

МАКСИМАЛЬНАЯ ОБЕСПЕЧИВАЕМАЯ СКОРОСТЬ КАК КЛЮЧЕВОЙ ПОКАЗАТЕЛЬ ТРАВМАТИЗМА ПРИ ДТП. СПОСОБЫ СБОРА, ОБРАБОТКИ И АНАЛИЗА

Пугачев Игорь Николаевич,
Хабаровский Федеральный исследовательский центр
Дальневосточного отделения Российской академии наук,
г. Хабаровск, Россия, ipugachev64@mail.ru

Евтюков Станислав Сергеевич,
Санкт-Петербургский государственный архитектурно-
строительный университет, г. Санкт-Петербург, Россия,
tm@spbgasu.ru

Шешера Николай Геннадьевич,
Дальневосточного юридического института МВД России
г. Хабаровск, Россия, kolyaka239@mail.ru

Григоров Денис Евгеньевич,
Дальневосточного юридического института МВД России,
г. Хабаровск, Россия, glowfisch8lan@gmail.com

DOI: 10.36724/2072-8735-2024-18-6-46-54

Manuscript received 07 May 2024;
Accepted 03 June 2024

Ключевые слова: массив данных, максимальная
обеспечиваемая скорость, травматизм,
координатные точки систем глобального
позиционирования Земли

При оценке безопасности дорожного движения необходимо учитывать множество факторов в совокупности. Одни из них являются статическими – такие как геометрические элементы дорог, которые не изменяются на протяжении длительного количества времени, другие динамические – погодные условия, интенсивность и скорость транспортного потока [1]. Последний значительно влияет не только на возникновение ДТП, но и на травматизм при нем. Данный показатель характеризуется способностью водителя принимать экстренные меры в узком коридоре времени и чем выше скорость движения, тем времени меньше. Иллюзия полного контроля за дорожной обстановкой наблюдается на прямых участках с неограниченной видимостью без нарушений поверхности дорожного полотна. Водитель перестает чувствовать скорость своего автомобиля и внезапно появившейся другой участник дорожного движения вынуждает применять экстренные меры, что в свою очередь сопровождается потерей контроля управления в результате заноса, увеличением тормозного пути, опрокидыванием и т.п. Чем выше скорость, тем серьезнее тяжесть последствий. Представление о скорости для большинства исследователей в соответствующей области сводятся к значениям максимальной разрешенной. Правила дорожного движения Российской Федерации принимаются как константа т.к. при ДТП определить нарушение виновнику – превышение скоростного режима не представляется возможным. Из-за этого отсутствует объективная картина ДТП. Соответственно, исследования в данной области требуют глубокого научного погружения и специализированного оборудования [2]. В работе авторами предлагается для оценки травматизма при ДТП включить в совокупность независимых переменных новый показатель УДС – максимальную обеспечиваемую скорость. Создана методика получения данного показателя с высокой точностью и минимальными затратами. Разработан алгоритм соединения данных с учетом локационного и временного логического соответствия.

Информация об авторах:

Пугачев Игорь Николаевич, заместитель директора по научной работе ХФИЦ ДВО РАН, д.т.н., доцент, Хабаровский Федеральный исследовательский центр Дальневосточного отделения Российской академии наук (ХФИЦ ДВО РАН), г. Хабаровск, Россия

Евтюков Станислав Сергеевич, Автомобильно-дорожный факультет, заведующий кафедры транспортных систем, д.т.н., доцент, Санкт-Петербургский государственный архитектурно-строительный университет, г. Санкт-Петербург, Россия

Шешера Николай Геннадьевич, доцент кафедры информационного и технического обеспечения ОВД к.т.н. Дальневосточного юридического института МВД России, г. Хабаровск, Россия

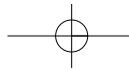
Григоров Денис Евгеньевич, Начальник кабинета специальных дисциплин кафедры информационного и технического обеспечения ОВД, Дальневосточного юридического института МВД России, г. Хабаровск, Россия

Для цитирования:

Пугачев И.Н., Евтюков С.С., Шешера Н.Г., Григоров Д.Е. Максимальная обеспечиваемая скорость как ключевой показатель травматизма при ДТП. Способы сбора, обработки и анализа // Т-Сотм: Телекоммуникации и транспорт. 2024. Том 18. №6. С. 46-54.

For citation:

Pugachev I.N., Evtukov S.S., Sheshera N.G., Grigorov D.E. (2024) Maximum speed as a key indicator of injuries in road accidents. Methods of collection, processing and analysis. T-Comm, vol. 18, no. 6, pp. 46-54. (in Russian)



ТРАНСПОРТ

Введение

При исследовании аварийности рассматривают обстоятельства каждого конкретного случая ДТП и скорость являясь одним из ключевых факторов анализа. Данный показатель принимают с учетом установленного на рассматриваемом участке скоростного режима, предполагая, что участники ДТП не нарушали действующих правил ПДД РФ. Основываясь на опыте, можно сделать вывод о том, что в большинстве случаев это не так. К тому же, в Российской Федерации наказание за нарушение скоростного режима устанавливается при превышении предельной нормы на 20 км/час. При ограничении скорости 60 км/час можно остаться безнаказанным двигаясь со скоростью 79 км/час [3]. Зная это, многие автолюбители пренебрегают личной безопасностью и увеличивают скорость до предельно допустимой, и выше. И наоборот, систематически припаркованные автомобили у края проезжей части не позволяют развивать необходимую скорость из-за риска столкновения основному потоку.

Самые трудно восполняемые потери от ДТП – это физические (травмы и смерть). Ежегодно в России от ДТП погибает около 30 тыс. чел. и порядка 250 тыс. получают ранения. По оценке специалистов ежегодный экономический ущерб от дорожных происшествий составляет порядка 9 % валового национального продукта страны [4].

Нет ничего дороже жизни и здоровья человека. Именно поэтому работа направлена на уменьшение травматизма и смертности на дорогах, путем совершенствования методов и средств оценки безопасности дорожного движения при проектировании и реконструкции транспортных сооружений, или своевременному изменению существующих режимов движения с учетом прогноза совокупности негативных динамических характеристик. Работа актуальна с научной и практической точек зрения.

Сбор данных

Для объективной оценки безопасности дорожного движения, ко всем исследуемым характеристикам, влияющим на аварийность, следует рассматривать и максимальную обеспечиваемую скорость на месте ДТП. Это можно сделать специализированными устройствами измерителями скорости, работающих с использованием эффекта Доплера, лазера [5]. Такой подход точен и эффективен если есть возможность вести скрытое наблюдение. В противном случае водители снижают скорость движения своего автомобиля принимая исследователя за сотрудника ГИБДД.

