such as Condensed Nearest Neighbor or Neighborhood Cleaning Rule is developed to shorten bigger classes by removing samples located near the border of the classes or similar samples that are further away from the decision border. On the other hand, focus resampling oversamples smaller classes that are close to the bigger classes. Experiments showed that these methods did not provide any significant improvement on the classifier's accuracy. Also, decision trees constructed from the oversampled datasets are usually unnecessary large and complex.[12][13]

Oversampling by duplication samples from minor classes leads to overfitting on multiple samples in the minor classes. This creates small and specific decision regions. Oversampling by creating new samples in minor classes can make decision region more general. The basic method is to create new samples by generating samples as average from N neighbors in the minor class with possibility to subsequently modify new samples by a function. One of the tested methods of creating new artificial samples is Synthetic Minority Oversampling Technique (SMOTE). SMOTE provides better results than random or focused oversampling/undersampling. Method generates more sophisticated samples. For this reason, minority class tends to be more covered, especially near boarders with the other classes. SMOTE was also used with ensemble-based methods. Ensemble-based methods accuracy is improved if the class distribution is balanced, in comparison to imbalanced datasets.[12]

3 Data analysis and models

In this chapter, I overview two main approaches of this work: the severity analysis and identification of hazardous sites. I describe two datasets, the Unified Transport Vector Map which contains accidents records in the Czech Republic in past years and OpenStreetMap elements and its features. I explain the selection of relevant features from given datasets and its preprocessing. Next, two main analyses, the severity analysis and identification of hazardous sites, are explained. Finally, machine learning models and models' evaluation and scoring techniques are described.

3.1 Accidents analysis

In this section, I will explain the two main analyses conducted on obtained datasets. The First analysis is called severity analysis. The Second analysis is called identifying dangerous sites.

3.1.1 Severity analysis

Severity analysis is an analysis which focuses on determining whether the accident is considered severe or not. In severity analysis, accidents are considered severe whenever any injury is suffered. On the contrary, non-severe accidents are accidents when no injury is suffered.

The aim of the severity analysis is to determine consequences of the accident from given input data about the accident. As input data, I used Unified Transport Vector Map [14] (data is described in section 3.2.1 Unified Transport Vector Map). Also, analysis can provide information about factors that are responsible for causing severe accidents.

3.1.2 Identifying dangerous sites

The aim of this approach is to identify hazardous sites on the traffic network. Sites are locations with high frequency of traffic accidents. If the number of accidents at defined site is higher than a reasonable threshold, the site is considered hazardous. Otherwise, the site is considered non-hazardous (safe). As input data, I use geographical locations of the recorded accidents and environment features around the accidents obtained from OpenStreetMap. OpenStreetMap is popular mapping tool in the world. Thus this approach can have a worldwide usage to identify dangerous sites.

3.2 Data analysis and preprocessing

This section analyze the two main datasets used (Unified Transport Vector Map and OpenStreetMap) and its preparations for machine learning methods. Afterwards, a method for defining sites as clusters of accidents is proposed.

3.2.1 Unified Transport Vector Map

Unified Transport Vector Map (UTVM) [14] is dataset from Ministry of Transport of the Czech Republic which contains records of the traffic accidents in the Czech Republic in past years. Records contain information about the crashes. Records have spatial information (geographic location, city and region), temporal information (time and date of the accident) and additional information specific to the accident (number of vehicles involved, weather conditions etc.).

3.2.2 Context and context-free features

I separate features from UTVM into three groups: context, context-free and unimportant features. Context features hold information which is specific to the given accident, for example weather conditions, alcohol measured, cause of the accident, type of vehicle etc. Context feature can hold very promising information that can help analyze crash accidents. However, they are specific to the accident record and specific to the UTVM dataset. It means the generality of this approach is lowered.

Context-free features are features that are not specific to the single accident but are specific to the crash site, thus they are more general than context features. Context-free features are for example: number of lanes, type of road, traffic control (traffic signals, right of way), a direction of driving etc.

Unimportant information is set of features that can have no or very little information that can help analyze causes of traffic accidents such as property damage cost, car's brand and model etc. These features were ignored from the UTVM dataset and were not used for analysis.

UTVM contains three attributes that define number of injured people:

a) Number of non-incapacitating injuries
b) Number of incapacitating injuries
c) Number of casualties

Absolute and relative counts of attributes defining number of injuries are depicted in Table 1. Incapacitating injuries and fatalities occurs very sparsely in accidents records. Two different representations of classes are proposed. In first representation, severe accidents are accidents with at least one incapacitating injury or casualty. This representation might better describe the nature of severity. Cost of this representation of the classes is that size of the minor class is very small compared to the major class. Second representation of classes defines severe accidents with at least one non-incapacitating, incapacitating injury or casualty. In other words, accidents are considered severe whenever any type of injury (including fatalities) is suffered. On the other hand, the accidents are considered not severe when no injuries are suffered in the event of the accident. In this case, minor class represents roughly 20% of the available records (Table 1).

At least - Injury	Description	Absolute Count	Relative Count [%]
$\boldsymbol{a}$)	Non-incapacitating	102243	16.80
$\boldsymbol{b}$)	Incapacitating	16429	2.70
$\boldsymbol{c}$)	Casualties	3989	0.66
$\boldsymbol{a}) + \boldsymbol{b}) + \boldsymbol{c}$)	Union	117314	19.28

Table 1: Injury distribution in crash accident recorded in the Czech Republic (2007-2013). Counts represent number of unique accident records where given injury occurred, not the total number of injured people.

3.2.3 Preprocessing UTVM

Most of the selected features (context and context-free features) in UTVM are categorical features. Categorical features have values from discrete domains. For each categorical feature, I created a separate table with two attributes: value from discrete domain and integer. Each table has N records, where N is the cardinality of the domain for given feature. This allows me to unite different values that have same or similar meaning for the crash site analysis, for example grouping months to seasons (Table 2). Grouping semantically similar or same values together drastically reduces search space. Also, grouping semantically similar or same values balances the distribution of frequent and less frequent values. Lastly, integer values are used directly as an input for classifiers so no more significant value preprocessing was needed.

On the other hand, countable variables such as the number of affected vehicles or number of injured people were not discretized.

Original Representation		Modified Representation	
JAN	1	4	Winter
FEB	2	4	Winter
MAR	3	1	Spring
APR	4	1	Spring
MAY	5	1	Spring
JUN	6	2	Summer
JUL	7	2	Summer
AUG	8	2	Summer
SEP	9	3	Fall
OCT	10	3	Fall
NOV	11	3	Fall
DEC	12	4	Winter

Table 2: UTVM feature preprocessing. Values from discrete domains that have same or similar semantical values are grouped together. This approach also increases balance between frequent and infrequent values of the particular features.

3.2.4 OpenStreetMap

OpenStreetMap (OSM) is community created free map data tool that allows users to insert, edit and receive data from the map. OSM has similar mapping interface as well-known Google Maps. A fraction of the map data can be exported in various formats for further operations.[15] OSM data consist of OSM elements and features describing the elements.

3.2.5 OSM elements

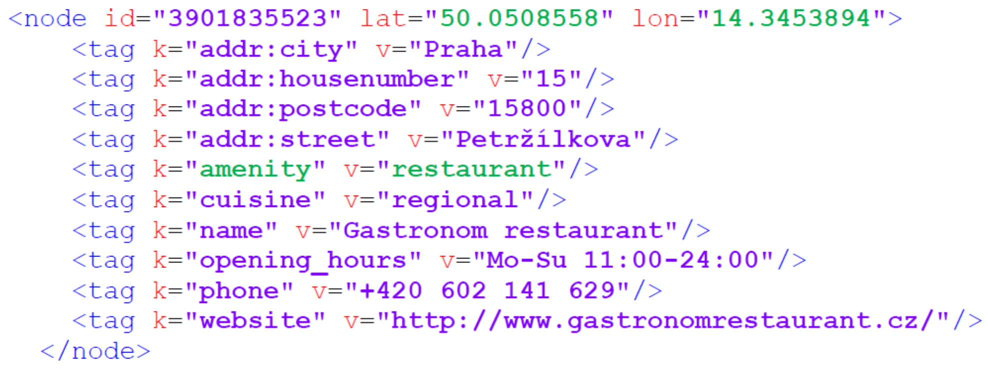
OSM data consists of three types of elements: *nodes*, *ways* and *relations*. *Nodes* are points that have defined location (geographic coordinates), *ways* are sets of *nodes* that define routes or areas (closed *ways*) and *relations* define logical or geographic relationships between other elements. Each of these three types of data is further described with tags. Tags are key-value pairs which provide more information about the element. [15][16]

Essential tag of OSM elements is a tag which is called primary. Primary tag defines a type of the element. There exists many different primary tags on OSM elements. Primary OSM features are represented as pre-defined key-value pairs.

Commonly denoted as `key=value`. A full list and description of all commonly used primary features can be found online.[16]

OSM *node* provides information about the geographical location. I only focus on the location and primary tag (type) of the *node*, for example whether the element is a school, traffic signal, restaurant etc. (Figure 1) *Ways* has a very similar data structure with the exception that instead of latitude and longitude *way* location is defined as a set of nodes. Regarding *ways*, I also only focus on the primary tag of the *way* such as highway, parking, roundabout etc. I omit analyzing *relations* because they provide no useful information for traffic accidents analysis. Figure 1 depicts XML data representation of OSM element. Information taken into account (geographical location and primary *tag*) is highlighted in green.

Analyzing secondary *tags* like opening hours of the restaurant can remove certain amount of noise when the time parameter is taken into account. However, secondary *tags* are less frequent and this analysis would add a lot of complexity to the task. Also, missing secondary *tags* in the map data could increase the error of the classification.



```
<node id="3901835523" lat="50.0508558" lon="14.3453894">
    <tag k="addr:city" v="Praha"/>
    <tag k="addr:housenumber" v="15"/>
    <tag k="addr:postcode" v="15800"/>
    <tag k="addr:street" v="Petržílkova"/>
    <tag k="amenity" v="restaurant"/>
    <tag k="cuisine" v="regional"/>
    <tag k="name" v="Gastronom restaurant"/>
    <tag k="opening_hours" v="Mo-Su 11:00-24:00"/>
    <tag k="phone" v="+420 602 141 629"/>
    <tag k="website" v="http://www.gastronomrestaurant.cz/"/>
  </node>
```

Figure 1: OSM node XML representation. Location and the primary tag is highlighted in green. Highlighted information is obtained and used for analysis.

3.2.6 Clustering accidents

Raw location of recorded accident does not provide information whether the site is actually a dangerous site or not. To determine dangerous sites I need to find locations or areas where there are more accidents recorded in comparison to the other sites. Accidents that happened reasonably near each other are defining the dangerous site. On the other hand, the site cannot be vast because it will no longer be a site but a large area. I define sites using a well-known method Hierarchical clustering. Using Hierarchical clustering I set the spatial size of clusters and I construct clusters without prior knowledge about the number of clusters constructed.

It is important that clusters are big enough to cover the dangerous site but two different potentially hazardous or non-hazardous places are different clusters. Complete-linkage clustering appears to be optimal for this purpose. Complete-linkage does not allow creating a single linkage among more sites. That means distant accident records belong to different clusters. The criterion to form clusters is Euclidean distance. The number of accidents in clusters defines hazardousness of the site. Sites are considered hazardous when they contain more elements than given threshold.

To divide sites into non-hazardous (negative) and hazardous (positive) I define two values:

- θ_N – non-hazardous (negative) threshold
- θ_P – hazardous (positive) threshold

Each site includes a number of accidents. Let's denote this number as $|S|$. Negative sites (S_N) are sites with less or equal number accidents than θ_N. Positive sites (S_P) are sites with greater or equal number of accidents than θ_P. Ignored sites are sites that are defined as neither positive nor negative.

$$S \in \mathrm{S_N} <=> |S| \leq \theta_N \tag{1}$$

$$S \in \mathrm{S_P} <=> |S| \geq \theta_P \tag{2}$$

The perfect way to determine a non-hazardous site is to sample geographical locations on roads and intersections where there is no or a minimal amount of the accidents recorded. However, I determine non-hazardous sites as clusters with a minimal amount of accidents only. Information loss about the places where zero accidents occurred is obvious. This information loss is not significant. A justification for this is that in urban areas accidents were recorded very densely and accidents cover most segments of the roads.

