

УДК 631.362.322

**ПОГРУЖЕНИЕ ТРАНСПОРТНЫХ ПРОБЛЕМ В ЦИФРОВУЮ МАТРИЦУ***М. В. Бурнышева<sup>1</sup>, Г. А. Гальченко<sup>2</sup>, Д. С. Дроздов<sup>3</sup>*<sup>1,2</sup>Донской государственный технический университет (г. Ростов-на-Дону, Российская Федерация)<sup>3</sup>Южный федеральный университет (г. Ростов - на-Дону, Российская Федерация)

На основе анализа натуральных наблюдений — больших баз данных о работе транспортной системы г. Ростова-на-Дону предлагается разработанная компьютерная программа, позволяющая отслеживать нарушения графика движения автобусов. Создана база данных номеров автобусов г. Ростова-на-Дону, остановок и диспетчерского расписания каждого номера автобуса. Решается задача распознавания номера автобуса по входному изображению. Экспериментальным путем установлен компромисс между качеством и скоростью работы программы. Программа позволяет провести улучшение контрастности, бинаризацию и скелетизацию. После распознавания номера автобуса программа переходит на выбор номера маршрута, соответствующих остановок и проводит проверку выполнения диспетчерского расписания.

**Ключевые слова:** база данных, программа, метод распознавания, диспетчерское расписание.

**ENTERING TRANSPORT PROBLEMS IN THE DIGITAL MATRIX***M. V. Burnyashева<sup>1</sup>, G. A. Galchenko<sup>2</sup>, D. S. Drozdov<sup>3</sup>*<sup>1,2</sup>Don State Technical University, (Rostov-on-Don, Russian Federation)<sup>3</sup>Southern Federal University, (Rostov-on-Don, Russian Federation)

Based on the analysis of field observations — large databases on the operation of the transport system in Rostov-on-Don, a developed computer program is proposed that allows you to track violations of the bus schedule. A database of Rostov-on-Don bus numbers, stops, and dispatching schedules for each bus number has been created. The problem of recognizing the bus number from the input image is solved. Experimentally, a compromise was established between the quality and the speed of the program. The program allows you to improve contrast, binarization, and skeletonization. After recognizing the bus number, the program switches to selecting the route number, the corresponding stops, and checks the execution of the dispatcher's schedule

**Keywords:** database, program, recognition method, dispatching schedule

**Введение.** Человечество погружается в цифровую матрицу. Этот процесс обеспечивается сбором и оцифровыванием больших баз данных о потребителях.

Современное общество нуждается в постоянном увеличении объема транспортного сообщения, повышении надежности, безопасности и качества перевозок. Эти требования влекут за собой увеличение затрат на совершенствование транспортной инфраструктуры, превращение ее в гибкую, высокоуправляемую логистическую систему [1–4]. Для этого необходимо расставить и зафиксировать следующие приоритеты:

- общественного транспорта большей вместимости над транспортом малой вместимости (в том числе автобусов над маршрутными такси);
- общественного пассажирского транспорта над индивидуальным транспортом.

Таким образом, при выборе средства передвижения по городу приоритет отдается общественному транспорту, а не средствам индивидуального передвижения или маршрутным такси ввиду его большей провозной способности и безопасности.

При движении общественного транспорта необходимо следить, чтобы на всех маршрутах перевозки осуществлялись качественно, в полном объеме и точно по расписанию [5–6].

**Основная часть.** Для того чтобы решить вышеизложенные задачи необходимо:

- равномерно распределить пассажиропотоки между маршрутами;
- сделать работу перевозчиков рентабельной в условиях повышения сервиса и конкуренции между предприятиями;
- наладить электронную систему наблюдения и сделать ее работу эффективной.

На самом деле в повседневной жизни на автобусной остановке обычно скапливается много автобусов, они не помещаются в остановочные карманы, график движения нарушается (рис. 1).



Рис. 1. Скопление автобусов на остановках

Реализация электронной системы наблюдения требует решения многих задач. Остановимся на одной из них — оцифровывании процесса наблюдения за соблюдением графика движения на примере автобуса № 83 г. Ростова-на-Дону.

Для этого на первом этапе была создана база данных остановок [6] и диспетчерского расписания маршрута (рис. 2).