Второй способ заключается в использовании автомобиля-лаборатории. Дорогой, но эффективный [6]. Альтернативой ему может послужить GRS трекер, перевозимый в активном состоянии. Он представляет собой устройство, оснащенное GPS-приемником, которое используется для определения местоположения объекта в реальном времени, отправки собранных данных о координатах, скорости движения и направлении на специальный сервер или приложение с помощью сети связи (например, с использованием сотовой сети или Интернета).

Современные смартфоны обычно оснащены встроенными GPS-модулями, которые позволяют им определять свое местоположение с высокой точностью. Пользователи могут использовать карты навигационных приложений для

построения маршрутов, нахождения мест и ориентации на местности, отслеживать местоположение смартфона в режиме реального времени [7, 8].

С учетом отсутствия профильного оборудования для получения данных о максимальных обеспечиваемых скоростях исследуемых участков дорог было решено использовать приложение для смартфона Huawei Health, разработанное компанией Huawei. Основное назначение программы – помощь пользователю в личных тренировочных процессах, в том числе отслеживание скорости передвижения при ходьбе, беге и езде на велосипеде. Построение и хранение маршрутов передвижения (треков) для анализа достижений.

GPS трекер в рабочем состоянии получает сигналы от спутников GPS и определяет текущие координаты местоположения объекта, сохраняет их во внутренней памяти устройства в файл gpx, tpx или kml, с определенным интервалом времени [9]. Каждый записываемый сигнал имеет три основных параметра – широта, долгота и время. Файл-трек формируется с определенным синтаксисом делая его интерпретируемым для большинства геоинформационных систем (рис. 1, 2).

```
<?xml version="1.0" encoding="UTF-8"?><gpx creator="Health" xsi:schemaLocation="http://www.topografix.com/GPX/1/0 http://www.topografix.com/GPX/1/0/gpx.xsd" version="1.0" xmlns="http://www.topografix.com/GPX/1/0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><metadata><time>2022-10-31T07:12:13.954Z</time></metadata><trk><type>BeLociMed</type><extensions><totalTime>1261.0</totalTime><cumulativeDecrease>0.0</cumulativeDecrease><cumulativeClimb>0.0</cumulativeClimb><totalDistance>7838.0</totalDistance></extensions><trkseg><trkpt lat="48.474985" lon="135.080465"><time>2022-10-31T07:12:13.954Z</time></trkpt><trkpt lat="48.475065" lon="135.080465"><time>2022-10-31T07:12:14.954Z</time></trkpt><trkpt lat="48.47516" lon="135.080346"><time>2022-10-31T07:12:15.954Z</time></trkpt><trkpt lat="48.475274" lon="135.080166"><time>2022-10-31T07:12:16.000Z</time></trkpt><trkpt lat="48.475421" lon="135.079929"><time>2022-10-31T07:12:21.000Z</time></trkpt><trkpt lat="48.475578" lon="135.079721"><time>2022-10-31T07:12:23.000Z</time></trkpt><trkpt lat="48.475733" lon="135.079504"><time>2022-10-31T07:12:25.000Z</time></trkpt><trkpt lat="48.475885" lon="135.079289"><time>2022-10-31T07:12:27.000Z</time></trkpt><trkpt lat="48.476036" lon="135.079087"><time>2022-10-31T07:12:29.000Z</time></trkpt><trkpt lat="48.476197" lon="135.078839"><time>2022-10-31T07:12:31.000Z</time></trkpt><trkpt lat="48.476369" lon="135.0786"><time>2022-10-31T07:12:33.000Z</time></trkpt><trkpt lat="48.476522" lon="135.078358"><time>2022-10-31T07:12:35.000Z</time></trkpt><trkpt lat="48.476739" lon="135.07811"><time>2022-10-31T07:12:37.000Z</time></trkpt><trkpt lat="48.476926" lon="135.077851"><time>2022-10-31T07:12:39.000Z</time></trkpt><trkpt lat="48.477108" lon="135.077598">
```

Рис. 1. Структура записи трека с использованием GPS модуля с расширением gpx

Контрольная координатная точка

Маршрут движения GPS трекера

Рис. 2. Чтение трека с использованием геоинформационной системы, пример работы

Измерив расстояния между ближайшими контрольными координатными точками и поделив его на разницу во времени получения данных будет найдена скорость [10].

Для определения максимальной обеспечиваемой скорости на рассматриваемых дорогах с определенной периодичностью осуществлялись опытные заезды на автомобильном транспорте с включенным приложением Huawei Health, после чего формировался трек, который сохранялся в файл с расширением gpx. Исследования проводились два года, сформировалось большое количество треков (файлов).



ТРАНСПОРТ

Для соединения данных с целью аналитической обработки была использована среда разработки PyCharm. В одном виртуальном окружении созданы 3 исполняемых файла (модуля): speed_handler.py, geopoint_mapper.py, concat_dtp_speed.py и 1 для последовательного их запуска – start.py (рис. 3).

```

1 from handlers.speed_handler import main as speed
2 from handlers.geopoint_mapper import main as map_geocode
3 from handlers.concat_dtp_speed import main as dtp_speed_concat
4 from handlers.concat_dtp_weather import main as dtp_weather_concat
5
6 import warnings
7 warnings.filterwarnings("ignore")
8
9 if __name__ == '__main__':
10 speed() #speed gpx to csv
11 map_geocode() #geolocate dtp streets and speed streets
12 dtp_speed_concat() #mapping dtp
    
```

Рис. 3. Корневой модуль start.py, для последовательного вызова команд

Группировка треков в формате gpx в один файл csv

Все треки в формате gpx которые представляют из себя опытные заезды по исследуемому участку в разные периоды времени размещаются в одну директорию.

Первым вызывается модуль speed_handler.py, в его задачу входит группировка данных с директории files/gpx в файл speed_handled.csv (который расположен в директории files/csv) и рассчитать скорость по значениям двух ближайших координатных точек и разницы во времени между их записями в трек (рис. 4). Предварительно были импортированы функциональные библиотеки.

```

2 import logging
3 import numpy as np
4 from alive_progress import alive_bar
5 from lxml import etree
6 import pandas as pd
7 import os
8 import re
9 # Путь к папке, содержащей файлы GPX
10 folder_path = 'files/gpx'
11 # Конечный файл для записи
12 output_file_path = 'files/csv/speed_handled.csv'
13 df_combined = pd.DataFrame()
14 file_list = os.listdir(folder_path)
    
```

Рис. 4. Модуль speed_handler.py, предварительные настройки, определение переменных

Библиотеки использованы для упрощения разработки и расширения функциональности программ. В модуле speed_handler были установлены:

- logging – ведение журнала (логирования) сообщений;
- numpy – работа с многомерными массивами и математическими функциями;
- alive_progress – отображение прогресса выполнения операций в терминале. Позволяет создавать стильные и анимированные индикаторы выполнения задач;
- lxml.etree – обработка и анализ xml и html;
- pandas – обработка и анализ данных;
- os – работа с операционной системой, включая взаимодействие с файловой системой;
- re – работа с регулярными выражениями.