3.2.7 Selecting OSM features

From commonly used primary *tags* I select 74 different primary *tags* (features). Selected features have a possibility to carry some information that can help defining hazardous sites. The more obvious features that can carry some information are for example pedestrian crossings (`highway=crossing`) or bus stops (`highway=bus_stop`). Less obvious features that has possibility to carry some information are for example bars (`amenity=bar`) or schools (`amenity=school`) where there is a higher possibility of incautious people causing accidents. I omit obviously unimportant features which cannot affect the accidents or features that are very rare (e.g. windmill, defibrillator station etc.).

For each site (cluster), I obtain the number of occurrences of all selected OSM elements within the certain radius from the centroid of the cluster. These data are used as input for machine learning models for detection potentially dangerous sites. Although the relative spatial position of the OSM elements to the accidents or among other features can provide more information than just number of occurrences, it adds a lot of complexity to the task.

3.3 Models and evaluation

In this section, I briefly describe machine learning models used for the severity analysis and identification of potentially hazardous sites. All selected models achieved promising results in previous studies and are robust to different representations of data and parameter selections. Evaluation methods are described in the last part of this section.

3.3.1 Random Forests

Random forest is a well-known ensemble method that generates and combines individual decision trees. Individual decision trees are learned using different fractions of the training set called bootstrap samples, also called bagging (bootstrap aggregating). Learning decision trees on bootstrap samples tends to decrease variance. This is a huge advantage in comparison to the single decision trees which are often deep and overfitted.[17] Random forest uses an out-of-bag estimation. Out-of-bag estimation is using a selection of bootstrapped samples to calculate prediction error on trees that do not contain said samples.[18]

Random forest results are also robust to parameter selection. In study [19] default values for parameters of the random forest achieved similar result that tuned parameters. The number of trees in the Random Forest increases computational time linearly. However, a large amount of trees slightly increased the stability of the model but the change of stability is negligible. [19]

Random forest uses sophisticated methods to calculate the feature importances. Importance of features is calculated by taking each feature into account individually and in combination with other features. The most popular are Gini importance and permutation accuracy importance measure.[17]

3.3.2 Gradient Boosting

Similarly to the Random forest, Gradient boosting is an ensemble method which constructs and combines multiple classification or regression models. Gradient boosting, in general, allows a combination of boosting and optimization.[20] Boosting is similar to the technique called Adaptive Resampling and Combining (ARCing), which is presented in paper [21]. Gradient Boosting

Machine was presented in paper [22]. Gradient boosting model generates individual decision trees, from which the final tree is constructed using voting methods and averaging. Boosting, in comparison to bagging, use a different method to resample data. Bootstrap samples that were consistently misclassified have a higher probability to be selected.[23]

Study [23] shows that Gradient boosting was not outperformed by classification and regressions trees and Generalized additive models.

3.3.3 Model Stacking

Past studies develop classifiers or ensemble of classifiers with single learning method. The most common models are decision trees and neural networks. A more sophisticated way is combining different types of classifiers or ensembles of classifiers (with heterogeneous model representation). This approach is called stacking.[24]

Study [24] compares the performance of model stacking with individual classifiers. Results show that stacked models performed similarly to the best individual classifier used.

3.3.4 Models evaluation

In case of binary classification, we typically describe classes as positive class and negative class. A number of samples correctly classified (predicted class is the same as actual) as positive are called True Positives (TP). A number of samples correctly classified as negative are called True Negatives (TN). A number of samples incorrectly classified as positives are called False Positives (FP). And finally, a number of samples incorrectly classified as negatives are called False Negatives (FN). [12]

From these values, we can derive terms *precision*, *recall* (true positive rate), *fallout* (false positive rate) and F value: [12][25]

$$precision = \frac{TP}{TP + FP} \tag{3}$$

$$recall = TPR = \frac{TP}{TP + FN} \tag{4}$$

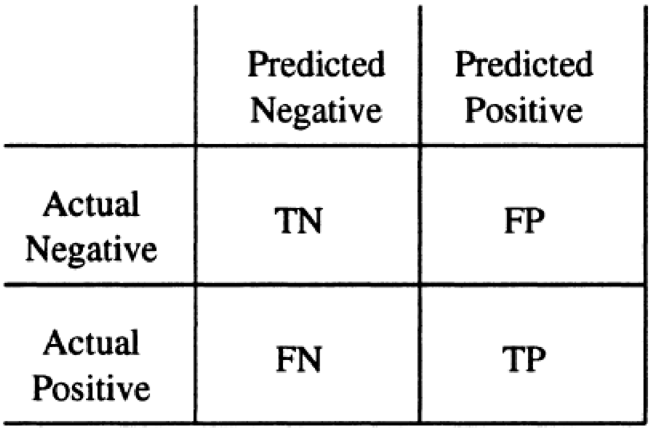
$$fallout = FPR = \frac{FP}{FP + TN}$$

$$F_\beta = \frac{(1 + \beta^2) * precision * recall}{\beta^2 * precision + recall} \tag{5}$$

ß represent relative importance between *precision* and *recall*, the usual value is 1, thus F1 Score:

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \tag{6}$$

Receiver Operating Characteristics (ROC) describes a relation between true positive rate and false positive rate. An area under the ROC curve (AUC) is representing classifier performance. Higher values represent better performance. In my case (binary classification) F1 score and AUC are both appropriate and very popular ways to determine the performance of the classifier. Classification results can also be depicted as a confusion matrix (Figure 2). In case of binary classification, confusion matrix consists of four values creating a 2x2 matrix (TN, FP, FN, TP). Numbers of correctly classified samples are located on the main diagonal of the confusion matrix. [12][25]



	Predicted Negative	Predicted Positive
Actual Negative	TN	FP
Actual Positive	FN	TP

Figure 2: Confusion matrix [12]

4 Implementation

In this chapter, I describe implementation of processes defined in previous chapter. Firstly, I preprocess accident records dataset from UTVM. Secondly, clustering method to define sites is described. Third section describes a method of obtaining map data from OSM and preparation of map data. Lastly, classification methods are implemented using popular machine learning platforms and libraries.

4.1 Retrieving and preprocessing UTVM

UVTM accidents records can be gathered from the webpage of UTVM [14] using web API scripts. The methodology used for acquiring the data is described in the paper [26]. I use already downloaded and available dataset online [27]. Dataset is a csv file that consists of 608 557 records of traffic accidents in years 2007 – 2013. Damaged records (11328) that contain corrupted values or where most values were missing were removed and not used for classification. Csv file is imported to the PostgreSQL database.

Each record contains 50 features that needed to be preprocessed (more information about features can be found in section 3.2.2 Context and context-free features). Unimportant features were left out. Most important features (context and context-free features) are categorical features. For better manipulation with the dataset, I created for each categorical feature a separate definition table (for more information see section 3.2.3 Preprocessing UTVM). This allows me to easily change values for each particular categorical feature. Also, I construct various database views to select different sets of features and create conditions for selection only specific portions of the dataset.

4.2 Defining sites

To define hazardous and non-hazardous sites I use beforementioned Hierarchical clustering. This method allows constructing geographical clusters without prior knowledge about the number of clusters. Also, with complete-linkage clustering, I am able to control the size of the clusters and separate relatively close sites into different clusters.

Using Hierarchical clustering on large datasets such as all available accident records in the Czech Republic from 2007 – 2013 is highly inefficient. To minimize computational time and memory allocation I develop a simple recursive method that allows construction of clusters in a small amount of time. The method consists of recursively splitting map region into four quarters. Each quarter is subsequently split into another four regions until region has less or equal to accidents recorded than given threshold (final region). I set the threshold to 10000 records. I construct clusters on each of the final regions separately. A drawback of this method is that accidents that occurred on the boundary of the final regions cannot belong to the same clusters even if they are reasonably close to each other. Reason for this is that their clusters are constructed in separated Hierarchical clustering runs.

To construct clusters I use `scipy.cluster.hierarchy.fclusterdata` method within Python-based library SciPy. The input to the method is set of geographical coordinates and the output is set of geographical locations of centroids and sets of accident records belonging to the particular clusters. Figure 3 depicts constructed clusters from accident records in residential area in Ostrava. Cluster positions are reasonably distributed on the roads. Clusters with centroids that are not located on the roads can still hold valuable information. OSM elements are obtained within given radius from cluster's centroid. The area around the site covers important environmental features in most cases.

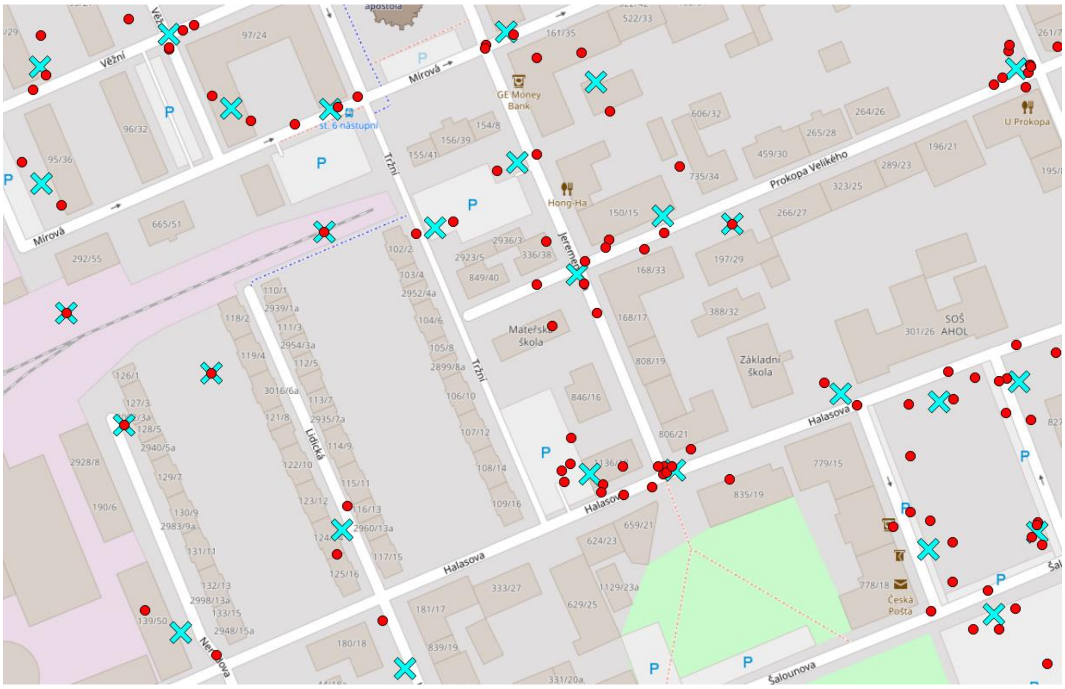


Figure 3: Clustering accidents to define sites. Accidents are represented as red points. Sites (clusters' centroids) are represented as blue crosses. Sites with a high number of accidents are considered hazardous.

4.3 Acquiring and preprocessing OSM data

For each site (cluster), I need to gather information about the environment and other objects that could affect hazardousness of the site. Information about the surroundings is contained in nearby OSM elements. Reasonably selected features, that could have some effect on the accidents, were selected (for more information about selected features see section 3.2.7 Selecting OSM features). Numbers of occurrences of selected features around the centroid of the cluster (site) were gathered from OSM elements. Radius for acquiring OSM elements is set to 50 meters. This radius can cover even bigger road segments, intersections or roundabouts.

4.3.1 Overpass API Query Language

To gather data from OSM I use an API called Overpass.[28] Using Overpass API I can run Overpass API queries to download and further analyze sections of data from the OSM. Overpass API queries provide functionality to search fragments of a map and filter OSM elements. To build Overpass API queries I use Overpass Query Language. By running Overpass queries I can filter multiple different elements and save them as sets. On these sets, Overpass API queries allow using basic set operations like union, difference or intersection of the sets. Sophisticated conditions can be constructed to filter OSM elements. It is possible to filter elements that contain a given key-value pair. Also, it is possible to filter elements that contain (or do not contain) a certain type of *tag*. Conditions can also be defined using regular expressions and wildcards. Last but not least, Overpass QL allows using recursion. Using recursion, elements that are directly or indirectly linked with a certain set of elements can be selected.

To restrict the search to a specific section of the map a rectangle called Bounding Box is used. Bounding Box is a rectangle defined by minimal and maximal latitude and longitude. A different method to constrain specific section of the map is using a set of geographic coordinates that define a polygon which bounds the search area. Last but not least, method `around` selects the circular area around a given geographic location and given radius. The area around the geographical location is the most straightforward method to select appropriate section of the map around the site.