Код	Остановка	Прибытие	Отправлен
1	Центральный рынок	5:00:00	5:04:00
2	Большая Садовая	5:07:00	5:10:00
3	Максима Горького (проспект Б	5:14:00	5:20:00
4	Малюгиной	5:24:00	5:30:00
5	Текучёва	5:34:00	5:36:00
6	Комсомольская пл.	5:40:00	5:43:00
7	Шеболдаева	5:45:00	5:48:00
8	Лиловая	5:52:00	5:56:00
9	Боевой пер	6:01:00	6:03:00
10	Цезаря Куникова	6:07:00	6:11:00
11	пл. 2 Пятилетки	6:12:00	6:14:00
12	Куликовская	6:17:00	6:20:00
13	Иловайская	6:25:00	6:27:00
14	АТП-3	6:30:00	6:33:00
15	Радиаторный	6:36:00	6:38:00
16	Изыскательский пер.	6:41:00	6:42:00
17	Неклиновская	6:44:00	6:48:00
18	Котельная	6:52:00	6:54:00
19	Королёва проспект д.4	6:56:00	7:00:00
20	пл. Добровольского	7:04:00	7:08:00
21	проспект Королёва д.15	7:14:00	7:16:00
22	пл. Королёва	7:20:00	7:23:00
23	Вологодская	7:25:00	7:29:00
24	Атарбекова	7:30:00	7:33:00
25	Нижний Темерник	7:35:00	7:37:00
26	Рынок Геркулес	7:38:00	7:39:00
27	Лелюшенко	7:43:00	7:45:00
28	Миронова	7:48:00	7:50:00
29	Верхний Темерник	7:52:00	
*	(№)		

Рис. 2. База данных остановок и диспетчерского расписания автобуса № 83

Далее стоит задача — распознать номер автобуса (на рис. 1 №83). Процесс распознавания номера автобуса начинается с получения информации с видеокамер. Запускается программа распознавания номера.

**Краткая теория методов решения задачи распознавания.** С использованием технологий машинного обучения зачастую решается задача классификации (или задача распознавания). Идея разрешения этой проблемы заключается в том, чтобы научить программу определять, к какому классу относится заданный объект. Необходимо решить задачу распознавания номера автобуса, т.е. поставить задачу распознавания цифр. В таком случае по входному изображению программа должна дать ответ — число от 0 до 9, которое изображено на картинке. Классифицировать можно не только цифры, но и любые другие символы, к примеру, буквы или знаки препинания.

Программная реализация распознавания обычно состоит из двух частей — предобработка изображений и выбор, затем настройка модели.

Для создания модели требуется набор обучающих данных. В случае с цифрами, это размеченный набор картинок, в котором каждому изображению ставится в соответствие целое число от 0 до 9. На различных ресурсах существуют размеченные наборы цифр. Обучающая выборка базы сайта MNIST составляет 60 000 изображений. Эта база считается эталонной для работы с машинным обучением. Одновременно выделяется тестовая выборка — набор изображений, на которых проверяется работа программы. Тестовый набор MNIST около 10 000 изображений. Точность алгоритма (процент верно распознанных изображений) показывает, насколько хорошо программа справилась с задачей.

Наиболее распространённые алгоритмы для распознавания: «Метод k-ближайших соседей», «Дерево решений», «Случайный лес».

Для проведения анализа авторами был выбран алгоритм «Случайный лес». Идея алгоритма состоит в построении нескольких деревьев решений, каждому из которых на вход подается случайная подвыборка данных. Далее используется метод голосования — несколько деревьев для данного изображения определяют его класс. В качестве окончательного ответа принимается решение большинства. При использовании алгоритма «Случайный лес» необходимо определить оптимальное количество моделей (деревьев). Исследование показало, что график функции, иллюстрирующий зависимость точности от количества моделей, не является монотонным, имеются всплески показателя точности (рис. 3).

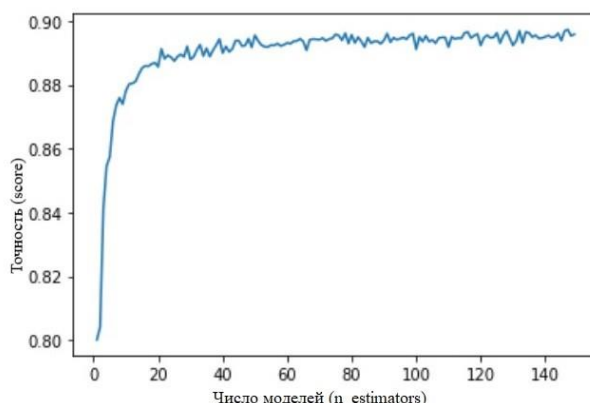


Рис. 3. Зависимость функции точности от количества моделей

Автоматический поиск оптимальных параметров показывает, что наилучшее число моделей (на отрезке [2; 200] с шагом в 2) составляет 194. В порядке компромисса между качеством и скоростью можно установить значение параметра `n_estimators` (оценщик) в 80 (исходя из графика на рис. 4.).