Создается пустой DataFrame df_combined для объединения данных из файлов gpx (рис. 4, строчка 13). Функция os.listdir(folder_path) из модуля os используется для получения списка файлов и папок (имен файлов), содержащихся в указанной директории (folder_path) (рис. 4, строчка 14).

Функция haversine_distance вычисляет расстояние между двумя точками на Земле, используя формулу о сферических расстояниях – haversine formula. Для определения единицы измерения скорости (км/час) используется средний радиус Земли R = 6371 (рис. 5).

```

17 def haversine_distance(lat1, lon1, lat2, lon2):
18     R = 6371
19
20     lat1_rad = np.radians(lat1)
21     lon1_rad = np.radians(lon1)
22     lat2_rad = np.radians(lat2)
23     lon2_rad = np.radians(lon2)
24     delta_lat = lat2_rad - lat1_rad
25     delta_lon = lon2_rad - lon1_rad
26     a = (np.sin(delta_lat / 2) ** 2 + np.cos(lat1_rad) *
27         np.cos(lat2_rad) * np.sin(delta_lon / 2) ** 2)
28     c = 2 * np.arctan2(np.sqrt(a), np.sqrt(1 - a))
29     distance = R * c
30     return distance
    
```

Рис. 5. Модуль speed_handler.py, функция haversine_distance. Расчет расстояния между координатными точками

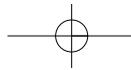
На строчках 32-35 рисунка 5 конвертируется широта и долгота точек из градусов в радианы, поскольку тригонометрические функции в библиотеке numpy работают с радианами, а на строчках 36-37 вычисляется разница между широтой и долготой точек.

Рассчитывается расстояние между точками на поверхности сферы (поверхности Земли) по широте и долготе, по формуле haversine (формула полусинуса) (рис. 5, строки 38-41).

Ниже приведен код функции main, который включает в себя основной процесс обработки файлов с расширением gpx из указанной директории, выполняется полный цикл извлечения и анализа данных о маршрутах, вычисления расстояний и скоростей, а также сохранения результатов в удобном формате для дальнейшего использования:

```

def main():
    global df_combined
    with alive_bar(len(file_list), force_tty=True, title='Speed GPX to CSV') as bar:
        for file_name in file_list:
            if file_name.endswith('.gpx'):
                file_path = os.path.join(folder_path, file_name)
                try:
                    tree = etree.parse(file_path)
                    root = tree.getroot()
                    namespace = root.nsmap
                    tracks = []
                    for trk in root.iterfind('./{%s}trk' % namespace[None]):
                        track = {}
                        track['latitude'] = []
                        track['longitude'] = []
                        track['elevation'] = []
                        track['time'] = []
                        for trkpt in trk.iterfind('./{%s}trkpt' % namespace[None]):
    
```



```

track['latitude'].append(float(trkpt.get('lat')))
track['longitude'].append(float(trkpt.get('lon')))
elevation_element = trkpt.find('./{%s}ele' %
namespace[None])
if elevation_element is not None:
    track['elevation'].append(float(elevation_ele-
ment.text))
else:
    track['elevation'].append(None)
time_element = trkpt.find('./{%s}time' %
namespace[None])
if time_element is not None:
    time_element_text = time_element.text
    try:
        regex_pattern = r'\b(20[01][0-9])\b'
        replaced_text = re.sub(regex_pattern, r'2022',
time_element_text)
        track['time'].append(pd.to_datetime(re-
placed_text))
    except Exception:
        track['time'].append(None)
    else:
        track['time'].append(None)
tracks.append(track)
if len(tracks) > 0:
    df = pd.DataFrame(tracks[0])
    df['date_time'] = df['time']
    df['distance'] = haversine_distance(df['latitude'].shift(-1),
df['longitude'].shift(-1),
                                df['latitude'], df['longitude'])
    df['time_diff'] = (df['time'] - df['time'].shift()).shift(-1)
    df['time_diff'] = df['time_diff'].dt.total_seconds() / 3600
    df['speed'] = df['distance'] / df['time_diff']
    df_combined = pd.concat([df_combined, df
        ['latitude', 'longitude', 'elevation', 'time', 'date_time',
'distance', 'time_diff',
        'speed']]], ignore_index=True)
except Exception as e:
    logging.error(f'Ошибка обработки файла {file_name}:
{e}')
bar()
df_combined.to_csv(output_file_path, index=False)
print("Обработка файлов GPX и запись в один файл завершена.")

```

Цикл `for file_name in file_list` позволяет итерировать по списку файлы в указанной папке. Каждый файл проверяется на наличие расширения `gpx` с помощью `if file_name.endswith('.gpx')`.

Для каждого файла `gpx` происходит следующее:

- формируется полный путь к файлу с использованием `file_path = os.path.join(folder_path, file_name)`;
- файл структурируется с помощью `etree.parse(file_path)` для дальнейшей обработки;
- извлекается информация о координатах, высоте и времени всех точек маршрута из `xml`-структуры.

Для каждого файла создается список словарей `tracks`, содержащий данные о координатах, высоте и времени точек маршрута, которые добавляются в `DataFrame df`, вместе со столбцами даты и времени, расстоянием между точками, разницей во времени и скоростью движения.

Новые данные из каждого файла добавляются в общий `DataFrame df_combined` с использованием метода `pd.concat`. В случае возникновения ошибок при обработке файла, сообщение о ней логируется с использованием `logging.error`.

ТРАНСПОРТ

После обработки всех файлов, данные из общего `DataFrame df_combined` записываются в `csv`-файл с именем `output_file_path` с помощью `df_combined.to_csv`. Двухмерный массив файла имеет ключи: `latitude`, `longitude`, `elevation`, `time`, `date_time`, `distance`, `time_diff`, `speed`. Выводится сообщение «Обработка файлов GPX и запись в один файл завершена».

Определение единой системы наименований улиц для группировки массивов данных

Цель исследования максимальной обеспечиваемой скорости – влияние данного показателя на травматизм при ДТП. С открытых источников ГИБДД УМВД России по г. Хабаровску были собраны данные об аварийности на рассматриваемых участках дорог с 2016-2021 г. Свыше 75 тыс. ДТП с пострадавшими и без. Сформированные данные записаны в файл `dtп.csv`.

После окончания работы модуля `speed_handle.py` запускается `geoint_mapper.py`. В его задачу входит подготовка двух массивов данных: о ДТП (`dtп.csv`) и максимальной обеспечиваемой скорости (`speed_handled.csv`), для корректного соединения в логическом локационном и временном соответствии.