4.3.2 Obtaining OSM data

For each cluster, I construct an Overpass query that use cluster centroid's geographic location as parameter and selects all OSM elements with primary *tags* that have been selected within 50 meters around the centroid of the cluster. To send requests to the server, I use a simple wrapper called *overpass*.[29] XML format as response format of the request is chosen. Afterwards, I parse the XML response using XML Path Language (XPath). XPath allows navigation through elements and attributes in XML documents. Using XPath I count occurrences of all OSM elements containing selected features. This data was subsequently saved to the database.

In the database, each cluster (site) record consists of centroid's geographical coordinates and counts of selected OSM elements near the site. Also, each accident record was updated with a link to its respective cluster. This database model allows obtaining UTVM context or context-free features for particular clusters. In addition, it is possible to filter clusters based on UTVM features which were used to select clusters in urban areas with help of UTVM records which hold information about the city where the accident happened. Otherwise, the information about the city can be found in OSM elements.

4.4 Classification

Classification models were constructed, trained and evaluated in Python language using popular machine learning platforms and libraries [30][31][32].

Creation of training and test set is described in section 4.4.2 Training and test datasets. Classifier performance is calculated on the test set. In severity analysis, I use the F1 score as a criterion of performance. For identification hazardous sites I used AUC as a criterion of performance.

4.4.1 Models implementation

Firstly, in severity analysis, I use `RandomForestClassifier` from free software machine learning library `scikit-learn`.[30] To achieve the best performance of classifier it is recommended to tune its parameters. Although, the Random Forest is a robust method in consideration of parameters I perform a parameter tuning process. Tuning process consists of an exhaustive search of the combination of parameters and cross-validation learning on the dataset. The exhaustive search for parameters is performed by `GridSearchCV` from the `scikit-learn` library.[30]

From parameters of `RandomForestClassifier,` I tune `n_estimators`, `max_depth` and `max_features`. The first mentioned parameter is a number of trees constructed. By increasing number of trees constructed, computational time increases linearly. The second is the maximal depth of the constructed trees. This value should be adjusted to control overfitting. The third is the maximal number of features considered in splitting. A higher value should improve individual tree performance for the cost of the computational time and less diversity of the trees. It is recommended to find a profitable trade-off between the individual tree performance and the diversity of the trees. The default value is a square root of the number of features in samples.

Secondly, in severity analysis I use Gradient boosting implementation called Extreme Gradient Boosting or `xgboost`.[31] Specifically, I use `XGBClassifier`. To improve the performance of classifier I find optimal parameters of the classifier using exhaustive search `xgboost.cv`. The process of the searching is same to the one mentioned above. I tune `n_estimators`, `max_depth`, `learning_rate`, `min_child_weight`, `colsample_bytree` and `scale_pos_weight`. The number of estimators and max depth of the tree has the same characteristics as in the Random Forest. `colsample_bytree` is equivalent to `max_features` mentioned above. `min_child_weight` is used to control overfitting. `learning_rate` is coefficient of weight minimization in each step. `scale_pos_weight` adds weight to the positive class. It is used when training on imbalanced datasets.

Thirdly, in identification hazardous sites analysis I use both Random Forest and Extreme Gradient Boosting classifiers. Models are implemented using machine learning platform H2O.ai [32] (`H2ORandomForestEstimator` and `H2OGradientBoostingEstimator`). Reason for choosing H2O.ai over `scikit-learn` library is that H2O.ai includes functionality to stack models. H2O.ai `H2OStackedEnsembleEstimator` is used to stack models. The input to the estimator consists of trained individual models and training set. The trained stacked model should provide better results than individual models. It is possible to specify metalearner algorithm type in training stacked ensembles: generalized linear model, Gradient Boosting Machine, Random Forest or deep learning.

4.4.2 Training and test datasets

For each approaches (severity analysis and identifying hazardous sites analysis), I need to use different datasets. Also, for each different type of experiment, I use slightly modified datasets. For this reason, for each type of experiment, I construct specific view on database tables. The view provides whole dataset already prepared as an input to a model. Data is consequently split into training and test set. Training set represents input data with the expected output. It is used for model training. Test set represents samples from dataset other than training set. On the test set model predicts value which is afterward compared to the expected value. The number of correct and incorrect predictions defines confusion matrix and performance score of the classifier is calculated. Separation of training and test set varies throughout the experiments.

Both of the approaches are binary classifications. For this reason, we can call samples positives and negatives. In severity analysis, I define positives as samples where at least one person was injured or worse. Negatives samples represent accidents where no injury was suffered. In identifying hazardous sites analysis, positives are sites where the number of accidents recorded is greater than hazardous (positive) threshold θ_P. Negatives are sites where the number of accidents is less than non-hazardous (negative) threshold θ_N. Hazardous and non-hazardous thresholds vary throughout the experiments. Also, different proportions of positive and negative samples in the training set is tested.

Retrieving datasets directly from views in the database did not require any more significant preprocessing. However, time spent to retrieve all data from the database appeared to be needlessly long. To improve time data retrieval I use serialization. Retrieving data from the serialized object, instead of directly retrieving data from database, significantly increased the computational time of preprocessing phase.

To connect to the PostgreSQL database from Python code I use a popular PostgreSQL adapter *psycopg*.[33] For python-object serialization and deserialization, I use a module called *pickle*.[34]

4.4.3 One-Hot Encoding

One-Hot Encoding (One-of-K scheme) is a process that transforms categorical features into a table of binary values. This scheme minimizes relation among categorical values that are not correlated. Table of binary values tends to improve the performance of classification models in comparison to the standard categorical variable input. To encode categorical features from UTVM I use `OneHotEncoder` from free software machine learning library `scikit-learn`. One-Hot Encoding greatly increases feature space which in larger datasets can lead to a well-known curse of dimensionality when by increasing feature space the predictive performance of model tends to decrease. On the other hand, One-Hot Encoding mitigates incorrect correlation among categorical features which can improve the performance of the model.

5 Experiments

This chapter provides results of conducted experiments of both severity and site hazard approaches. Classifiers performance is compared, significant factors in identification of the severity and hazardous sites are highlighted. Chapter concludes with brief discussion of achieved results and offers ideas for the future work in the traffic accidents analysis.

5.1 Accident severity analysis

The first part of my work was to analyze accident severity. Analysis mainly consists of how well classifier can classify accident as severe and not severe solely on recorded context and context-free data. In this analysis spatial location was part of the data. The significance of particular features affecting the severity of the accidents was calculated. Datasets consist of features from UTVM and were later enriched with data of map features from OSM to analyze improvement of classification.[14][16]

5.1.1 Severity prediction using context dataset

Context dataset includes all selected features (context and context-free) from UTVM accidents records in the Czech Republic within years 2007 – 2013. Context dataset holds 608557 records with 24 features. To simplify classification I chose binary output of accident severity: severe or not severe accident. A severe accident is an accident where at least one person was injured or worse. Otherwise, the accident is identified as not severe. Property damage was not taken into account.

I struggle to find an optimal boundary between severe and not severe accident. Due to small a number of incapacitating injuries and casualties recorded (Table 1), I am marking accident as severe even if only one person suffers a non-incapacitating injury. Reason for this is that this type of injury is the most common injury among other injuries (71.68%). Identifying accidents where only a few people suffer non-incapacitating injuries as not severe or ignoring them completely will greatly decrease the size of the severe class. The proportion of positive samples will decrease from 19.28% to approximately 5% (depending on modification of the severity threshold).

Despite this fact, I conduct an experiment where the accident is considered severe if at least one incapacitating injury is suffered. Feature distribution in positive class this small is very sparse and contains little to none information due to the randomness of severe accidents (Figure 4).

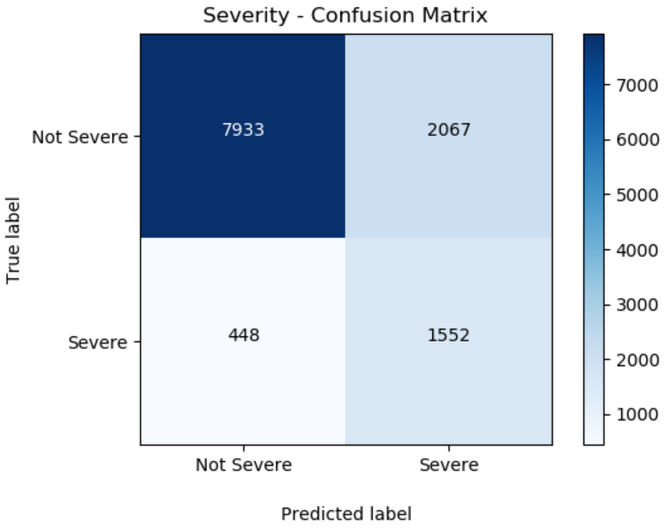


Figure 4: Severity classification – confusion matrix. Severe accidents are defined as at least incapacitating injury. Prediction of severe accidents is poor due to a small number of positive samples and high sparseness of the positive (severe) class.

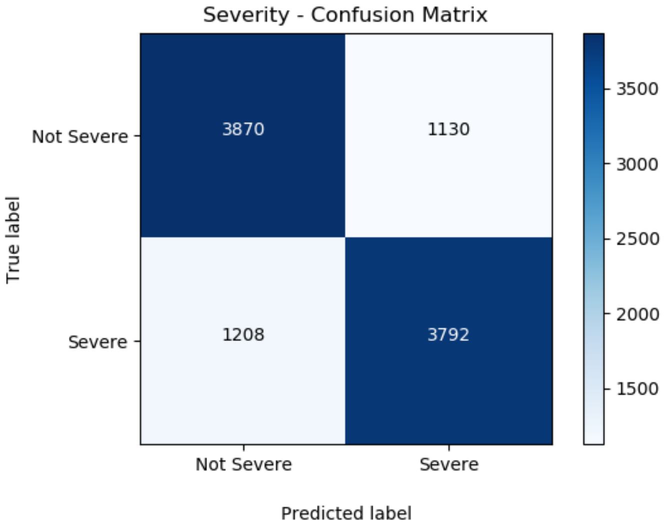


Figure 5: Severity classification – confusion matrix. Context dataset. Severe accidents are defined as at least one injury or fatality. Prediction is more accurate and balanced than experiment shown in Figure 4.

Next, severe accidents were defined as a union of all injuries (Table 1). In other words, even a single non-incapacitating injury is causing the accident to be considered severe. Confusion matrix (Figure 5) shows that result of classification on this class definition is more balanced. FP/TP ratio is significantly smaller. Classification error is still high. Approximately 30% of samples are incorrectly classified (Figure 5). Classification performance was slightly increased with parameter turning and implementation One-Hot Encoding (Table 3). On the other hand, One-Hot Encoding significantly increased computation time, which might be ineffective in training on larger datasets. Gradient boosting method slightly outperformed Random forest classifier (Table 3). Also, training on larger dataset slightly improved performance (Table 3).

Model	Parameters Tuned	One-Hot Encoding	Training Samples	Runs	AVG (F1)	MAX (F1)
XGB	no	no	10000	51	0.764603	0.771096
XGB	no	yes	10000	117	0.766313	0.778148
RDF	yes	no	10000	45	0.755002	0.766283
XGB	yes	yes	10000	16	0.766207	0.776271
XGB	yes	no	10000	50	0.765439	0.774052
XGB	yes	no	100000	11	0.773901	0.778300

Table 3: Severity analysis - comparison of experiments. Random Forest Classifier performed slightly worse than Extreme Gradient Boosting model. One-Hot Encoding has a small effect in improving the performance of the classifier.

The most important features used in classification can be seen in Figure 6. Affected object is the most dominant feature. It holds a type of vehicle or other object with which the car collides, for example non-track vehicle (car, bus), stationary object (lamp, parked car), tram, train, pedestrian or others. Among other strongly influenced feature is a direction of impact. The direction of impact can be rear end, head on, sideward or none. Driving direction can be direct, opposite or turn. All of these features are reasonable factors in determining severity (injuries) of the accident.