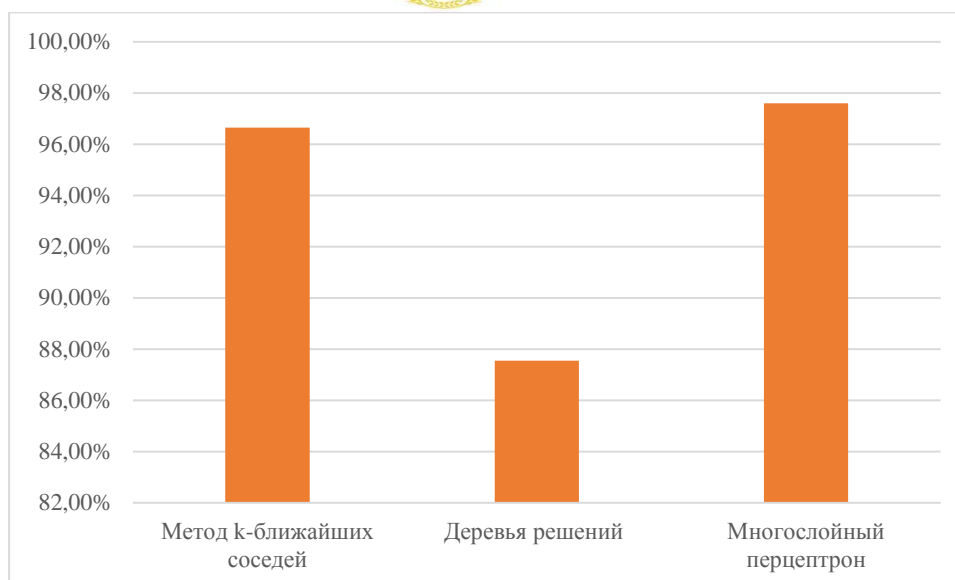


Рис. 4. Точность работы различных методов распознавания

Важный этап в машинном обучении — предобработка входных данных. На примере распознавания цифр этот этап — обработка изображений и сокращение вектора признаков.

Проанализирована следующая последовательность обработки данных: улучшение контрастности; бинаризация, скелетизация, сокращение вектора признаков. Первый этап состоит в выравнивании гистограммы изображения. Эквализация гистограммы представляет собой отдельный алгоритм и используется для повышения качества изображений. На следующем этапе происходит бинаризация изображения, то есть получение его черно-белого аналога. На третьем этапе происходит скелетизация.

Утончение — процесс, при котором все линии становятся тонкими и составляют 1 пиксель. Последний этап требует творческого подхода. На вход классификатору изображение (при использовании MNIST) подается в виде матрицы размером  $28 \times 28$  пикселей, что весьма замедляет работу алгоритмов. Предпочтительнее использовать более короткий, но не менее содержательный вектор признаков. На изображение цифры накладывается сетка, т.е. распознаваемое изображение помещается в матрицу, нумеруется каждая строка и каждый столбец. В качестве вектора признаков указывается число пересечений каждой прямой с изображением. Данный метод не является идеальным, но прост в реализации. Многие линии, расположенные по краям, не имеют пересечений с цифрой. Оптимально накладывать сетку, представляющую линии, проходящие через центр. В таком случае эффективность повышается, так как почти все линии будут иметь общие точки с цифрой (рис. 5), существенно сократится время работы алгоритма, не сильно понизив при этом качество работы. Так, на рассмотренных трёх алгоритмах объем верных распознаваний понизился в среднем на 4 %.