После установки необходимых библиотек определяются адреса с расположениями файлов переменным `filepath_speed` и `filepath_dtп` (рис. 6).

```

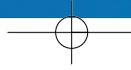
8 import os
9 from time import sleep
10 import pandas as pd
11 from datetime import timedelta
12 from alive_progress import alive_bar
13 from dateutil.tz import tzutc
14 from dateutil.parser import parse
15 import hashlib
16 import math
17 from geopy.geocoders import Nominatim
18 filepath_speed = './files/csv/speed_handled.csv'
19 filepath_dtп = './files/csv/dtp.csv'
20 geolocator = Nominatim(user_agent="Mozilla/5.0 (platform; rv:geckoversion) "
21                        "Gecko/geckotrail Firefox/firefoxversion",
22                        timeout=100)

```

Рис. 6. Модуль `geoint_mapper.py`, подготовка программы, определение адресов переменным

К уже известным библиотекам добавлены:

- `time.sleep` – создание задержки в выполнении цикла. Эта мера необходима для исключения перезагрузки сервера геору запросами, из-за чего вычислительному устройству блокируется доступ к ресурсу по `ip` адресу;
- `datetime.timedelta` – работа с временными интервалами;
- `dateutil.tz`, `dateutil.parser` – обработка и работа с временными данными;
- `hashlib` – генерация хэшей данных. Запросы на сервер геору посылались циклически непрерывно в течении более 20 дней, результаты ответов записывались в оперативную память вычислительного устройства, с последним запросом формируется база данных, которая сохранялась в `csv` файл. В случае сбойной ситуации приходилось повторять заново циклическую функцию продолжительное количество времени. Данная библиотека позволяет сохранять хэш каждого запроса и в случае прерывания цикла продолжить работу с места останова;
- `math` – математические функции и константы;



ТРАНСПОРТ

– `geopy.geocoders.Nominatim` – работа с геокодированием, перевода адресов в координаты и наоборот.

Последняя библиотека имеет ключевое значение в данном модуле. При соединении двух файлов было решено первой точной фильтрации установить наименование улиц. Для этого они должны иметь одинаковую запись (одинаковый набор символов). Например, если в файле `speed_handled.csv`, в строке записано «улица Ленина», то для корректного отбора ДТП с такими же улицами в файле `dtп.csv` должно быть также записано «улица Ленина». Сортировка не состоится в случае следующих записей: «Ленина», «ул. Ленина», «Ленина улица» и т.п. Для унификации данных было решено использовать геолокационную систему `geopy`. Каждый массив данных имеет координаты, которые последовательно посылаются на сервер `geopy`, в ответ приходит наименование улицы [10].

В файлах были созданы новые колонки – `street_geo`, в которые после ответа на запрос записывались улицы, соответствующие координатам.

Стоит отметить, что такой подход обоснован большим количеством данных. Соответственно, для исключения ошибок этот процесс был автоматизирован.

В модуле `geopoint_mapper.py` созданы три вспомогательные функции `_get_street`, `_date_utc` и `_parse_datetime` для реализации основной `main`.

Функция `_get_street` выполняет алгоритм по получению координат, широты (`lat`) и долготы (`lon`) точки на карте. Для каждой из координат (`lat` и `lon`) вычисляется хэш (MD5) в формате шестнадцатеричной строки (рис. 7, строки 25-26). Далее происходит соединение полученных сигнатур в одну строку, которая присваивается переменной `path` (рис. 7, строка 27). Если координаты не содержатся в базе, то сконкатенированная строка будет уникальной, тем самым обеспечивается возможность точно идентифицировать требуемую геолокационную точку. В свою очередь стоит отметить, что данный подход снижает время выполнения алгоритма, уменьшая его сложность за счет ускоренного ответа геокодирования.

```

24 def _get_street(lat, lon):
25     latHash = hashlib.md5(repr(lat).encode()).hexdigest()
26     lonHash = hashlib.md5(repr(lon).encode()).hexdigest()
27     path = f'./files/geopoints/{latHash}{lonHash}.txt'
28     if os.path.isfile(path):
29         f = open(path, "r", encoding='utf-8')
30         street = f.read()
31         location = 'cache'
32     else:
33         location = geolocator.reverse((lat, lon))
34         address = location.raw['address']
35         street = address.get('road', '')
36         location = 'internet'
37         print('Download from Internet...')
38         with open(path, 'w', encoding='utf-8') as f:
39             f.write(street)
40             f.close()
41     return street
    
```

Рис. 7. Модуль `geopoint_mapper.py`, функция `_get_street` – работа с хэш

На строчках 28-40 рисунка 7 выполняется условие проверки существования файла с соответствующим запросу именем. Если условие удовлетворяется, то его содержимое (название улицы) считывается. В этом случае переменная `location` принимает значение `cache`. Если файл не найден, производится обратное геокодирование для получения информации о местоположении по заданным координатам. Затем из полученного объекта `location` извлекается адрес и отдельно название улицы. Также в этом случае переменная `location` принимает значение `internet`. При этом, информация о найденной улице записывается в файл, после чего его содержимое возвращается в качестве результата функции.

Код, представленный в функции `_date_utc`, использует библиотеку `dateutil` для преобразования строки даты и времени в объект `datetime` в часовом поясе UTC (рис. 8, строки 43-53), а в `_parse_datetime` анализируется строка с датой и временем для представления в объект `datetime` с максимальным согласованием (`fuzzy matching`). Если парсинг не удался, функция возвращает значение `None` (рис. 8, строки 54-68).

```

43 def _date_utc(s):
44     > ...
53     return parse(s, tzinfos=tzutc)
54     1 usage
55 def _parse_datetime(s):
56     > ...
65     try:
66         return parse(s, fuzzy=True)
67     except ValueError:
68         return None
    
```

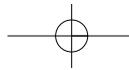
Рис. 8. Модуль `geopoint_mapper.py`, функции `_date_utc` и `_parse_datetime`. Преобразование даты и времени