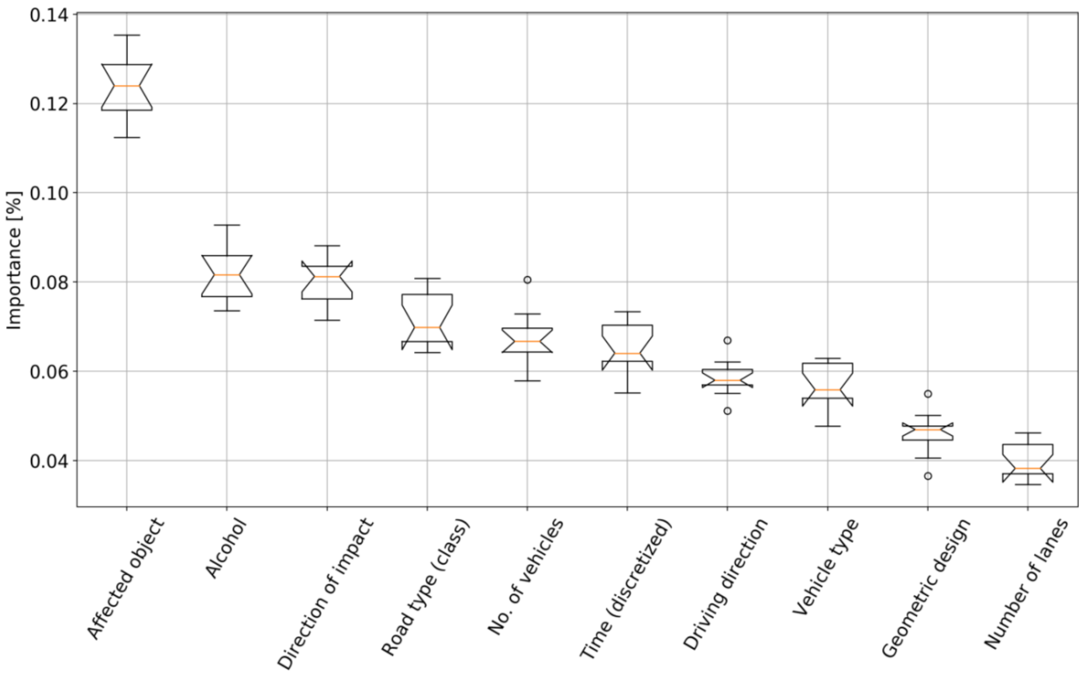


Figure 6: Feature importances in severity analysis – context data.

5.1.2 Severity prediction using context-free dataset

The aim of restriction to context-free data is to generalize classifier on environment variables and not to be coupled to the accident-specific features. The context-free dataset is a portion of the dataset from UTVM that does not consist of any specific information about the one given accident record. Context-free dataset holds 608 557 records with 11 features. Features are mostly environment features at the crash site such as the number of lanes, type of intersection and others. To improve classifier performance I added, in addition to environment features, temporal information to the context-free dataset (discretized time and date). The generality of the classifier is not discarded. However, real-data input for classifier needs to include discretized time and season information.

Classifier performance on this data is considerably lower than classification on context dataset (Table 4). Only three context-free features appear as relatively important features in classification on the context dataset (Figure 6). I can state that UTVM context-free data does not contain enough information needed to determine the severity of accidents.

Model	Parameters Tuned	One-Hot Encoding	Training Samples	Runs	AVG (F1)	MAX (F1)
XGB	yes	no	10000	50	0.668983	0.682830
XGB	yes	yes	10000	50	0.665612	0.677111
XGB	yes	no	100000	96	0.676324	0.684576

Table 4: Severity analysis on the context-free dataset – comparison of experiments. Larger training set performed better on average, but the best score is only slightly better than classifier trained on a small training set.

Performance decrease is also evident on confusion matrix on Figure 7 in comparison to classification on the context dataset (Figure 5). The classifier is biased towards positives samples. Figure 8 depicts the importance of features in severity classification on context-free dataset. From Figure 8, we can see various anomalies. It is interesting how high classifier apportioned importances of some features like time and day of the week. Time and day of the week can determine traffic volume. However, the relation between traffic volume and severity of the accidents is unknown and could be analyzed in the future works. Accidents on different road types, like highways, can cause more serious injuries.

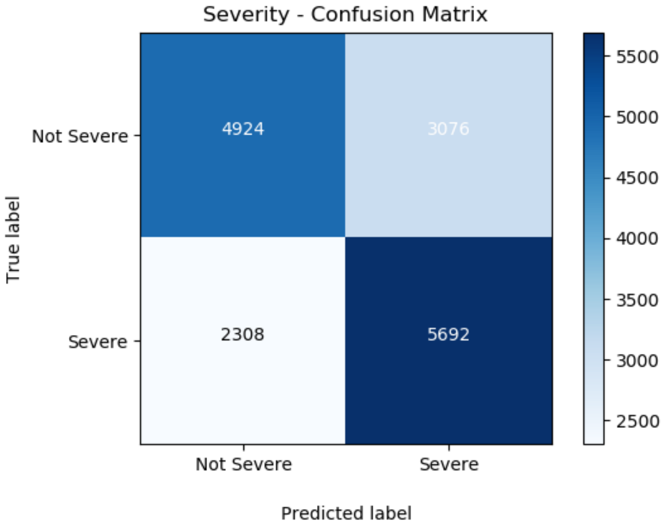


Figure 7: Severity analysis on the context-free dataset – confusion matrix. Prediction is less accurate in comparison to the context dataset (Figure 5). Classification is biased towards positive samples.

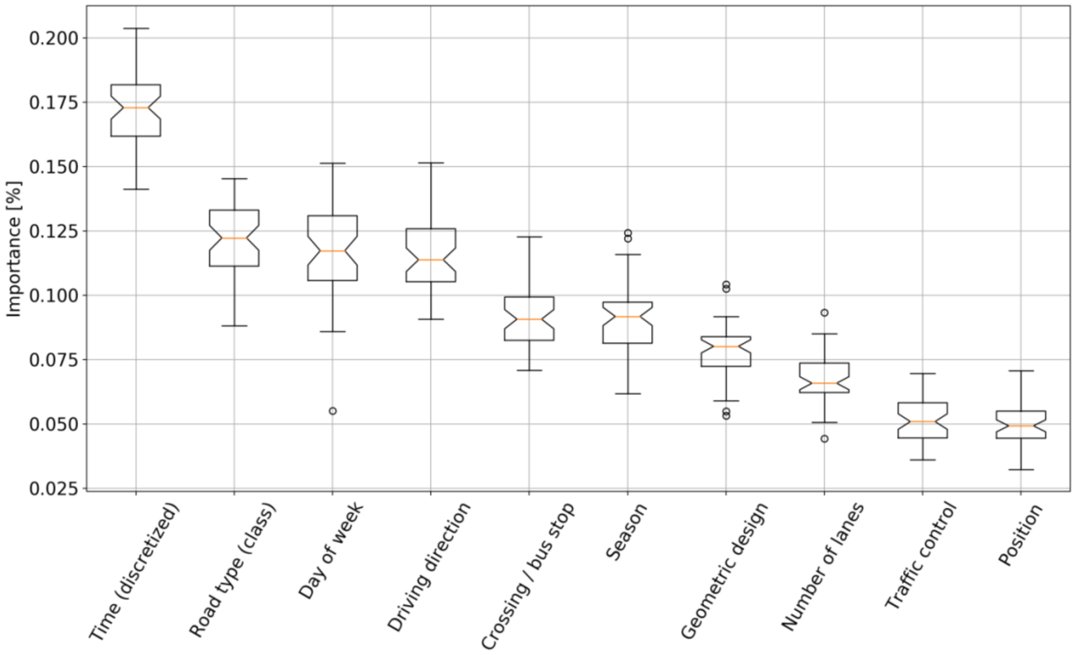


Figure 8: Feature importances on context-free dataset.

5.1.3 Enriching datasets with OSM data

In this section, I describe experiments with enriching context and context-free datasets with features from the OSM. The aim of these experiments was also to determine the severity of the accidents. Input data was enriched with counts of OSM elements. Firstly, enriching context data with counts of OSM elements slightly improved classification performance (Figure 9, Table 5). Even though score increase is not large OSM features do have an effect on classification on context dataset. A justification for this is that four OSM features are in ten most important features during classification (Figure 10). Important OSM features can indicate the occurrence of pedestrians (footways, highway crossings and residential streets). This result demonstrates that accidents, where cars are hitting a pedestrian can be closely linked with severity of the accidents.

Dataset	Model	Training Samples	Runs	AVG (F1)	MAX (F1)
Context	XGB	100000	11	0.773901	0.778300
Context + OSM	XGB	100000	8	0.786742	0.794713
Added value				+0.012841	+0.016413

Table 5: Context dataset: comparison of UTVM data and enriched UTVM data with OSM features

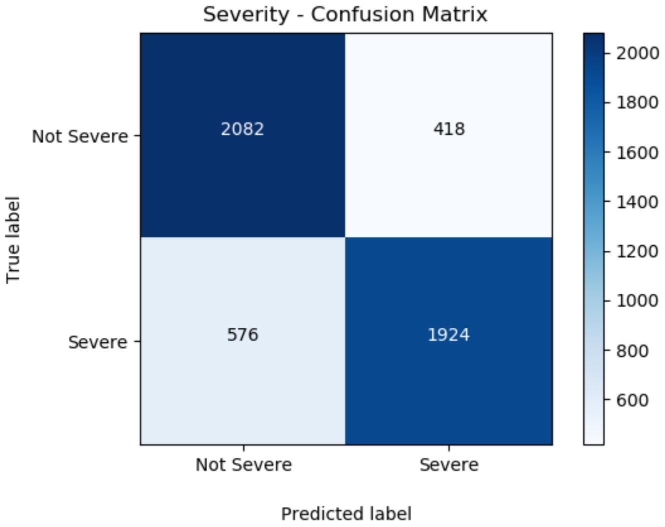


Figure 9: Severity analysis on context dataset enriched with OSM features. Ratio FP/TP is lower in comparison to basic context dataset (Figure 5).

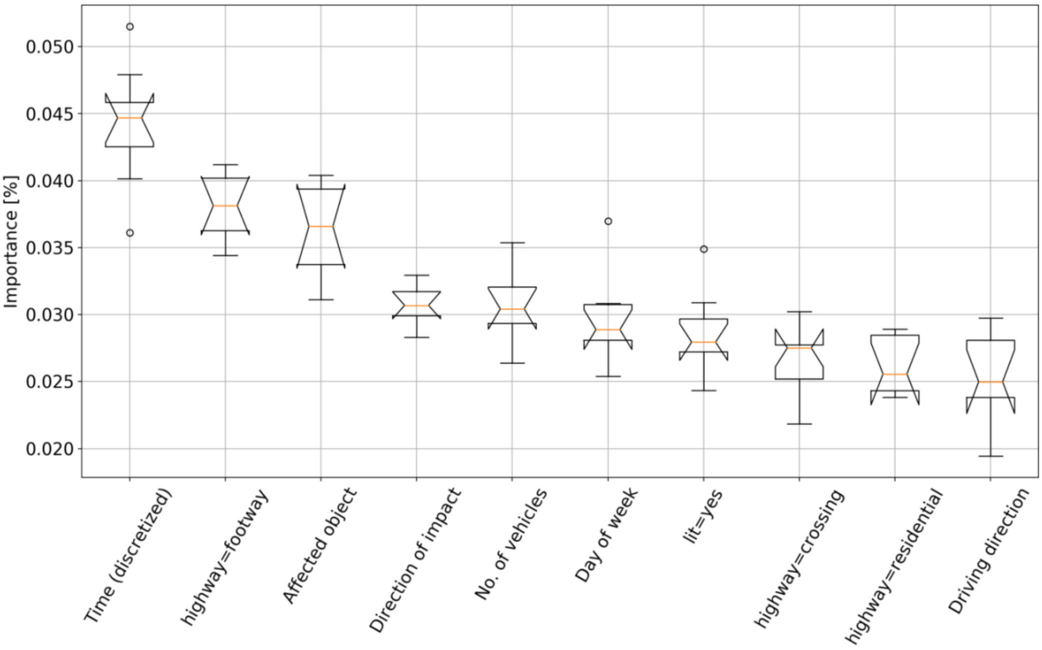


Figure 10: Feature importances on context data enriched with OSM features.

Secondly, an enriching context-free dataset with counts of OSM elements did not improve classification performance (Figure 11, Table 6). Performance of classification on the enriched context-free dataset is slightly worse than on basic data. However, fall of the score is very small and can be caused by stochastic nature of the model. Also, the relative number of true positives, true negatives, false positives and false negatives is very similar to result from classifier trained on the basic context-free dataset. Within the ten most important features in this experiment, we can see only one OSM feature – `highway=footway` (Figure 12). The existence of footway near the recorded accident is also the most important OSM feature in classification on context dataset.

On the other hand, footway is very common element near most roads besides highways, trunk roads or by-pass motorways.