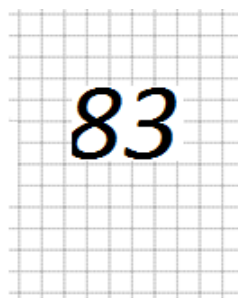


Рис. 5. Наложение сетки на рисунок

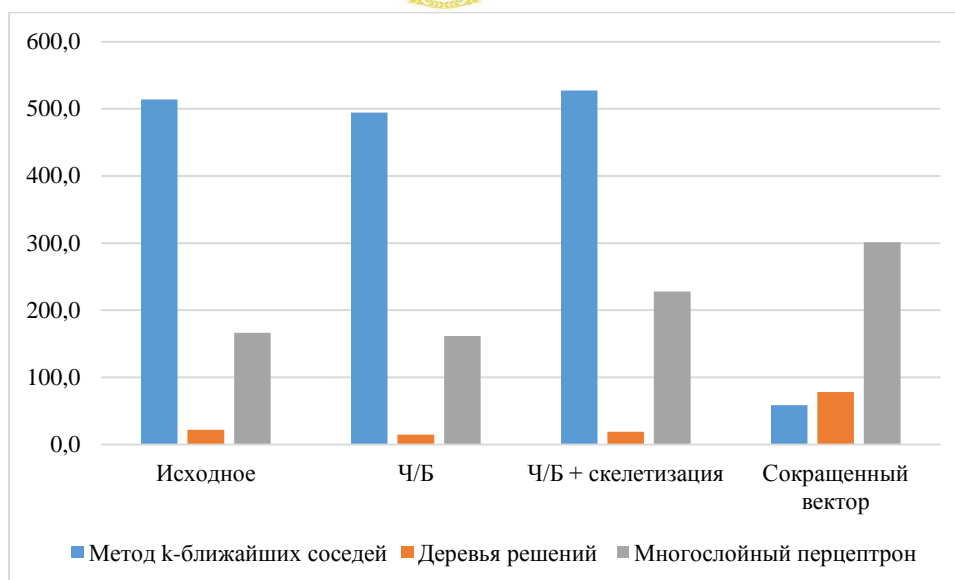


Рис. 6. Зависимость времени работы программы от предобработки изображений

Бинаризация несущественно ускоряет работу, несмотря на то, что пиксель стал занимать не 256 бит, а всего 2. Скелетизация лишь увеличила время работы программы и снизила качество. Это объясняется тем, что «схожесть» между двумя скелетизированными изображениями меньше, чем между обычными. После утончения снижается  $L_2$ -норма, которая показывает степень «одинаковости» изображений. Промежуточный этап позволил сократить вектор признаков и время работы метода k-ближайших соседей.

**Алгоритм работы дерева решений.** Программа задает обучающей выборке вопросы (с ответом только «да» или «нет»), по которым строится бинарное дерево. Далее для изображения алгоритм получает вероятности, согласно которым оно принадлежит каждому классу. Класс с наибольшей вероятностью и будет ответом.

**Метод случайного леса** — фрагмент программы на языке python.

```
def RandomForest(X_train, X_test):
    start_time = time.time()           #запуск таймера
    rf = RandomForestClassifier() #создаем классификатор методом случайного леса
    #обучаем модель, X_train — массив обучающих изображений,
    #labels_train — метки, эталонные объекты [0..9]
    #Во время обучения мы используем и параметры и метки, чтобы модель научилась
    #сортировать данные на основе параметров.
```

```
    rf.fit(X_train, labels_train)
    labels_pred = rf.predict(X_test)    # предсказание модели, в labels_pred уже цифры
    print("RandomForest")
    #Вычисление времени работы (обучения и предсказания) и оценка эффективности
    print("--- %s seconds ---" % (time.time() — start_time))
    print("--- %s percent ---" % accuracy_score(labels_test, labels_pred))
```

**Анализ леса деревьев решений** — фрагмент программы

```
def RandomForestAnalyseEstimators(X_train, X_test, m):
    n_estimators_values = range(1, m) #Количество решающих деревьев
    scores_data = pd.DataFrame()#Создание таблицы для занесения результатов
    np.random.seed(0)
    #Цикл по количеству деревьев
```



```

for n_estimators in n_estimators_values:
    clf = RandomForestClassifier(n_estimators=n_estimators, max_depth=20)
    clf.fit(X_train, labels_train)
#Предсказание и вычисление его точности
test_score = clf.score(X_test, labels_test)
temp_score_data = pd.DataFrame({
    'n_estimators': [n_estimators],
    'test_score': [test_score]
})
scores_data = scores_data.append(temp_score_data)
#Построение графика зависимости точности распознавания от количества моделей
#(решающих деревьев)
score_data_long = pd.melt(scores_data, id_vars=['n_estimators'], value_vars=['test_score'],
var_name='set_type', value_name='score')
sns.lineplot(x="n_estimators", y="score", hue="set_type", data=score_data_long)

```

После того, как номер автобуса распознан, программа переходит к выбору номера маршрута и соответствующих остановок (рис. 7, 8).