Запускается функция `main`, в которой происходит инициализация прогресс-бара в строке 72 рисунка 9. Это необходимо для отслеживания процесса вычислений и получения данных. На строчках 73-74 читаются данные из двух `csv`-файлов `filepath_dtp` и `filepath_speed` при помощи библиотеки `pandas` и сохраняются в соответствующие `dataframe` `df_dtp` и `df_speed`. Преобразуется столбец `date_time` в `dataframe` `df_speed`, в объекты `datetime` с помощью функции `_parse_datetime` и добавляется к ним 10 часов (`timedelta(hours=10)`) (рис. 9, строка 76). Таким образом переводится время по Гринвичу на местный часовой пояс.

```

def main():
    with alive_bar(total=3, force_tty=True, title='Geopoint Mapper') as bar:
        df_dtp = pd.read_csv(filepath_dtp, sep=',')
        df_speed = pd.read_csv(filepath_speed, sep=',', date_parser=_date_utc)
        df_speed['date_time'] = df_speed['date_time'].apply(_parse_datetime) + timedelta(hours=10)
        df_dtp[['latitude2', 'longitude2']] = df_dtp['closest_point'].str.split(',', expand=True)
        df_dtp['latitude2'] = df_dtp['latitude2'].str.replace(' ', '').astype(float)
        df_dtp['longitude2'] = df_dtp['longitude2'].str.replace(' ', '').astype(float)
        streets_speed = df_speed.apply(lambda row: _get_street(row['latitude'], row['longitude']) if not (
            math.isnan(row['latitude']) or math.isnan(row['longitude'])) else '', axis=1)
        bar()
        streets_dtp = df_dtp.apply(lambda row: _get_street(row['latitude2'], row['longitude2']) if not (
            math.isnan(row['latitude2']) or math.isnan(row['longitude2'])) else '', axis=1)
        bar()
        df_dtp['street_geo'] = streets_dtp
        df_speed['street_geo'] = streets_speed
        df_dtp.to_csv(path_or_buf='./files/csv/dtp_with_streets.csv', index=False, sep=',', encoding='UTF-8-SIG')
        df_speed.to_csv(path_or_buf='./files/csv/speed_with_streets.csv', index=False, sep=',', encoding='UTF-8-SIG')
        bar()
    
```

Рис. 9. Модуль `geopoint_mapper.py`, функция `_date_utc`



ТРАНСПОРТ

Стоит отметить, что данные о ДТП предварительно наполнились координатными точками house (координата дома) и closest_point1 (ближайшая координатная точка соответствующей дому дороге в базе дорожных координат геору). Эти действия были направлены для привязки его к дороге координатными точками, т.к. при регистрации аварии корректно указывается только адрес (город, улица, дом). Этот процесс выходит за рамки данного исследования, поэтому реализацией данного преобразования массива пренебрегается.

В строчках 77-79 рисунка 9 столбец closest_point1 в dataframe df_dtp разделяется на два новых столбца latitude2 и longitude2, после чего значения столбцов преобразуются в числовой тип. Для каждого массива данных (df_speed и df_dtp) выполняется операция по присвоению улицы (street) по координатам с использованием описанной выше функции _get_street, которые записываются в новые столбцы street_geo (рис. 9, строчки 81-88).

В результате работы алгоритма модуля geopoint_mapper.py формируются новые файлы, dtp_with_streets.csv и speed_with_streets.csv (рис. 9, строчки 90-91), в которых имеются однородные значения улиц (в столбцах street_geo, необходимых для фильтрации и последующего соединения данных).

Соединение массивов данных в соответствии с локацией и временем

После завершения работы модуля geopoint_mapper.py запускается concat_dtp_speed.py. В его задачу входит сравнение файлов и логического соединения данных в них по значениям ключевых признаков (времени и места).

В модуль были импортированы необходимые для работы библиотеки (рис. 10). К уже известным стоит выделить – ast, для преобразования кода в структурированные данные и дальнейшего анализа.

```

1 import pandas
2 import pandas as pd
3 from alive_progress import alive_bar
4 import ast
5 from math import radians, sin, cos, sqrt, atan2
6 import math
7 from dateutil import parser
8 import warnings
9 warnings.filterwarnings("ignore")
    
```

Рис. 10. Модуль concat_dtp_speed.py, установка зависимостей

Перед началом обработки данных для снижения расчетной нагрузки создан список улиц г. Хабаровска, которыми было решено ограничить исследования. На них установлены рубежи контроля автомобилей системы ИНТЕГРО КДД (<https://www.integra-s.ru/>), в следствии чего была рассчитана интенсивность транспортного потока, которая является предметом изучения других работ. Тем не менее, в последующих научных трудах максимальная обеспечиваемая скорость будет исследоваться в совокупности с данным показателем, поэтому выборку из массива было решено подготовить универсальной.

Список улиц был передан переменной streets (рис. 11).

```

11 streets = ['улица Ленина', 'улица Серышева', 'улица Ленинградская', 'улица Синельникова',
12 'Уссурийский бульвар', 'улица Муравьева-Амурского', 'Амурский бульвар',
13 'улица Карла Маркса', 'Матвеевское шоссе', 'улица Ганарника', 'улица Шевченко',
14 'улица Тургенева', 'улица Комсомольская', 'улица Истомина', 'улица Калинина',
15 'улица Фрунзе', 'улица Запарина', 'улица Дзержинского', 'улица Волочаевская',
16 'улица Шеронова', 'улица Гоголя', 'улица Пушкина', 'улица Ким Ю Чена',
17 'улица Дикопольцева', 'улица Московская', 'улица улица Большая', 'улица Промышленная',
18 'улица Выборгская', 'Чернореченское шоссе', 'улица Воронежская', 'улица Шелеста',
19 'Студенческий переулок', 'Ленинградский переулок']
    
```

Рис. 11. Список улиц в переменной streets

После списка улиц в модуле concat_dtp_speed.py написан код вспомогательной функции find_closest_point для определения ближайших точек к дому на заданной дороге и последующего возвращения их в виде списка closest_points. Она принимает два аргумента: house (координаты дома) и road (список координат точек на дороге) (рис. 12).

```

26 def find_closest_points(house, road):
27     closest_points = [None, None]
28     min_distances = [float('inf'), float('inf')]
29     road1 = road.copy()
30     for point in road1:
31         distance = math.sqrt((point[0] - house[0])**2 + (point[1] - house[1])**2)
32         if distance < min_distances[1]:
33             if distance < min_distances[0]:
34                 min_distances[1] = min_distances[0]
35                 min_distances[0] = distance
36                 closest_points[1] = closest_points[0]
37                 closest_points[0] = point
38             else:
39                 min_distances[1] = distance
40                 closest_points[1] = point
    
```

Рис. 12. Функция find_closest_points модуля concat_dtp_speed.py

В функции find_closest_points создается список closest_points с двумя элементами None, который будет содержать ближайшую точку к дому и два списка min_distances с изначальными значениями бесконечности (рис. 12, строчки 27-28), но с перспективой заполнения минимальными расстояниями от дома до точек на дороге. Также создается копия списка road под именем road1 (рис. 12, строчка 29).

Для каждой точки на дороге (point) рассчитывается расстояние до дома с помощью формулы евклидова расстояния (рис. 12, строчки 30-40). Определяется минимальное значение и на выходе функция возвращает список ближайших точек closest_points.