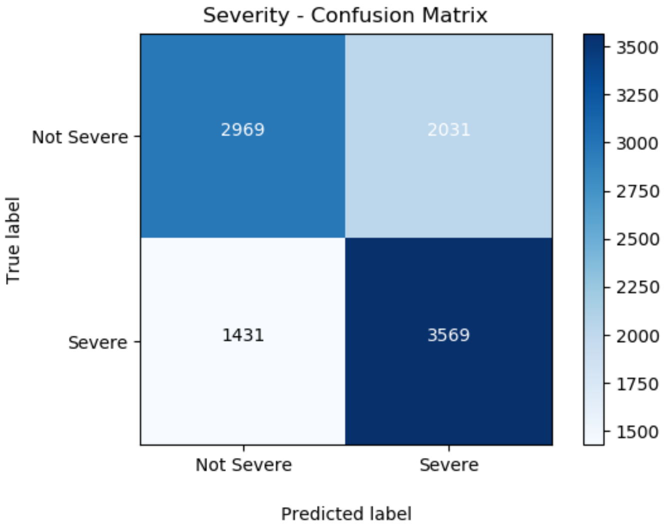


Figure 11: Severity analysis on context-free data enriched with OSM data. The classifier is biased towards the positive samples. The relative number of TP,TN,FP,TN is very similar to result from classifier trained on the basic context-free dataset.

Dataset	Model	Training Samples	Runs	AVG (F1)	MAX (F1)
Context-free	XGB	10000	50	0.668983	0.682830
Context-free + OSM	XGB	10000	52	0.664005	0.673396
Added value				-0.004978	-0.009434

Table 6: Context-free dataset: comparison of UTVM data and enriched data with map features

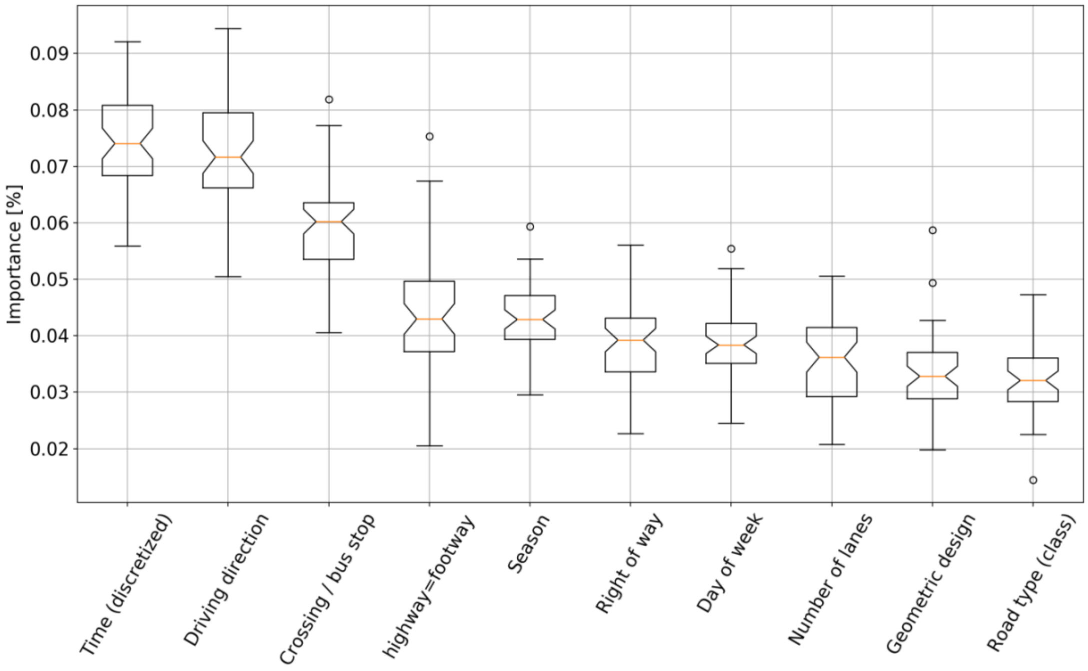


Figure 12: Feature importances on context-free data enhanced with OSM data.

5.1.4 Severity analysis evaluation

Experiments conducted to analyze severity of the accidents provided promising results. Classification on the context dataset achieved significantly better result than on context-free dataset. This was anticipated. Important features from context datasets, namely direction of impact, vehicle type and alcohol support results of previous studies. Weather conditions appeared to not have significant effect to the accidents severity. This fact was also recognized by previous studies.

On the other hand, results of classification on the context-free dataset show that context-free dataset does not hold enough information needed for accurate identification of the severity of the accidents. Map features obtained from OSM have marginal effect on classification. However, more complex data from OSM like geometric design of the roads or position of the OSM elements related to the location of the accidents can hold valuable information for the severity analysis.

5.2 Identifying hazardous sites

The aim of this approach is dedicated to identify hazardous locations on the maps. Data needed to identify dangerous sites are only obtained from OSM. In other words, a trained classifier can determine hazardousness of any location in the world which is sufficiently mapped in OSM. Thus, this approach can theoretically have world-wide usage. Unfortunately, world-wide usage of the classifier is practically impossible, due to uneven traffic behavior in different parts of the world. Different parts of the world have different types of roads or traffic volumes. I assume that a classifier trained on bigger cities in the Czech Republic can be efficiently used only in similar cities in Europe, the US and other parts of the world where traffic behavior is similar to the traffic in the Czech Republic. The classifier trained on the Czech Republic will most probably fail in regions with different behavior in traffic such as India where traffic is much more chaotic (no lanes, no traffic lights etc.).

5.2.1 Clustering

Firstly, I defined sites. Using Hierarchical clustering on recorded accidents I created clusters with their respective centroids. The site is a circular area with the center as the centroid of the cluster and radius of approximately 50 meters. Each site contains information how many accidents are recorded belonging to the site. The number of accidents in the cluster is later used to define hazardous and non-hazardous sites.

As accident records I used UTVM. To preserve generality of the classifier I only used geographic coordinates of the accidents. The total number of accidents recorded is 608 557. Geographic location was not available in 11 328 records. These records were ignored in process of constructing the clusters. From available accidents records with valid geographic coordinates (597 227), I constructed the total of 253 849 clusters in the Czech Republic. For the construction of the clusters, I used complete-linkage clustering. The distance was chosen Euclidean with a threshold of 0.0005 in the decimal format of geographic coordinate system. This value represents approximately 50 meters radius and was calculated using the haversine formula. [35] Constructed clusters can be seen on Figure 3.

5.2.2 Defining positive and negative samples

Each site contains the number of recorded accidents. In Figure 13, we can see that majority of the clusters contains only one accident. This is caused by accident records outside the big cities. Large-scale rural areas cover a major portion of the Czech Republic. Accidents in rural areas are sparser, thus probability that two or more accidents occurred within the same site is lower. Outside the ten biggest cities in the Czech Republic, approximately 62% of the recorded accidents happened. On the other hand, distribution of the number of accidents in the clusters in Prague (Figure 14) and the ten biggest cities in the Czech Republic (Figure 15) is much more balanced.

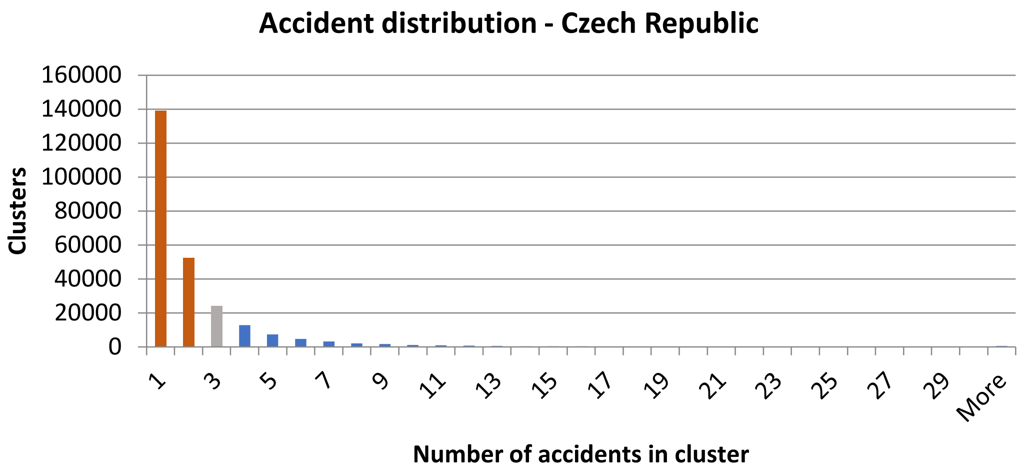


Figure 13: Distribution of the number of accidents in clusters in the Czech Republic. Red bars indicate non-hazardous sites. Blue bars indicate hazardous sites. Note: threshold on hazardous sites varies throughout the experiments.

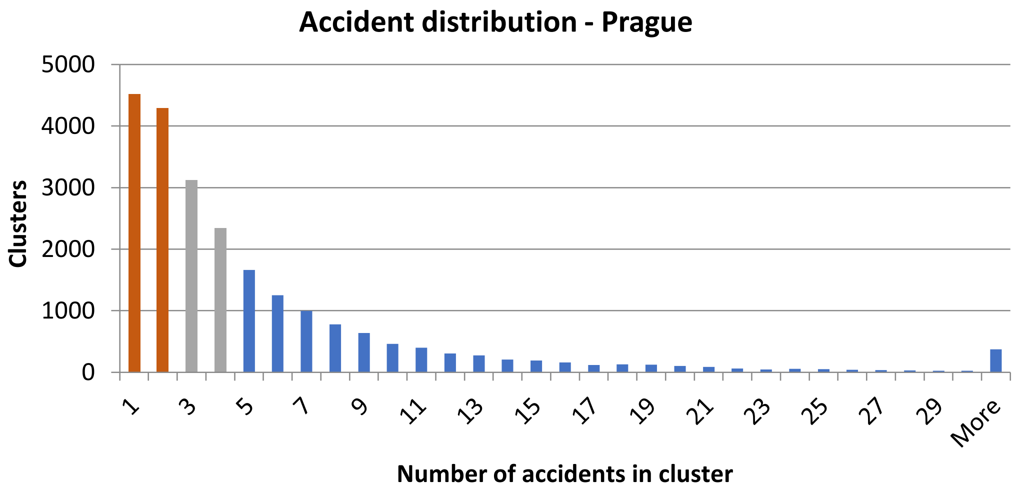


Figure 14: Distribution of the number of accidents in clusters in Prague. Red bars indicate non-hazardous sites. Blue bars indicate hazardous sites. Note: threshold on hazardous sites varies throughout the experiments.

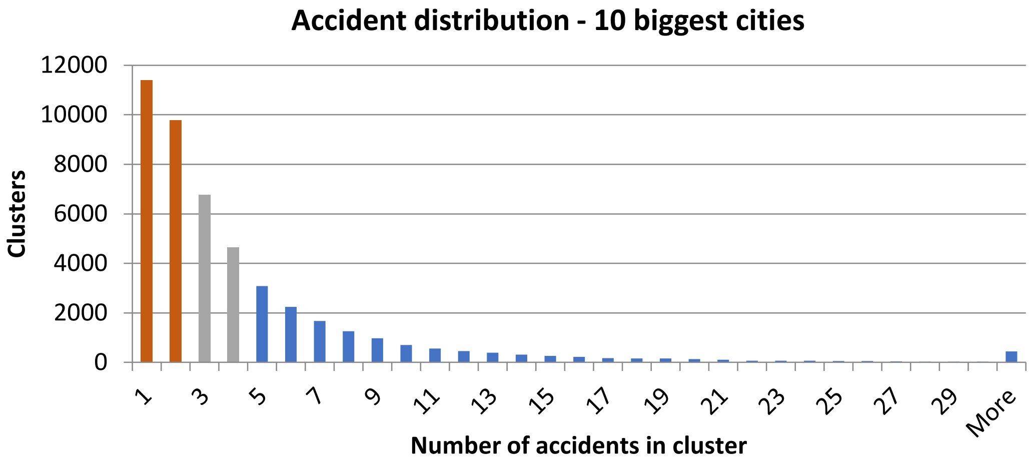


Figure 15: Distribution of the number of accidents in clusters in ten biggest cities in the Czech Republic. Red bars indicate non-hazardous sites. Blue bars indicate hazardous sites. Note: threshold on hazardous sites varies throughout the experiments.

Due to different distributions of accidents count per cluster I struggled to identify the thresholds of the non-hazardous and hazardous sites. Thus I conducted various types of experiments on urban/rural areas in the Czech Republic with different thresholds of hazardousness. For all experiments, I used a model of stacked ensembles mentioned in section 3.3.3 Model Stacking, as it provides slightly better performance than individual models. Model stacking consists of Extreme Gradient Boosting model and Random Forest Estimator model. Experiments are denoted as following: `<scope>-<representation>-<`θ_P`>`, where `<scope>` is either Czech Republic (CZE), ten biggest cities (URB) or Prague (PRG), `<representation>` is either counts (C) or binary (B) and θ_P is hazardous threshold.