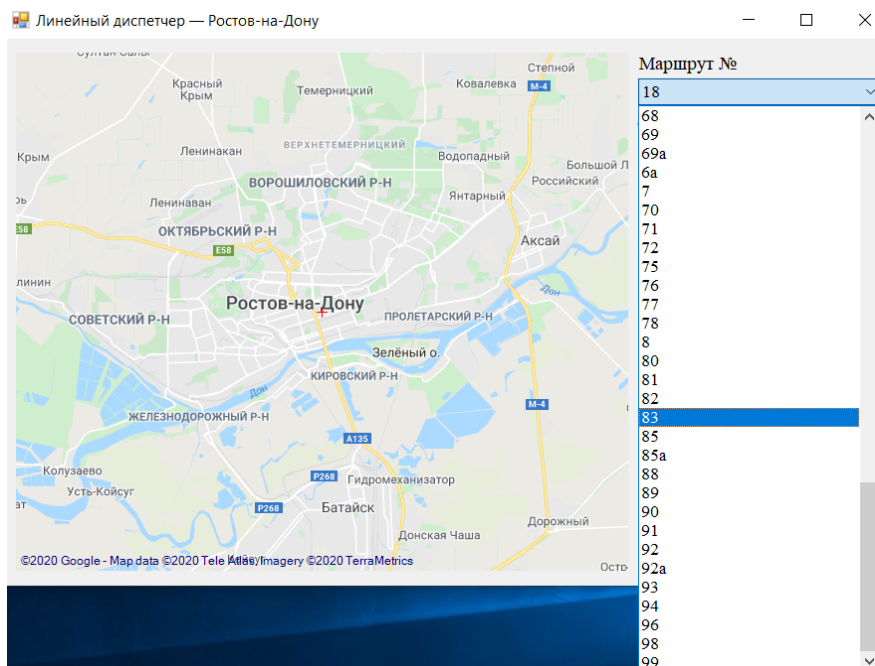


Рис. 7. Меню головной программы. Выбор номера маршрута

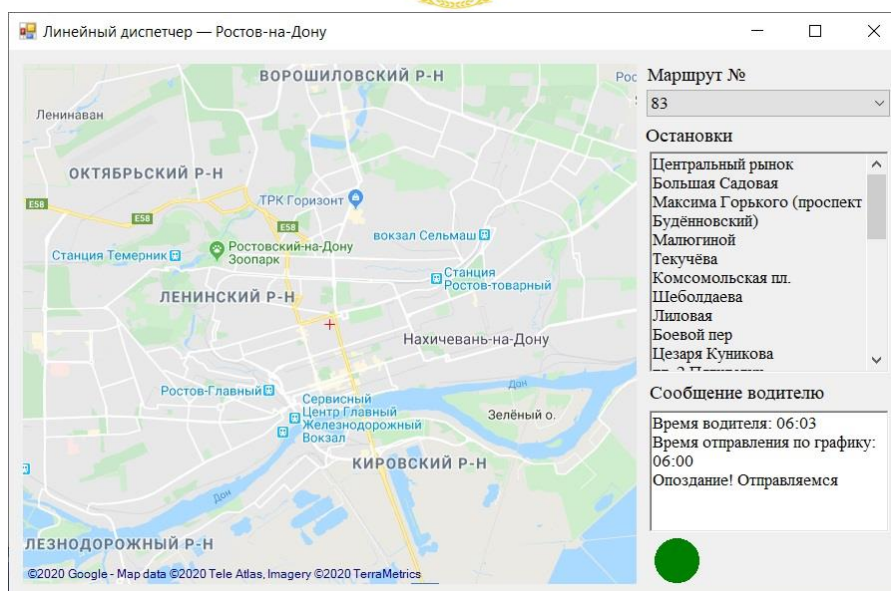


Рис. 8. БД остановок на маршруте автобуса № 83

**Заключение.** Анализ различных методов распознавания позволил сформировать оптимальный подход к поставленной проблеме. Разработанная программа распознавания номера автобуса на остановке встраивается в головную программу. В головной программе происходит выбор соответствующего номера маршрута, сравнение диспетчерского расписания и реального. Сообщение может быть отправлено водителю и в диспетчерский пункт [7, 8].