Главной функцией модуля concat_dtp_speed.py является main, код приведен ниже:

```

def main():
    df_dtp = pd.read_csv('./files/csv/dtp_with_streets.csv', sep=';')
    df_speed = pd.read_csv('./files/csv/speed_with_streets.csv',
    sep=';')
    sorted_streets = sorted(streets)
    df_dtp = df_dtp[df_dtp["street_geo"].isin(sorted_streets)]
    df_speed = df_speed[df_speed["street_geo"].isin(sorted_streets)]
    ndexport = []
    column_names = []
    column_names.append(df_dtp.columns.tolist())
    column_names[0].append('datetime')
    column_names[0].append('max_speed')
    with alive_bar(len(df_dtp), force_tty=True, title='Concat DTP
    Speed') as bar:
        for index, row_dtp in df_dtp.iterrows():
            try:
                df_speed_filtered = df_speed[df_speed["street_geo"] =
    row_dtp[
                "street_geo"]].copy()
    
```



ТРАНСПОРТ

```

df = df_speed_filtered.apply(lambda row: [row['latitude'],
row['longitude']], axis=1)
ndarray = df.values
house = ast.literal_eval(row_dtp['house'])
df_speed_filtered[
    'latitude_longitude'] = df
closest_points = find_closest_points(house, ndarray.tolist())
def distance(lat1, lon1, lat2, lon2):
    R = 6371000 # радиус Земли в метрах
    phi1 = radians(lat1)
    phi2 = radians(lat2)
    delta_phi = radians(lat2 - lat1)
    delta_lambda = radians(lon2 - lon1)
    a = sin(delta_phi / 2) * sin(delta_phi / 2) + cos(phi1) *
cos(phi2) * sin(delta_lambda / 2) * sin(
    delta_lambda / 2)
    c = 2 * atan2(sqrt(a), sqrt(1 - a))
    return R * c
closest_point_lat, closest_point_lon = closest_points[0]
radius = 50 # радиус в метрах
points_within_radius = []
for point in df:
    point_lat, point_lon = point
    dist = distance(closest_point_lat, closest_point_lon,
point_lat, point_lon)
    if dist <= radius:
        points_within_radius.append(point)
sorted_streets_speed = sorted(
    points_within_radius)
df_speed_filtered1 = df_speed_filtered[
    df_speed_filtered["latitude_longitude"].isin(sorted_streets_speed)]
row_dtp['datetime'] = parser.parse(row_dtp['Дата'])
def find_nearest_month(target_month, months):
    nearest_month = sorted(months, key=lambda x:
min(abs(x - target_month), 12 - abs(x - target_month)))[1]
    return nearest_month
df_speed_filtered1['datetime'] = df_speed_filtered1['date_time'].apply(lambda x: parser.parse(x))
available_months = df_speed_filtered1['datetime'].dt.strftime('%m').unique().astype(int)
nearest_month = find_nearest_month(row_dtp['datetime'].month, available_months)
ndf = df_speed_filtered1[df_speed_filtered1['datetime'].dt.month == nearest_month.pop()][:10]
max_speed = ndf['speed'].max()
row_dtp['max_speed'] = max_speed
ndexport.append(row_dtp.tolist())
except Exception as err:
    print(index)
    print(err)
bar()
expdf = pandas.DataFrame(ndexport, columns=column_names)
expdf.to_csv('files/csv/dtp_speed_concatenated.csv', index=False,
sep=';', encoding="UTF-8-SIG")

```

Приведенная выше функция main сначала читает два файла, dtp_with_streets.csv и speed_with_streets.csv, и передает их в переменные df_dtp и df_speed. Сортирует улицы, которые находятся в df_dtp, и создает список sorted_streets. Фильтрует df_dtp и df_speed, оставляя только строки, где значения столбца street_geo соответствует хотя бы одному из элементов sorted_streets. Создает пустой список ndexport и добавляет названия столбцов из df_dtp в список column_names, затем добавляет столбцы datetime и max_speed.

Используя цикл for в функции main для итерации по строкам df_dtp выполняются следующие действия:

- по ближайшей точке на дороге к указанному дому вычисляются скоростные точки в радиусе 50 м на дороге в df_speed. Таким образом сортируются скоростные данные на основе полученных координат для уменьшения области обработки;
- определяется ближайший месяц в данных скорости для каждой даты ДТП и выбираются данные только для этого месяца;
- определяется максимальная скорость в отфильтрованных данных, которая добавляется в соответствующую строку ДТП;
- с каждой итерацией добавляется строка ДТП в новый DataFrame expdf.

На этом работа модуля concat_dtp_speed.py заканчивается. В результате выполнения функции main создается файл dtp_speed_concatenated.csv, в котором соединены два массива данных (о ДТП и максимальной обеспечиваемой скорости) по принципу максимального соответствия места и времени.

Оценка значимости максимальной обеспечиваемой скорости в совокупности с другими характеристиками УДС и погодно-климатическими условиями в оценке травматизма при ДТП

Анализ максимальной обеспечиваемой скорости стал частью большой работы по оценке травматизма при ДТП, соответственно, влияние данной независимой переменной на зависимую необходимо исследовать в структуре совокупного взаимодействия всех факторов, характеризующих аварийность [11, 12]. В их число вошли: время (time), опыт вождения с интервалом 5 лет (driving_experience_5), день недели (week_day), наличие светофора (traffic_light), интенсивность с интервалом 250 авт/час (int_250) и многие другие погодно-климатические факторы и характеристики УДС, в том числе – максимальная обеспечиваемая скорость (max_speed) (GitHub. Репозиторий glwofisch8lan/3691308f2a4c2f6983f2880d32e29c84).

Путем экспериментального подбора были отобраны самые эффективные признаки с использованием машинного обучения – Метода случайных деревьев, обучена прогностическая модель высокого качества (Accuracy =97.99, r2 = 0.8212487585413919), для исследования которой проанализирована значимость признаков с использованием метода перестановок (permutation importance) для модели случайного леса (random_forest) на обучающих данных (X_train и y_train).

В результате были получены коэффициенты значимости: [[0.1658, 'max_speed'], [0.0044, 'availability_of_a_pedestrian_crossing_within_a_radius_of_50_m'], [0.0041, 'part_of_the_day'], [0.0036, 'time'], [0.0036, 'driving_experience_5'], [0.0032, 'road_width_5'], [0.0032, 'longitudinal_slope'], [0.003, 'int_250'], [0.0025, 'vehicle_type_category'], [0.0019, 'atmospheric_pressure_at_station_level'], [0.0018, 'atmospheric_pressure_at_sea_level'], [0.0017, 'air_temperature'], [0.0016, 'relative_humidity'], [0.0014, 'soil_temperature'], [0.0014, 'distance_to_bus_stop_from_the_center_of_the_house'], [0.0013, 'number_of_forward_lanes_(total_number_in_all_directions)'], [0.0012, 'week_day'], [0.0012, 'distance_to_parking'], [0.0011, 'saturation_deficit'], [0.0011, 'dew_point_temperature'], [0.001, 'partial_pressure_of_water_vapor'], [0.001, 'left_stripe_length'], [0.0007, 'traffic_light'], [0.0007, 'intersection_type'], [0.0006,

['right_stripe_length'], [0.0005, 'wind_speed'], [0.0005, 'month'], [0.0004, 'visibility_code (VV)'], [0.0004, 'type_of_road'], [0.0004, 'number_of_left_lanes (in_all_directions: oncoming_and_parallel)'], [0.0003, 'precipitation_mm'], [0.0, 'traffic_direction_separator'], [0.0, 'number_of_right_lanes (in_all_directions)']]).