In the experiment CZE-C-4 I set the non-hazardous (negative) threshold $\theta_N = 2$ and hazardous (positive) threshold $\theta_P = 4$ on the sites in the whole country. In other words, sites with 2 or fewer accidents were considered safe, sites with 4 or more accidents were considered dangerous, sites with exactly 3 accidents were ignored (for more information see section 3.2.6 Clustering accidents). Although a high number of samples (100 000) are available for training, classifier did not perform as well as the other experiments conducted later (Table 8). A possible reason for this is that traffic in urban and rural areas has very different nature. Also, a difference between positive and negative samples was not significant. Spreading gap between θ_N and θ_P better describes the nature of the classes.

In addition, samples that include very few distinct features cannot hold relevant information about the hazardousness of the site. A justification for this is that sample with one feature, e.g. highway crossing, cannot define the hazardous site because all of the locations where there is a highway crossing should be then considered hazardous. For this reason, I chose to ignore samples with only zero, one or two different features. For future experiments, I only used samples where there are present at least three or more different features.

In the experiment URB-C-5, the classifier was fed with data from ten biggest cities in the Czech Republic. Also, I overspread difference between positive and negative samples by increasing θ_P to 5. Performance increase is significant (Table 8). But the cost of this approach is that efficient spatial scope of the classifier is decreased. This classifier is trained to determine site hazard only in urban areas.

In the experiment URB-C-7, I spread the gap between negative and positive samples by increasing θ_P to 7. Results are slightly better than in experiment URB-C-5 (Table 8). However, increasing θ_P is causing that hazardous sites tend to be mostly on the roads with high traffic volumes (bypasses, trunk roads etc.). This phenomenon can be seen in Figure 16 where blue points are true positives and red points are false positives (incorrectly classified as positives). In most cases, false positives are also on the roads with high traffic volumes. This implies that classifier found a correlation between dangerous sites and high-frequent roads. This correlation is reasonable, but the aim of the classifier is to determine that site is dangerous due to a combination of the aspects in the vicinity like schools, bus stops, bars or others and not simply because the site has a high traffic volume.

In addition, most negatives samples occur in residential areas (Figure 17). Purple points are hazardous sites that classifier evaluated as non-hazardous (false negatives). This supports correlation mentioned above.

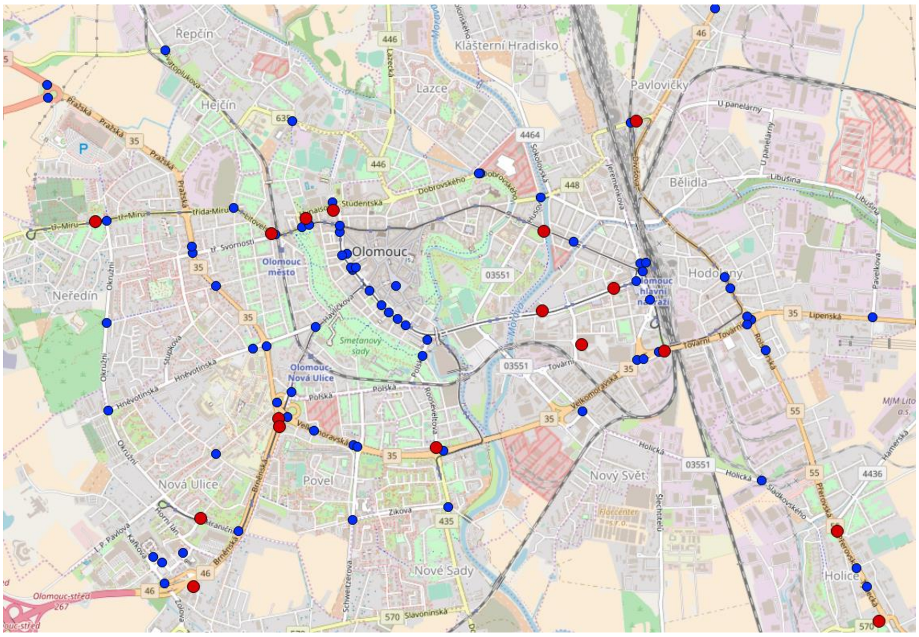


Figure 16: Samples classified as positive in Olomouc. Blue points represent true positives (hazardous sites that are correctly classified as hazardous). Red points are false positives (samples that are negative but incorrectly classified as positives).

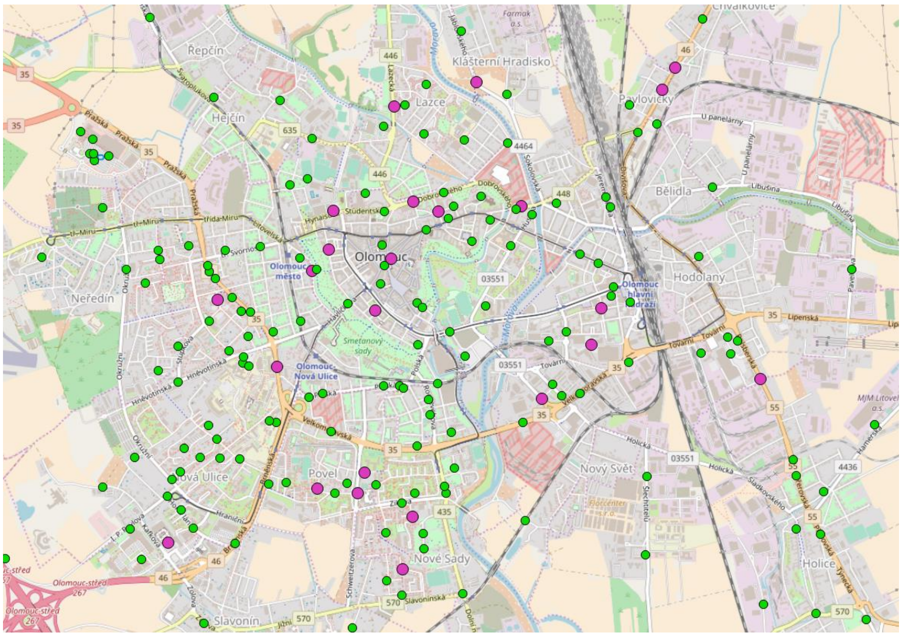


Figure 17: Samples classified as negative in Olomouc. Green points represent true negatives (correctly classified as negatives), purple points represent false negative (incorrectly classified as negatives).

5.2.3 Mitigating effect of frequent roads

In previous experiments, input to classifiers was counts of all selected OSM features. For example, the site has six traffic lights, four crossings and two bus stops. The average sum of counts of all selected feature for positive samples is 21.08, for negative samples is 11.33 (Table 7). Sites with higher occurrences of OSM elements tend to have higher traffic volumes.

To decrease the correlation between site hazard and traffic volume I used binary a representation of the OSM features as input data. The binary representation of the OSM features is a representation that for each feature there can be only true-false value instead of the number of occurrences of given element is on the site. Sites then have features whether the element is present or not. In this data representation, positive samples have an average sum of all distinct present features equal to 7.07, negative samples have 4.81 (Table 7). The gap between the number of features in positive and negative samples has decreased.

Representation	$\boldsymbol{\theta_N}$	$\boldsymbol{\theta_P}$	Positives	Negatives
Counts	2	5	19.02	11.33
Counts	2	7	21.08	11.33
Counts	2	9	22.64	11.33
Binary	2	5	6.71	4.81
Binary	2	7	7.07	4.81
Binary	2	9	7.3	4.81

Table 7: Average sums of features in the input dataset. Difference between average sums of features in positive and negative samples is increasing with increasing $\boldsymbol{\theta_P}$. Using binary representation of OSM features significantly reduces the gap.

In the experiment URB-B-5, the binary representation is used. The positive threshold is $\theta_P = 5$. Performance of classifier, in comparison to experiment URB-C-5, has slightly lowered (Table 8). Performance decrease was expected due to the removal of the information about the counts. Also, in experiment URB-B-7 the binary representation achieved slightly worse results than the experiment URB-C-7, which have the same θ_P=7 (Table 8).

Using binary representation, I expected a decrease in correlation between hazardousness of the sites and traffic volume. Figure 18 depicts no significant decrease in this correlation. False positives (red squares) are the non-hazardous

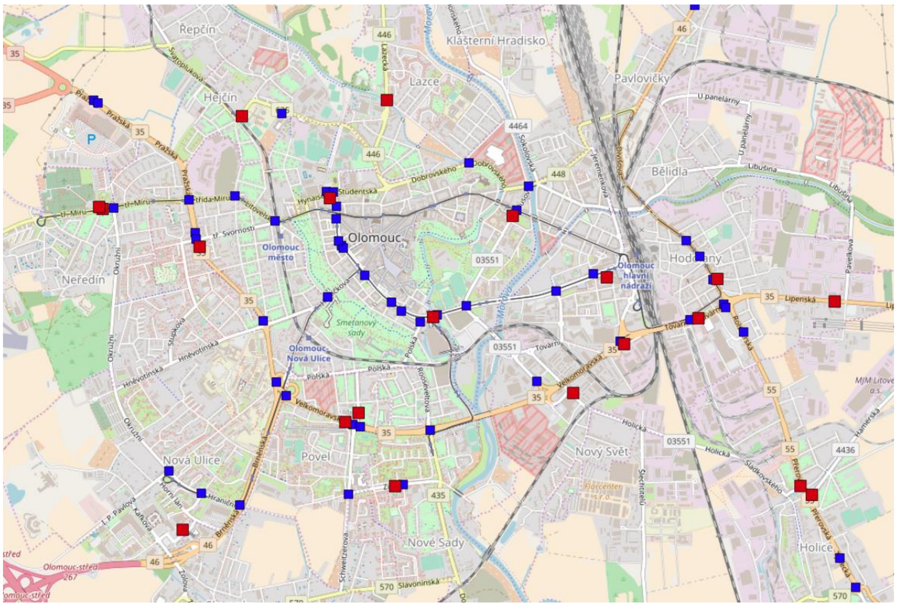


Figure 18: Samples classified as positive in Olomouc – binary representation. Blue squares represent true positives (hazardous sites that are correctly classified as hazardous). Red squares are false positives (samples that are negative but incorrectly classified as positives).

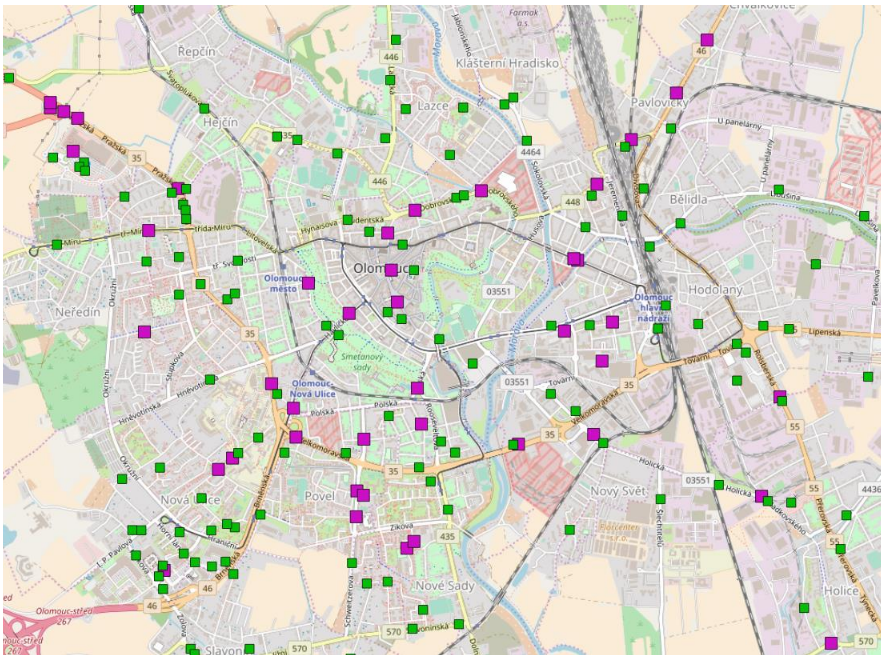


Figure 19: Samples classified as negative in Olomouc – binary representation. Green squares represent true negatives (correctly classified as negatives), purple squares represent false negative (incorrectly classified as negatives).

sites that were incorrectly classified as hazardous. They are mostly on trunk roads or other frequent roads. Also, purple squares in Figure 19 are hazardous sites that were classified as non-hazardous. In most cases, they are located in residential areas but classifier was unable to identify them as positives.