#### Библиографический список:

1. Марченко, Ю. В. Программный комплекс моделирования движения автотранспортных средств в городских условиях / Ю. В. Марченко, Г. А. Гальченко // Известия высших учебных заведений. Северо-Кавказский регион. Технические науки. — 2018. — № 1 (197) — С. 132–136.
2. Кущенко, С. В. Повышение эффективности организации движения на основе моделирования транспортных потоков : диссерт...канд. техн. наук / С. В. Кущенко. — Курск, 2012. — 134 с.
3. Попов, С. И. Использование математических методов и прикладных программных продуктов для расчета оптимизации дорожного движения / С. И. Попов, Г. А. Гальченко, Д. С. Дроздов / Исследование и проектирование интеллектуальных систем в автомобилестроении, авиастроении и машиностроении «ISMCA' 2019»: сб. тр. III Всерос. науч.-практ. конф. с междунар. уч. — Таганрог, 2019. — С. 117–121.
4. Программный комплекс построения оптимального маршрута при караванном движении транспортных средств / А. А. Короткий [и др.] // Инновационные технологии в науке и образовании (ИТНО-2019) : сб. трудов VII междунар. науч.-практ. конф., посв. 90-летию ДГТУ. — Ростов-на-Дону, 2019. — С. 68–71.
5. Гальченко, Г. А. Оптимизации грузоперевозок в логистической структуре АПК / А. А. Короткий, Г. А. Гальченко, Д. С. Дроздов // Состояние и перспективы развития агропромышленного комплекса : сб. науч. трудов XII Междунар. науч.-практ. конф. в рамках XXII Агропромышленного форума юга России и выставки «Интерагромаш». Донской гос. техн. ун-т. Аграрный науч. центр «Донской». — 2019. — С. 721–723.
6. Гальченко, Г. А. Использование баз данных на транспорте. Теория создания: учеб. пособие / Г. А. Гальченко, С. И. Попов, Ю. В. Марченко. — Ростов-на-Дону : Изд-во ДГТУ, 2018. — 111 с.

7. Короткий, А. А. Информационно-коммуникационная логистическая система для оптимизации транспортных маршрутов в урбанизированной среде / А. А. Короткий, Г. А. Гальченко, В. В. Иванов // Вестник Брянского гос. тех. ун-та. — 2018. — № 4 (65) — С. 63–67.

8. Гальченко, Г. А. Дигитализация и моделирование оптимального пути доставки грузов к станции Ростов-Товарная / Г. А. Гальченко, О. А. Останин, В. В. Иванов // Экономика и управление инновациями. — 2018. — № 4. — С. 61–70.

*Об авторах:*

**Бурняшева Мария Владимировна**, магистрант кафедры «Эксплуатация транспортных средств и логистика» Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), [m.burnyasheva@yandex.ru](mailto:m.burnyasheva@yandex.ru)

**Гальченко Галина Алексеевна**, доцент кафедры «Эксплуатация транспортных средств и логистика» Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), кандидат физико-математических наук, доцент, [ggalchenko@inbox.ru](mailto:ggalchenko@inbox.ru)

**Дроздов Дмитрий Сергеевич**, студент кафедры «Информатика и вычислительный эксперимент» Южного федерального университета (344000, Ростов-на-Дону, Большая Садовая, 105), [ds-drozdov@yandex.ru](mailto:ds-drozdov@yandex.ru)

*Authors:*

**Burnyasheva, Mariya V.**, Master's degree student, Department of Vehicle Operation and Logistics, Don State Technical University (1, Gagarin sq., Rostov-on-Don, RF, 344003), [m.burnyasheva@yandex.ru](mailto:m.burnyasheva@yandex.ru)

**Galchenko, Galina A.**, Associate professor, Department of Vehicle Operation and Logistics, Don State Technical University (1, Gagarin sq., Rostov-on-Don, RF, 344003), Cand.Sci., Associate professor, [ggalchenko@inbox.ru](mailto:ggalchenko@inbox.ru)

**Drozdov, Dmitriy S.**, Student, Department of Computer Science and Computational Experiment, Southern Federal University (105, Bolshaya Sadovaya street, Rostov-on-Don, RF, 344000), [ds-drozdov@yandex.ru](mailto:ds-drozdov@yandex.ru)