Из 34 отобранных признаков лучше всего влияет на травматизм при ДТП максимальная обеспечиваемая скорость (max_speed), коэффициент которой составил 0,1658. Это подтверждается и матрицей корреляции (рис. 13).

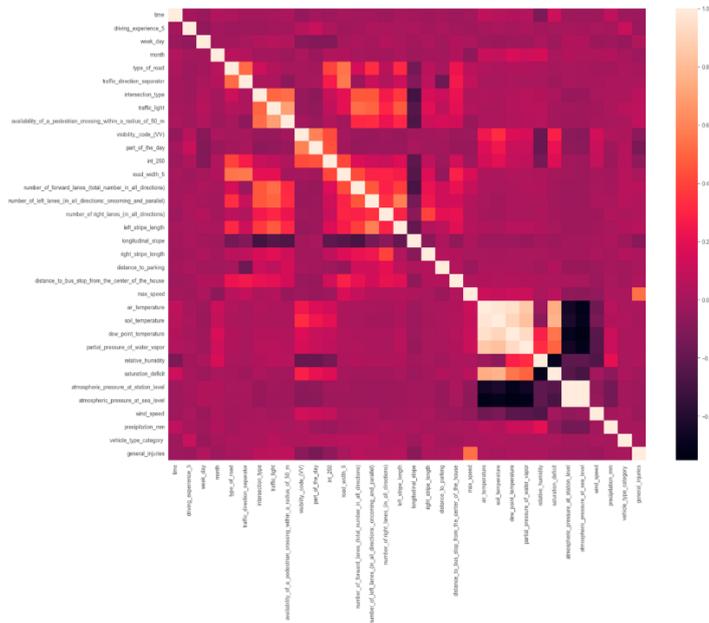


Рис. 13. Корреляционная матрица влияния признаков

Заключение

Скорость является важнейшим показателем для оценки травматизма при ДТП. Чем она выше, тем больше кинетическая энергия автомобиля, соответственно, повышается степень ранения людей, меньше времени остается водителю проанализировать возникшую на пути опасность и выполнить экстренные меры [13].

В данной работе определен важнейший показатель травматизма при ДТП – максимальная обеспечиваемая скорость. Он исследован с позиции системности более двух лет.

Для повышения точности при определении скорости движения опытного автомобиля использовался современный инструментальный систем глобального позиционирования, в том числе применена формула евклидова расстояния и полусинуса, которая учитывает сферическую поверхность Земли.

Разработаны алгоритмические инструментарии, основанные на полученных теоретических результатах и программных прототипах их основных компонентов, методов и алгоритмов аналитики для соединения массивов данных из разных информационных ресурсов.

Влияние максимальной обеспечиваемой скорости обоснована в совокупности с другими показателями травматизма

при ДТП. Она оценена как самая высокая из независимых переменных методом перестановок (permutation importance) для модели случайного леса (random_forest). Коэффициент значимости 0,1658 при точности модели 97,99 (GitHub. Репозиторий glowfish8lan/3691308f2a4c2f6983f2880d32e29c84).

Литература

1. Бабков В.Ф. Дорожные условия и безопасность движения. М.: Транспорт, 1993. 271 с.
2. Пугачев И.Н., Каменчуков А.В., Канский Д.В., Кот Е.Н., Буртыль Ю.В., Щеглов В.И. Управление безопасностью дорожной инфраструктуры. Хабаровск, 2022.
3. Эксплуатация автомобильных дорог: учебное пособие / сост.: И.Н. Пугачев, А.В. Каменчуков, Н.С. Нестерова. Хабаровск: Изд-во ДВГУПС, 2022. 168 с.
4. Пугачев И.Н., Скрипко П.Б., Шешера Н.Г. Программный подход к комплексному сбору и подготовки данных об интенсивности движения транспортных средств, погодных условий и естественной освещенности в часовых интервалах // Т-Comm: Телекоммуникации и транспорт. 2023. Том 17. №10. С. 43-51.
5. Руденко Н.В. Характеристики дорожного движения по автомобильным дорогам и методы оценки безопасности дорожного движения. Международный журнал информационных технологий и энергоэффективности. 2022. Т. 7. № 3-3 (25). С. 100-106.
6. Батищева О.М., Ганичев А.И., Старикова А.Г. Факторы, влияющие на обеспечение безопасности дорожного движения в крупных городах. Вестник транспорта Поволжья. 2023. № 2 (98). С. 110-115.
7. Бакланова К.В. Влияние параметров транспортного потока и характеристик дорог на безопасность дорожного движения. Интеллект. Инновации. Инвестиции. 2023. № 2. С. 99-110.
8. Кириллова Е.В., Бурдина Е.В. Анализ эффективности работы выделенных полос для общественного транспорта на основе GPS-трекеров бортового оборудования подвижного состава. Мир транспорта и технологических машин. 2023. № 1-1 (80). С. 34-42.
9. Гаврилов В.С. Логистика и управление на автомобильном транспорте В сборнике: Научно-технические аспекты развития автотранспортного комплекса 2023 // Материалы IX Международной научно-практической конференции (заочно-дистанционная) в рамках 9-го Международного научного форума Донецкой Народной Республики «Инновационные перспективы Донбасса: инфраструктурное и социально-экономическое развитие». Горловка, 2023. С. 250-253.
10. Арутюнян М.А. Разработка алгоритмического аппарата по обеспечению безопасности дорожного движения. Моделирование, оптимизация и информационные технологии. 2023. Т. 11. № 3 (42). С. 9-10.
11. Пушкарева Г.В., Халтурин Д.Д. Анализ методов повышения безопасности движения на нерегулируемых перекрестках крупных городов (на примере города Томска). Вестник Томского государственного архитектурно-строительного университета. 2023. Т. 25. № 5. С. 198-223.
12. Пугачев И.Н., Шешера Н.Г., Щеглов В.И. Анализ геометрических элементов дорог с помощью современных геоинформационных систем при оценке их аварийности // Вестник гражданских инженеров. 2021. № 3 (86). С. 127-133.
13. Пугачев И. Н., Шешера Н. Г. Применение методов статистического анализа для оценки параметров транспортных потоков и характеристик улично-дорожной сети // Дальневост. юрид. ин-т МВД России. Хабаровск: РИО ДВЮИ МВД России. 2020. 108 с.