Finally, in the experiment PRG-B-5, the classifier was fed with data only from Prague as the biggest and the densest city in the Czech Republic. The binary representation with $\theta_P = 5$ was used. Results are slightly better than experiment URB-B-5 (Table 8). However, the classifier was trained and tested only on the data from Prague. Thus, the classifier is less general.

5.2.4 Evaluation of results

In this section, I conclude results of the experiments in identifying dangerous sites. Although, results of experiments in urban areas are promising the biggest problem occurred was that I was unable to determine the precise boundary between hazardous and non-hazardous sites. Spreading the gap between θ_N and θ_P better describe nature of non-hazardous and hazardous sites. However, with increasing gap and by using only the number of accident records at the defined sites a phenomenon appeared. Sites classified as hazardous were sites at the roads with high traffic volumes such as trunk roads, bypasses and frequent roads. Sites classified as non-hazardous were mostly in residential areas where the traffic volume is not that high. Figure 20 depicts only positive samples from the dataset. The classifier was able to correctly identify hazardous sites in frequent roads but was unable to identify positive (hazardous) sites in the residential areas.

Experiment	Representation	θ_N	θ_P	Training Samples	P/N Ratio	Runs	AVG (AUC)	MAX (AUC)
CZE-C-4	Counts	2	4	100000	0.3	7	0.738283	0.738714
URB-C-5	Counts	2	5	26800	0.36	9	0.830776	0.832507
URB-C-7	Counts	2	7	22000	0.2	10	0.833085	0.834840
URB-B-5	Binary	2	5	26800	0.36	15	0.816555	0.817070
URB-B-7	Binary	2	7	22000	0.2	6	0.820818	0.821172
PRG-B-5	Binary	2	5	13300	0.5	14	0.826125	0.826926

Table 8: Comparison of experiments and performance of classifiers on various data setup. P/N ratio represents ratio of positive and negative samples in training set. AVG (AUC) and MAX (AUC) are average and maximum (best) of area under a ROC curve throughout all runs.

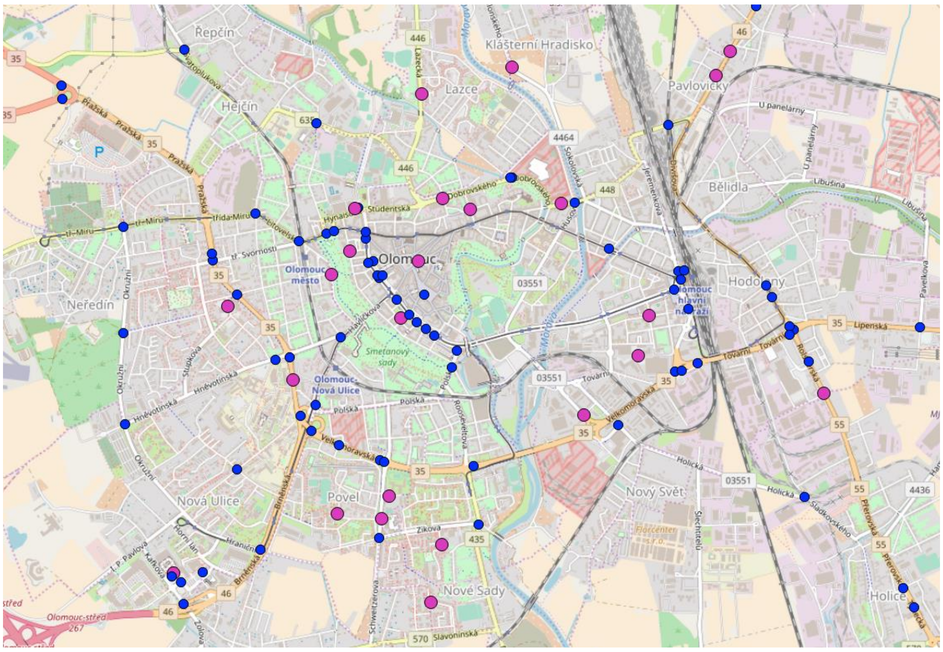


Figure 20: Positive samples in Olomouc. Blue points represent sites correctly classified as positives (true positives). Purple points represent sites that are hazardous but classified as non-hazardous (false negatives).

Confusion matrices in Figure 21 and Figure 22 compares results of the classifiers trained in urban areas with count and binary representations. Training on binary representation increases false negative rate (Figure 22). Binary classification slightly lowers performance in identification positive samples.

Importances of particular OSM elements calculated by classifiers fed by data from urban areas are similar in both count and binary representation (Figure 23, Figure 24). Highway crossings have more importance than other features in both representations. However, crossings occur at intersections which are expected to be more dangerous than other sections of the roads.

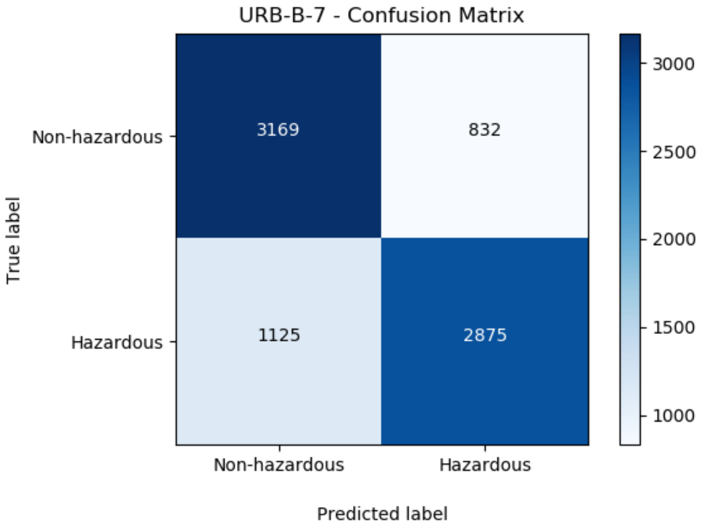


Figure 21: Identifying hazardous sites in urban areas – confusion matrix. Counts representation is used. The number of false negatives is slightly higher than false positives. The classifier is failing in identifying hazardous sites on non-frequent roads and residential areas (Figure 17).

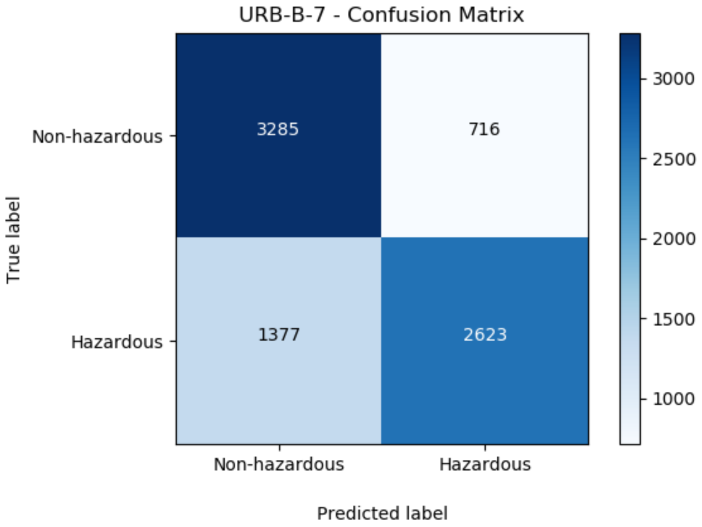


Figure 22: Identifying hazardous sites in urban areas – confusion matrix. The binary representation of OSM features is used. The number of false negatives is higher than false positives. The classifier is failing in identifying hazardous sites on non-frequent roads and residential areas (Figure 19).

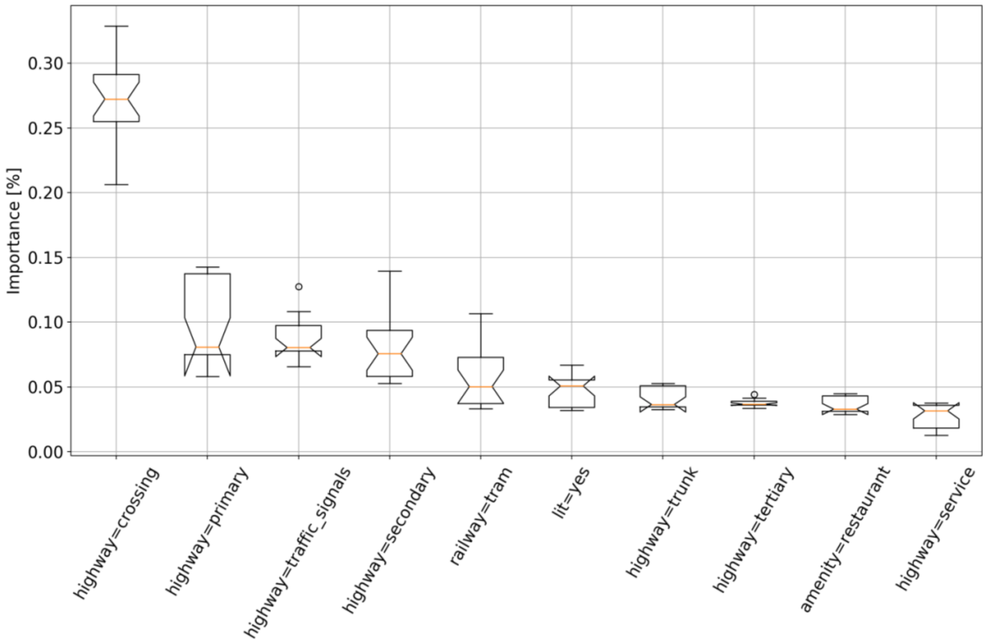


Figure 23: Importance of OSM elements using count representation on urban areas in the Czech Republic. Note: lit=yes indicates that site was lit by street lights, highway=service is a service station to eat something, often found at motorways.

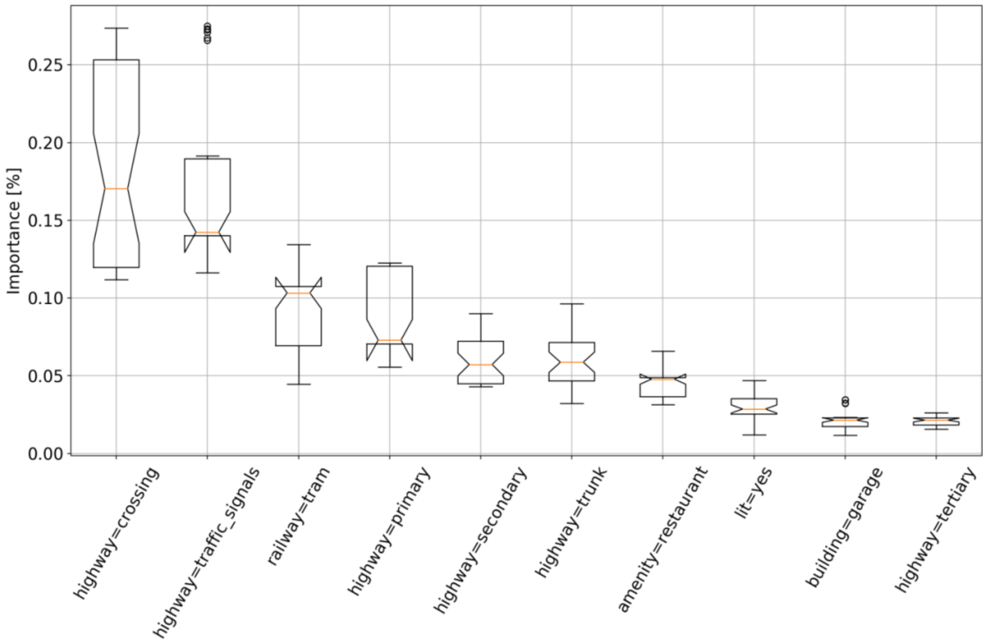


Figure 24: Importance of OSM elements using binary representation on urban areas in the Czech Republic. Note: lit=yes indicates that site was lit by street lights.

5.3 Discussion

The approach was dedicated to identification the severity of the accidents and key factors affecting severity of the accidents. Results of experiments showed that context dataset hold information from which the severity of the accident can be predicted. Key factors appeared to be affected object, direction of impact, alcohol, type of vehicle and others. These factors were also recognized, by previous studies, to have effect on the severity of the accidents.

The approach of identification hazardous sites can be examined deeper. The concept has potential to have world-wide usage. Identifying dangerous sites can reduce the number of accidents, thus save people's lives, health and property. Although results of conducted are promising, the main problem of analysis was that classifiers, trained on designed specification of hazardous sites and selection of OSM features, tend to identify roads with higher traffic volumes as hazardous. Reason for this is that number of accidents has a correlation with frequency of traffic.

To minimize correlation between the number of accident and traffic volume, the information about the traffic volume can be used to normalize absolute numbers of accidents. This information has potential to train the classifier for its original purpose, which is identifying dangerous sites.

Also, more information about the surroundings of the site, like traffic signs or speed limits, can significantly improve the performance of identification. Next, I only used the number of OSM elements present near the site. The relative position of the elements or trajectories of the roads at the site can hold promising value for classification. Trajectories of the roads and intersections' shape can be primary factors affecting the safety on the roads.

This approach has potential to have various applications. Firstly, machine learning methods can be used in designing new traffic network. A proposed segment of the road or intersection can be furthermore analyzed using machine learning methods to verify its safety or draw attention to potentially dangerous factors.

Secondly, machine learning methods can be used on already existing traffic network sections that are confirmed as hazardous. I assume that, by analyzing the section, models can find factors responsible for site hazard and offer a modification to reduce the hazard. For example, analysis on the modified intersection, with an added stop sign or on-ramp, can deliver valuable information to reduce the site hazard.

Thirdly, analysis regions where accidents are not recorded or do not contain information about the location can be used to identify dangerous sites as well.

Last but not least, models can be used for real-time notifications about hazardous sites for drivers. In other words, application in smartphones or GPS devices can request a remote server with current location or planned route. The server can afterwards run an analysis to alert drivers to increase wariness near potentially hazardous locations.

6 Conclusion

This work studies two approaches to the traffic accident analysis. The first approach focuses on determining the severity of the accidents and factors responsible for severe accidents. The second approach focuses on identifying hazardous sites in the traffic network. In both approaches, ensemble models provide promising results. Research on learning on imbalanced datasets was performed. Methods to improve learning on imbalanced datasets were not needed due to a large amount of samples in datasets.

The main dataset used for severity analysis was obtained from Ministry of Transport of the Czech Republic. The dataset contains records with spatial, temporal and specific information about accidents such as alcohol measured, weather conditions and others. Selection and preprocessing of appropriate features which are responsible for the severe accident was a complex task. Different sets of features were selected (specific and/or general). Accident records were later enriched with numerous environmental elements (highway crossings, bus stops, schools) from OpenStreetMap (OSM) to improve classification performance. Results of experiments support findings of previous studies that alcohol, direction of impact and type of vehicle are key factors in determining accident severity and weather conditions do not have a significant effect on severity. Map features have a marginal effect.

The dataset for identification hazardous sites consists of the geographical location of recorded accidents in the Czech Republic. Sites were defined as clusters of accidents based on distance. The number of accidents in cluster defines hazardousness of the site. For each site, nearby environmental elements from OSM were obtained. Although results of experiments are very promising, a correlation between site hazard and traffic volume is present. More factors invariant to the traffic volume can be examined to decrease the correlation. Key factors in identifying hazardous sites are highway crossings, traffic lights, presence of secondary road and tram rails. These factors indicate frequent intersections. Also, classifiers performed better in the identification of hazardous sites in urban areas. A possible reason for this is that accidents in rural areas are sparser and rural areas have less environmental elements.

More information about the accidents could significantly improve classifiers performance. A geometric design of the traffic network can hold essential information needed for both analyses. Information about traffic volume can remove correlation between site hazard and frequent roads.

Bibliography

[1] S. Krishnaveni and M. Hemalatha, “A Perspective Analysis of Traffic Accident using Data Mining Techniques,” *Int. J. Comput. Appl.*, vol. 23, no. 7, pp. 975–8887, 2011.

[2] D. Delen, R. Sharda, and M. Bessonov, “Identifying significant predictors of injury severity in traffic accidents using a series of artificial neural networks,” *Accid. Anal. Prev.*, vol. 38, no. 3, pp. 434–444, 2006.

[3] S. Y. Sohn and S. H. Lee, “Data fusion, ensemble and clustering to improve the classification accuracy for the severity of road traffic accidents in Korea,” *Saf. Sci.*, vol. 41, no. 1, pp. 1–14, 2003.

[4] M. Bédard, G. H. Guyatt, M. J. Stones, and J. P. Hirdes, “The independent contribution of driver, crash, and vehicle characteristics to driver fatalities,” *Accid. Anal. Prev.*, vol. 34, no. 6, pp. 717–727, 2002.

[5] L. Kim, K., Nitz, L., Richardson, J., & Li, “Personal and Behavioural Predictors of Automobile Crash and Injury Severity,” *Accid. Anal. Prev.*, vol. 27, no. 4, pp. 469–481, 1995.

[6] W. T. Yang, H. C. Chen, and D. B. Brown, “Detecting safer driving patterns by a neural network approach,” *Intell. Eng. Syst. Through Artif. Neural Networks*, vol. 9, pp. 839–844, 1999.

[7] P. J. Ossenbruggen, J. Pendharkar, and J. Ivan, “Roadway safety in rural and small urbanized areas,” *Accid. Anal. Prev.*, vol. 33, no. 4, pp. 485–498, 2001.

[8] I. M. Abdalla, R. Raeside, D. Barker, and D. R. D. Mcguigan, “An investigation into the relationships between area social characteristics and road accident casualties,” *Accid. Anal. Prev.*, vol. 29, no. 5, pp. 583–593, 1997.

[9] M. Chong, A. Abraham, and M. Paprzycki, “Traffic Accident Analysis Using Machine Learning Paradigms,” *Informatica*, vol. 29, pp. 89–98, 2005.

[10] H. C. Chin and M. A. Quddus, “Applying the random effect negative binomial model to examine traffic accident occurrence at signalized intersections,” *Accid. Anal. Prev.*, vol. 35, no. 2, pp. 253–259, 2003.

[11] L. Y. Chang and W. C. Chen, “Data mining of tree-based models to analyze freeway accident frequency,” *J. Safety Res.*, vol. 36, no. 4, pp. 365–375, 2005.

[12] N. V. Chawla, “Data Mining for Imbalanced Datasets: An Overview,”

Data Min. Knowl. Discov. Handb., pp. 875–886, 2009.

[13] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data," *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 20–29, 2004.

[14] "Unified Transport Vector Map." [Online]. Available: http://www.jdvm.cz/. [Accessed: 01-Jan-2017].

[15] P. Weber and M. Haklay, "User-Generated Street Maps," pp. 12–18, 2008.

[16] "OpenStreetMap Wiki." [Online]. Available: http://wiki.openstreetmap.org. [Accessed: 01-Jan-2017].

[17] C. Strobl, A. L. Boulesteix, A. Zeileis, and T. Hothorn, "Bias in random forest variable importance measures: Illustrations, sources and a solution," *BMC Bioinformatics*, vol. 8, 2007.

[18] L. Breiman, "Out-of-Bag Estimation," *Tech. Rep.*, pp. 1–13, 1996.

[19] R. Díaz-Uriarte and S. Alvarez de Andrés, "Gene selection and classification of microarray data using random forest," *BMC Bioinformatics*, vol. 7, pp. 1–13, 2006.

[20] J. H. Friedman, "Stochastic Gradient Boosting," *Comput. Stat. Data Anal.*, vol. 1, no. 3, pp. 1–10, 1999.

[21] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," vol. 139, pp. 23–37, 1995.

[22] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Stat.*, vol. 29, no. 5, pp. 1189–1232, 2001.

[23] G. G. Moisen, E. A. Freeman, J. A. Blackard, T. S. Frescino, N. E. Zimmermann, and T. C. Edwards, "Predicting tree species presence and basal area in Utah: A comparison of stochastic gradient boosting, generalized additive models, and tree-based methods," *Ecol. Modell.*, vol. 199, no. 2, pp. 176–187, 2006.

[24] B. Zenko, "Is Combining Classifiers Better than Selecting the Best One?," *Mach. Learn.*, vol. 54, no. 3, pp. 255–273, 2004.

[25] D. M. W, "Evaluation: From Precision, Recall and F-Measure To Roc, Informedness, Markedness & Correlation," *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011.

[26] L. Veselý, "Návrh a tvorba databáze dopravních nehod v ČR," Technical University of Liberec, 2014.

[27] L. Veselý, "Databáze dopravních nehod ČR." [Online]. Available: https://www.ludekvesely.cz/databaze-dopravnich-nehod-cr/. [Accessed: 01-

Jan-2017].

[28] “Overpass Turbo.” [Online]. Available: https://overpass-turbo.eu/. [Accessed: 01-Jan-2017].

[29] Martijn van Exel, “Overpass API Python Wrapper.” [Online]. Available: https://github.com/mvexel/overpass-api-python-wrapper. [Accessed: 01-Jan-2017].

[30] “scikit-learn.” [Online]. Available: http://scikit-learn.org.

[31] “Scalable and Flexible Gradient Boosting.” [Online]. Available: http://xgboost.readthedocs.io/en/latest/. [Accessed: 01-Jan-2017].

[32] “H2O.ai.” [Online]. Available: https://www.h2o.ai/.

[33] “Psycopg - PostgreSQL adapter for Python.” [Online]. Available: http://initd.org/psycopg/. [Accessed: 01-Jan-2017].

[34] “Python object serialization,” 2009. [Online]. Available: http://docs.python.org/library/pickle.html. [Accessed: 01-Jan-2017].

[35] J. J. Mwemezi and Y. Huang, “Optimal Facility Location on Spherical Surfaces: Algorithm and Application,” *New York Sci. J.*, vol. 44, no. 77, pp. 21–28, 2011.

Appendix A

User Guide

This section provides a guide for running methods used in this work. Data for classifiers are included as serialized Python objects. Also, full PostgreSQL database of accident records, clusters and OSM features is included to allow generation of new datasets. Python scripts allow operations over datasets and classifiers training to visualize results and to replicate the experiments.

Requirements

- Python3	(version used Python 3.6)
- PostgreSQL	(version used PostgreSQL 9.5)

Organization of appended CD

- data – directory contains serialized objects of preprocessed data for training classifiers on various datasets
- database – contains SQL script to restore full PostgreSQL database
- tested_sites – contains classified sites in experiments URB-C-7 and URB-B-7. CSV consists of records of following structure: `gps_x, gps_y, classification(TP,TN,FP,FN)`
- src – contains Python scripts used for obtaining datasets (OSM), preprocessing methods, learning classifiers and evaluating results
- This paper in digital form (PDF and Microsoft Word)

Implementation remarks

In file `data_preprocessing.py`, methods for creating and preprocessing datasets are implemented. `data_postprocessing.py` contains methods for saving experiment entries and calculating feature importances. `OSM.py` contains methods for OSM data retrieval, preprocessing and persisting preprocessed data to PostgreSQL database. `clustering.py` and `map_splitting.py` is used to generate clusters of accidents from accident records based on accidents location.

Results of experiments analyzing accidents severity can be replicated using `model_run.py` script. Script contains parameters for selection desired datasets and classification parameters. Python library `scikit-learn` and `xgboost` is used. Results of experiments analyzing hazardous sites can be replicated using

H2O.py script. Script contains parameters for selection desired datasets and classification parameters. A machine learning framework H2O is used. Methods to visualize various results of the experiments are located in script plotting.py.

Database

Included SQL scripts restore full database designed for the traffic analysis. Database consist various definition and data tables. Remarks: table crashdataset contains raw UTVM data (context and context-free features). Table cds_to_clusters_cr defines relation between accidents and clusters. Table clusters_cr contains defined sites locations created by clustering method with occurrences of nearby OSM elements. Table out_experiments2 represents experiment runs, parameters and results. Also, database includes various views used for creating datasets.

Analysis	Dataset / Experiment	Source code notation
Severity	Context	v1.2
Severity	Context	v1.21*
Severity	Context-free	v2.0
Severity	Context enriched with OSM	v3.0
Severity	Context-free enriched with OSM	v20.1
Hazardous Sites	CZE-C-4	v5.0
Hazardous Sites	PRG-B-5	v6.0
Hazardous Sites	URB-B-5	v7.0
Hazardous Sites	URB-C-5	v8.0
Hazardous Sites	URB-C-7	v9.0
Hazardous Sites	URB-B-7	v10.0

Table 9: Dataset notation is source codes. *v1.21 accidents are considered severe if at least an incapacitating injury is suffered