MAXIMUM SPEED AS A KEY INDICATOR OF INJURIES IN ROAD ACCIDENTS. METHODS OF COLLECTION, PROCESSING AND ANALYSIS

Igor N. Pugachev, Khabarovsk Federal Research Center of the Far Eastern Branch of the Russian Academy of Sciences (HFRC FEB RAS), Russia, Khabarovsk, ipugachev64@mail.ru

Stanislav S. Evtukov, Petersburg State University of Architecture and Civil Engineering, St. Petersburg, Russia, tm@spbgasu.ru

Nikolai G. Sheshera, Far Eastern Law Institute of the Ministry of Internal Affairs of Russia, Russia, Khabarovsk, kolyaka239@mail.ru

Denis E. Grigorov, Far Eastern Law Institute of the Ministry of Internal Affairs of Russia, Russia, Khabarovsk, glowfisch8lan@gmail.com

Abstract

When assessing road safety, many factors must be taken into account together. Some of them are static – such as geometric elements of roads that do not change over a long period of time, others are dynamic – weather conditions, intensity and speed of traffic flow [1]. The latter significantly influences not only the occurrence of road accidents, but also the number of injuries caused by them. This indicator is characterized by the driver's ability to take emergency measures in a narrow time period and the higher the speed, the less time. The illusion of complete control over the road situation is observed on straight sections with unlimited visibility without disturbances to the road surface. The driver stops feeling the speed of his car and the sudden appearance of another road user forces him to take emergency measures, which in turn is accompanied by loss of control as a result of skidding, increased braking distance, rollover, etc. The higher the speed, the more serious the consequences. The idea of speed for most researchers in the relevant field is reduced to the maximum allowed values. The traffic rules of the Russian Federation are accepted as a constant because in case of an accident, it is not possible to determine the violation of the culprit – exceeding the speed limit. Because of this, there is no objective picture of the accident. Accordingly, research in this area requires deep scientific immersion and specialized equipment [2]. In the work, the authors propose to include in the set of independent variables a new indicator of road traffic speed – the maximum ensured speed – to assess injuries in road accidents. A method has been created for obtaining this indicator with high accuracy and minimal costs. An algorithm for connecting data has been developed taking into account locational and temporal logical correspondence.

Keywords: data array, maximum sustained speed, injuries, coordinate points of the Earth's global positioning systems.

References

1. V.F. Babkov. Road conditions and traffic safety. Moscow: Transport, 1993. 271 p.
2. I.N. Pugachev, A.V. Kamenchukov, D.V. Kapsky, E.N. Kot, Yu.V. Burtyl, V.I. Shcheglov. Road infrastructure safety management. Khabarovsk, 2022.
3. I.N. Pugachev, A.V. Kamenchukov, N.S. Nesterova. Operation of highways: textbook / comp.: Khabarovsk: Publishing house DVGUPS, 2022. 168 p.
4. I.N. Pugachev, P.B. Skripko, N.G. Sheshera. A software approach to the integrated collection and preparation of data on vehicle traffic intensity, weather conditions and natural light at hourly intervals. *T-Comm*. 2023. Vol. 17. No. 10, pp. 43-51.
5. N.V. Rudenko. Characteristics of traffic on highways and methods for assessing road safety. *International Journal of Information Technology and Energy Efficiency*. 2022. Vol. 7. No. 3-3 (25), pp. 100-106.
6. O.M. Batishcheva, A.I. Ganichev, A.G. Starikova. Factors influencing road safety in large cities. *Bulletin of transport of the Volga region*. 2023. No. 2 (98), pp. 110-115.
7. K.V. Baklanova. The influence of traffic flow parameters and road characteristics on road safety. *Intelligence. Innovation. Investments*. 2023. No. 2, pp. 99-110.
8. E.V. Kirillova, E.V. Budrina. Analysis of the efficiency of dedicated lanes for public transport based on GPS trackers of on-board equipment of rolling stock. *The world of transport and technological machines*. 2023. No. 1-1 (80), pp. 34-42.
9. V.S. Gavrilov. Logistics and management in road transport In the collection: Scientific and technical aspects of the development of the road transport complex 2023. *IX International Scientific and Practical Conference (correspondence-distance) within the framework of the 9th International Scientific Forum of the Donetsk People's Republic "Innovative prospects of Donbass: infrastructural and social economic development"*. Gorlovka, 2023, pp. 250-253.
10. M.A. Harutyunyan. Development of an algorithmic apparatus to ensure road safety. *Modeling, optimization and information technology*. 2023. Vol. 11. No. 3 (42), pp. 9-10.
11. G.V. Pushkareva, D.D. Khalturin. Analysis of methods for improving traffic safety at unregulated intersections in large cities (using the example of the city of Tomsk). *Bulletin of Tomsk State University of Architecture and Civil Engineering*. 2023. Vol. 25. No. 5, pp. 198-223.
12. I.N. Pugachev, N.G. Sheshera, V.I. Shcheglov. Analysis of geometric elements of roads using modern geographic information systems in assessing their accident rate. *Bulletin of Civil Engineers*. 2021. No. 3 (86), pp. 127-133.
13. I.N. Pugachev, N.G. Sheshera. Application of statistical analysis methods for assessing the parameters of transport flows and characteristics of the road network. *Dalnevost. legal Institute of the Ministry of Internal Affairs of Russia*. Khabarovsk: RIO DVUI Ministry of Internal Affairs of Russia. 2020. 108 p.

Information about authors:

Igor N. Pugachev, Deputy Director for Research, HFRC FEB RAS, Dr. tech. Sciences, Associate Professor, Khabarovsk Federal Research Center of the Far Eastern Branch of the Russian Academy of Sciences (HFRC FEB RAS), Khabarovsk, Russia

Stanislav S. Evtukov, Automotive and Highway Faculty, Head of the Department of Methodological Systems, Dr. tech. Sciences, Associate Professor, St. Petersburg State University of Architecture and Civil Engineering, St. Petersburg, Russia

Nikolai G. Sheshera, cand. tech. Sciences, Associate Professor of the Department of Information and Technical Support of the Department of Internal Affairs Far Eastern Law Institute of the Ministry of Internal Affairs of Russia, Khabarovsk, Russia

Denis E. Grigorov, head of the office of special disciplines of the Department of Information and Technical Support of the Department of Internal Affairs Far Eastern Law Institute of the Ministry of Internal Affairs of Russia, Khabarovsk, Russia